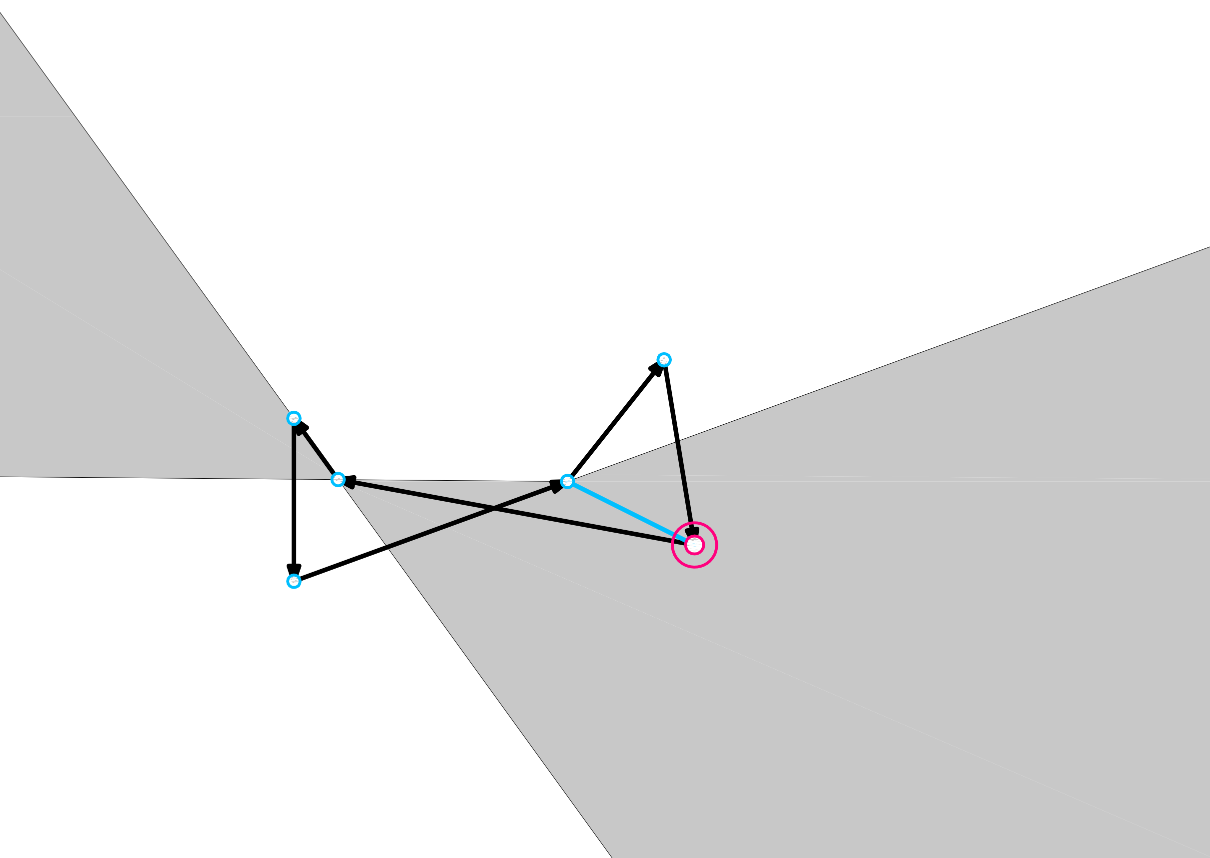
CONSTRAINT BASED GRAPHIC STATICS

A GEOMETRICAL SUPPORT
FOR COMPUTER-AIDED
STRUCTURAL EQUILIBRIUM DESIGN



Corentin Fivet · ingénieur civil architecte
LOCI · faculté d'architecture, d'ingénierie architecturale, d'urbanisme
Académie universitaire Louvain · Belgique · décembre 2013

thèse présentée en vue de l’obtention du grade de docteur en sciences de l’ingénieur

thèse financée par les Fonds spéciaux de recherche (FSR) de l'UCLouvain

composition of the jury

Denis Zastavni · supervisor
Architectural Engineer · Associate Professor
Faculty of Architecture, Architectural Engineering, Urbanism · UCLouvain · Belgium

Jean-François Cap · member of the accompanying committee
Structural Engineer · SECO · Technical Control Bureau for Construction · Belgium
Lecturer at the Louvain School of Engineering · UCLouvain · Belgium

Aurelio Muttoni · member of the accompanying committee
Structural Engineer · Full Professor of Concrete Structures
School of Architecture, Civil and Environmental Engineering · EPFLausanne · Switzerland

John Ochsendorf · reader
Structural Engineer · Associate Professor
Civil and Environmental Engineering and Architecture · MIT · Boston · USA

Laurent Ney · reader
Structural Engineer · Head of Ney & Partner sa · Belgium
Lecturer at the Université Libre de Bruxelles · Belgium

Pascal Lambrechts · reader
Mathematician · Professor
Institute of Mathematics and Physics · UCLouvain · Belgium

André De Herde · president of the jury
Architectural Engineer · Full Professor
Faculty of Architecture, Architectural Engineering, Urbanism · UCLouvain · Belgium

summary

area of research · tools and methods for structural design

keywords · computer-aided structural design · static equilibrium · constraint-based geometric solver · strut-and-tie models · graphic statics

abstract · This thesis introduces “*constraint-based graphic statics*”, a geometrical support for computer-aided structural design. This support increases the freedom with which the designer interacts with the plane static equilibriums being shaped.

Constraint-based graphic statics takes full advantage of geometry, both its visual expressiveness and its capacity to solve complex problems in simple terms. Accordingly, the approach builds on the two diagrams of classical graphic statics: a form diagram describing the geometry of a strut-and-tie network and a force diagram vectorially representing its inner static equilibrium. Two new devices improve the control of these diagrams: (1) nodes — considered as the only variables — are constrained within Boolean combinations of graphical regions; and (2) the user modifies these diagrams by means of successive operations whose geometric properties do not at any time jeopardise the static equilibrium of the strut-and-tie network.

These two devices offer useful features, such as the ability to describe, constrain and modify any static equilibrium using purely geometric grammar, the ability to compute and handle multiple solutions to a problem at the same time, the ability to switch the hierarchy of constraint dependencies, the ability to execute dynamic conditional statements graphically, the ability to compute full interdependency and therefore the ability to remove significantly the limitations of compass-and-straightedge constructions and, finally the ability to propagate some solution domains symbolically.

As a result, constraint-based graphic statics encourages the emergence of new structural design approaches that are highly interactive, precognitive and chronology-free: highly interactive because forces and geometries are simultaneously and dynamically steered by the designer; precognitive because the graphical region constraining each points marks out the set of available solutions before they are even explored by the user; and chronology-free because the deductive process undertaken by the designer can be switched whenever desired.

Applications cover the design of reticular systems — regardless of whether they are isostatic, indeterminate, prestressed, self-stressed or mechanisms — regular and irregular beams subject to bending, compression-only structures described by lines of thrusts, and structures that can be modelled with discontinuous stress fields.

This thesis is divided into five sections. The first section describes the context and expectations behind the sought environment. The second section defines the few fundamental axioms characterising graphic statics diagrams geometrically. They are then exploited in the third section to specify how geometric constraints applied to these diagrams can be maintained when points are dragged. Operations allowing the user to construct equilibriums are subsequently identified in the fourth section. The final section discusses the results.

résumé

domaine de recherche · outils et méthodes de conception structurale

mots-clefs · conception structurale assistée par ordinateur · équilibre statique · solver géométrique par contraintes · modèles de bielles-et-tirants · statique graphique

abrégé · Cette thèse présente la *"statique graphique par contraintes"*, un support géométrique pour la conception assistée par ordinateur de structures architecturales. Ce support est destiné à augmenter la liberté et le contrôle avec lesquels le concepteur donne forme aux équilibres statiqes plans.

La statique graphique par contraintes tire avantage de la géométrie, à la fois pour son expressivité visuelle et pour sa capacité à résoudre des problèmes complexes en termes simples. Elle s'appuie à cet égard sur les deux diagrammes de la statique graphique classique : le diagramme de situation décrivant la géométrie du réseau de bielles-et-tirants et le diagramme des forces représentant vectoriellement son équilibre statique. Deux nouveaux dispositifs enrichissent la manipulation de ces diagrammes : (1) la contrainte des noeuds, considérés comme uniques variables, à l'intérieur de combinaisons Booléennes de régions graphiques; (2) la modification successive de ces diagrammes au moyen d'opérations dont les propriétés géométriques ne mettent jamais l'équilibre statique du réseau de bielles-et-tirants en péril.

Ces deux dispositifs permettent de décrire, de contraindre et de modifier tout équilibre statique à l'aide d'une grammaire purement géométrique, de calculer et de manipuler simultanément l'ensemble des solutions multiples du problème, d'inverser la hiérarchie de dépendance des contraintes, de réaliser des déclarations conditionnelles dynamiques graphiquement, d'exécuter des interdépendances de contraintes, et par conséquent, de s'affranchir considérablement des limitations liées aux constructions à la règle et au compas et, enfin, de propager certains domaines de solutions de manière symbolique.

Par conséquent, la statique graphique par contraintes encourage l'émergence de nouvelles approches de conception structurale hautement interactives, précognitives et libres de chronologie: hautement interactives car forces et géométries sont dirigées simultanément et dynamiquement par le concepteur; précognitives car la région graphique contraignant chaque point

informe de l'ensemble des solutions possibles avant même que celles-ci ne soient explorées par l'utilisateur; et libres de chronologie car le processus déductif suivi par le concepteur peut être renversé à volonté.

Les applications concernées englobent la conception des systèmes réticulés (isostatiques, hyperstatiques, précontraints, auto-contraints ou mécanismes), des poutres régulières et irrégulières sujettes à la flexion, des structures en compression pure décrites par lignes de poussées et des structures pouvant être décrites par champs de contrainte discontinus.

La thèse est divisée en cinq parties. La première décrit le contexte et les attentes liées à l'environnement recherché. La seconde section définit les quelques axiomes fondamentaux caractérisant géométriquement les diagrammes de statique graphique. Ils sont ensuite exploités dans la troisième section pour spécifier comment les contraintes géométriques appliquées sur ces diagrammes sont maintenues lors du déplacement de points. Les opérations permettant à l'utilisateur de construire les équilibres sont ensuite identifiées dans la quatrième section. La dernière section discute des résultats obtenus.

contents

INTRODUCTION

The shaping of structures is an art in itself and the practice of it requires specific tools and methods.

Sub-section 01 ("fact: contemporary structural design practice", page 3) of this introductory section briefly sets out what the shaping of structures is, who does it, how it is done and what tools are used. Sub-section 02 ("critique: the lack of adequate tools for the initial shaping of structures", page 11) then provides a critique of these tools. Exemplary practices in structural design are then highlighted in sub-section 03 ("answers: exemplary practices", page 17) in order to identify alternatives approaches.

The purpose and features of the tool developed in this thesis are then outlined in sub-section 04 ("proposal: a tool to accompany the construction of static equilibriums", page 29). This tool is aimed at assisting the structural designer during the early phase of the shaping process.

Finally, sub-section 05 ("precedents", page 45) compares the proposed approach with existing tools that serve part of its purpose. The precise definition of the proposed tool will be detailed in the three sections that follow.

01 fact: contemporary structural design practice

the shaping of structures · Thinking about the structure of an object involves seeking to understand the way form and material behave when subjected to forces. Structures can be found everywhere in nature, from the microscopic arrangement of atoms to gigantic underground caves. However, structures can also be shaped explicitly in order to meet specific requirements. Other than ensuring stability under expected loads, these requirements are generally not related to purely mechanical considerations.

Indeed, form and material affect buildings and industrial and civil engineering constructions in many other ways. These include their integration within the spatial context, their architectural quality, their robustness, their sustainability over time, their recyclability, their process of building, their aesthetics, their symbolism, their cost and their functional uses.

Shaping a structure means juggling with all these considerations.

In contemporary practice, the two main protagonists of this art are the architect and the structural engineer.

the architect and the structural engineer · In ancient times and up until the Renaissance, building expertise was primarily the concern of craftsmen — *e.g.* carpenters and stone-cutters — who at times were directed by an individual — *i.e.* the master builder. As knowledge grew about the strength of materials, building techniques, stylistic forms and spatial qualities, practitioners became more specialised and a schism occurred in the eighteenth century to produce two new practitioners: the architect and the engineer — read Picon·1988, Addis·2007 and Saint·2007 for further insight into this.

Although the architect and the engineer certainly develop different sensibilities and knowledge, it is a fairly crude caricature to associate the former with the artist and the latter with the pragmatic (Addis·1994, Wells·2008 and

Flury/…·2012). Another no less caricatured way of differentiating between them would be to say that the former has responsibility for the spatial form and the latter has responsibility for the structure's stability.

This comparison highlights the endless interference between the architect and the engineer: the former cannot do anything to the spatial form without challenging its stability and the latter cannot guarantee the stability of the structure without affecting its spatial form. Owing to this interference, they are forced to work together in one way or another.

Based on Baumberger·2012, four basic relationships between the architect and the structural engineer can be identified: monologues from the architect, monologues from the structural engineer, dialogues and soliloquies.

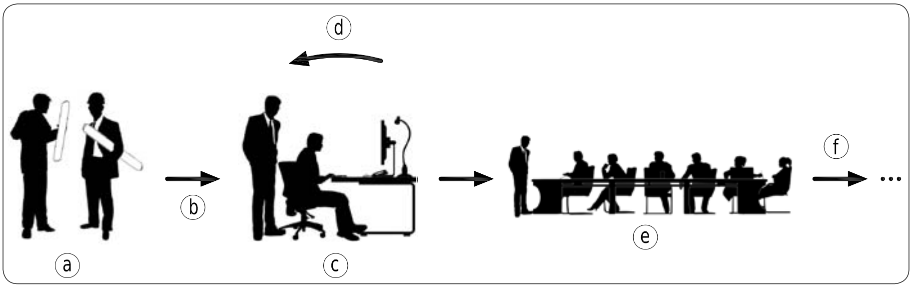
Monologues result from the total control of the project by the architect or the engineer, with the other at his service. Monologues from architects generally lead to formal, demonstrative buildings. Monologues from engineers generally lead to pragmatic, cost-effective buildings.

In contrast, dialogues occur when "*architect and engineer are equal partners in the discussion*" (Baumberger·2012, page 60). This kind of cooperation is generally hard to maintain because of the opposition between protecting personal egos and the necessary challenging of the other's work. However, it usually tends to produce the most successful outcomes.

Finally, soliloquies occur in rare cases where the architect and the engineer are one and the same person. As a consequence of both domains being mastered, the work produced occasionally has a tendency to be fairly demonstrative.

These four different relationships are particularly noticeable at the beginning of the project (figure 1, step a) when the initial ideas and design decisions are established — in other words during the drafting or concept stage.

figure 1 schematic illustration of the contemporary structural design process (characters from http://architexts.us).



the structural design process · The role of the structural engineer after the concept stage is relatively similar in every office. Depending on the level of specificity in the sketched project, in the first instance the engineer adds to it so that a load-bearing structure can be identified (figure 1, step b). Consciously or not, it is at this moment that the future behaviour of the structure is fixed — *i.e.* the way the structure behaves once loaded.

The engineer then uses tools and methods (figure 1, step c) to ensure that the proposed structure will resist the stresses effectively to which it may be subjected, will remain stable and safe in extreme situations and, over time, will deform very slightly compared to its intended purposes, will be feasible with available, cost-effective materials and methods of construction and, on occasion, will represent an optimum of these various objectives.

The results obtained by this step might compel the designer to backtrack (figure 1, step d) in order to readjust assumptions made in step b, before completing the computations of step c again. Consequently steps b, c and d form a cyclical process.

As the project becomes more concrete, the engineer proposes the structural solution to other design team members (figure 1, step e): the architect, but also the client, the building contractor and all the other specialists, such as HVAC consultants, urban planners, fire authorities *etc*.

Coming away from these meetings, the structural engineer modifies the proposal (figure 1, steps d and b), subsequently re-computes it (figure 1, step c) and resubmits it to the other members of the design team. This results in a new cyclical process (steps b, c, e, d) of varying length, ultimately leading to the completion of every detail (figure 1, step f) regarding geometries, manufacturing processes and implementation.

available theories ·There are three types of theories available to structural engineers to fulfil their duty:

(1) theories concerning the rheological understanding of materials. Research in this field mainly comprises laboratory testing of full-scale structural elements and the results produced remain relevant over time. New tests are only necessary when new building materials — *e.g.* high-performance concrete or fibre-reinforced polymers — or new im-

plementations — *e.g.* interlocking cross-laminated timber (Smith·2011) or reinforced timbers (Trautz/Koj·2009a and Trautz/Koj·2009b) — emerge

(2) theories concerning the understanding of the behaviour of structures. The most significant advances in this field date back to the nineteenth century and the first half of the twentieth century and still constitute the core of current knowledge (Heyman·1996, Heyman·1999b and Charlton·2002)

(3) theories concerning tools and methods enabling the designer to approach and predict the effective behaviour of the structure. This has always been the principal concern of research efforts, and it is still the case today.

the role of design tools · The presence of and importance attached to the development of appropriate tools and methods is directly linked to the essence of structural design: the shaping of structures is not a science. It is not a process undertaken to find the unique solution to a given problem through rational calculations. It is a real project. It is a nonlinear process at the beginning of which the definition of the problem is as unknown as the final result.

“[Structural] *art is solving problems which cannot be formulated before they have been solved. The search goes on, until a solution is found, which is deemed to be satisfactory. There are always many possible solutions, the search is for the best — but there is no best — just more or less good.*” (Ove Arup, in Addis·1994, page 7)

“*All the great masters of structural design have reminded us repeatedly that structural design is not a science; it is a craft that relies on judgment rather than absolute certainty*” (Allen·2009, page xiv)

Structural design is indeed a craft and the engineered product will owe more to the designer’s previous experience and the nature of the tools than the purely theoretical knowledge acquired. Indeed experience, intuition and tools greatly influence the final product in a variety of ways: they condition the conduct of the design process; they determine the set of parameters that can be acted upon, the set of values that will result and the set of choices that have to be made beforehand; and they impose the speed of execution, the level of interactivity, the level of accuracy *etc.*

Therefore the choice of tools and methods is dependent not only on the initial data available — *e.g.* the material that will be used or the quality of the bearing soil — but also on the desired results — which sometimes lead engineers and architects to develop their own customised tools and methods in order to achieve particular results within particular frameworks (Krasny·2008).

available tools and methods · The range of tools available to the structural designer is fairly extensive and heterogeneous. It encompasses a huge variety of tasks and a huge variety of means. Firstly, the designer has all the tools of the architect at his disposal, either manual instruments or software solutions created to sketch, depict, reproduce, model and communicate — *e.g.* pencils, compasses, straightedges, foam models and computer-aided drawing tools. A comprehensive list of these tools can be found in Krasny·2008, page 149. Secondly, there are additional tools specifically related to the shaping of the structure or its study.

Rather than compiling an extensive and heterogeneous list of past and contemporary structural design tools, their common and different properties are identified here according to four partial but complementary questions:

(1) What is the assumed rheological model used by the tool?
(2) What is the chosen model of representation of the structure?
(3) What are the required inputs and the expected outputs?
(4) What is the inner solution method?

The following paragraphs attempt to formulate a list of possible answers to these four questions — each answer can be looked upon as describing a group of tools sharing the same property.

(1) Since the real rheological behaviours of material are too complex to be handled efficiently, theoretical simplifying hypotheses are made about them. As theories are generally founded upon one particular ideal rheological model and most tools build on one specific theory, these hypotheses consequently condition the tools' scope of application. The following rheological models are possible answers to criterion 1:

- non-deformable — which was the only assumption encountered before the Renaissance and Galileo
- brittle elastic — mainly used to compute deflections
- rigid plastic — mainly used to check stability
- elasto-plastic — see Ruiz/Muttoni·2007 for an example of an application
- …

(2) Models of representations of the structure and its behaviour arising from criterion 2 establish particularly diverse groups of tools:

- scaled physical models (Drew·1976, Huerta·2006a, Ney/…·2010a)
- elementary geometrical rules (Huerta·2006c)
- typical typologies previously encountered or models found in nature
- physical equations — the most widely taught models in contemporary structural design classes
- calculators and spreadsheet files
- indicators of performance (Samyn·2004)
- graphs (Levens·1975)
- photoelastic materials (Heywood·1969)
- strut-and-tie models
- diagrams of graphic statics (Rankine·1858, Maxwell·1864)
- load paths (Palmisano/…·2008)
- discretisations by finite elements (Frey/Jirousek·2001)
- discontinuous stress fields (Marti·1985, Muttoni/…·1997)
- continuous stress fields (Muttoni/…·1997)
- …

(3) Criterion 3 distinguishes tools by grouping together the ones that require the same input data and produce the same output data. A classification of this kind produces six major groups:

- sketching tools — they begin with a blank page and some predetermined requirements about the structure and its use; they lead to the primary geometric and/or mechanical establishments of the structure; they are sometimes referred to as tools for the conceptual stage or structural morphogenesis
- analysis tools — they take loads and the geometries of the structure as inputs and produce a description of internal stresses and/or deformations
- sizing tools — they take loads and rough geometries (such as the position and length of rods in a reticular structure or the bounding contour of a concrete beam) as initial parameters and provide the remaining geometries (respectively the cross-sectional geometries of the rods and the type of reinforcement bars in the beam) required to sustain the loads occurring in the structure

- checking tools — they take loads, geometries and inner stresses as inputs and compute a Boolean value which says whether the structure is stable or not; nowadays most requirements for stability checks come from government codes and standards
- optimisation tools — they take loads, the geometries of a structure and an objective performance condition as initial data and provide a modification of this structure that satisfies the condition or comes as close as possible to it; tools falling within this category can be distinguished further by clarifying whether the modification affects the shape of a continuum structure, the topology of a network of rods or the sizes of these rods
- form-finding tools — they take a consistent set of loads, typological and boundary conditions as arguments and return a shape that satisfies these conditions within the laws of equilibrium and strength of materials
- …

The distinction between these groups of tools is not always as rigorous in literature or in practice — *e.g.* analysis, sizing and checking tools are often confused and the same applies for sketching tools and form-finding tools. It should also be noted that some optimisation and form-finding tools may be seen as iterations of analysis and checking tools.

(4) Although criterion 4 is also applicable for recognising hand-guided solving methods — such as

- the method of joint (Fairman/…·1932, page 26)
- the method of section (Fairman/…·1932, page 34)
- the method of substitution (Fairman/…·1932, page 42)

to identify internal stresses in trusses, for example — it is specifically useful for distinguishing computerised tools from one another. Such tools are generally made with algorithms that combine and adapt multiple strategies, including the use of:

- force densities
- virtual works
- dynamic relaxations
- particle spring systems
- rainflow analogies
- solid isotropic material with penalisation strategies

- sequential element rejections and admission strategies (also known as bi-directional evolutionary structural optimisation)
- genetic strategies
- …

Surveys on this subject can be found in Christensen/…·2009, Rozvany·2009, Spillers/…·2009 and Deaton/…·2013.

popular habits · Among all the tools available, two are used much more in contemporary practice than any of the others:

(a) If the project can be made of basic, well-known and easy-to-build typologies, engineers generally choose them without any further consideration concerning their shaping. They analyse each structural part directly by entering the handful of parameters into a spreadsheet — *e.g.* Microsoft Excel — from which they then obtain inner stresses and, in the most comprehensive cases, a building code-compliant sizing.

(b) Otherwise — *i.e.* when complex or less common typologies are encountered — engineers would generally simply reproduce the architect's conceptual sketch inside a finite-element analysis tool from which they automatically acquire a depiction of the magnitudes of principal stresses. They would then modify the initial sketch surgically until the structure meets expected minimum requirements.

As a result, today's structural design practices are usually isolated in a process that determines the size of the structural parts following the analysis of a predetermined shape. The next sub-section develops a targeted critique of these practices.

02 critique: the lack of adequate tools for the initial shaping of structures

drawbacks of (most) contemporary structural design tools · The tools and methods currently most widely used — *i.e.* computerised analysis, optimisation and form-finding tools — have benefited from developments in informatics since the mid-twentieth century. They now allow the designers to achieve unprecedented results: they approach real behaviour with increasing accuracy; they allow the detailed design of daring structures with growing confidence and produce the optimum shapes of increasingly complex structures. However, most of these tools have serious drawbacks in the context of the chronological process of contemporary structural design. These drawbacks can be synthesised as follows:

(1) These tools require important choices to be made prior to their use, such as the choice of the tool used, the nature of the model and/or the simplifying assumptions. This might force the designer to make these choices too early in the process, which would be prejudicial because (a) tools are generally highly specific and time-consuming, (b) the model might not have the desired degree of accuracy and (c) initial simplifying assumptions might end up being at odds with the final structural behaviour. Moreover, these tools give no or very limited help in determining these choices or modifying them afterwards in the process.

Computerised tools used by architects present the same issue:

> *"With a computer you arrive at a precise solution very quickly, a precise rendering that gives you an idea at a very early stage. One of the things that this tool means which while not dangerous is somewhat problematic is that you make decisions too fast"*
> (Anne Lacaton, in Krasny·2008, page 82)

(2) These tools impose a specific chronology of resolution — *i.e.* precise inputs and outputs. Unless the structural designer dedicates sufficient time to writing his own software, he is forced to resolve the problem in the way imposed by the tool, make lengthy detours in order to get to a solution that could have been found directly or, worse still, favour a solution that can be computed by the software rather than another solution that cannot be computed, but is just as easy to solve manually.

For example, current undesirable issues of this kind are:

- What geometric changes would make a particular rod in a truss become in tension?
- What additional weight would enable the reaction stresses on a given footing to be sufficiently vertical so as to be supported by the ground?

These two questions are not unsolvable, they just require the use of static equilibrium laws in a different way from that proposed by customary analysis tools.

(3) These tools usually have such precise and limited purposes that the complete, composite structure generally has to be divided into several sub-parts that are explored in isolation, irrespective of their interactions — *i.e.* using separate, independent tools. For reasons of time efficiency, this would either require the user to make as few changes in each sub-part as possible or would result in uncoordinated local corrections without a global assessment of any kind.

(4) Inner computations of these tools are hermetic black boxes for most users, impervious to customised adjustments by the user. This has the potential of resulting in elegant resolutions being dropped and computational efficiency being decreased.

(5) The lack of interactivity also stems from the fact that users have to answer very specific questions that are directed by the software's own algorithmic reasoning. It follows that users all put their trust in the software and gradually become unaccustomed to asking questions themselves. This does not encourage original contributions and may stifle creativity:

"Structural analysis and calculation have become increasingly precise and detailed. Proportioning pushed to its limits has allowed structures to be even more daring and efficient, but unfortunately all this has had a negative effect on structural design, leading to a slow and inexorable deterioration of the creative element involved." (Muttoni·2005, page v)

(6) Black boxes also have the disadvantage of producing results that are difficult to interpret or even to understand. This does not encourage a direct return to initial choices in order to fix or improve them. Hence, it does not favour adequate control of the structural behaviour being shaped:

"Computer programs look only at local stresses and have a flagrant disregard for the principles of structure." (Addis·1994, page 12)

(7) Owing to this, the results are rarely communicated to the architect who subsequently becomes distanced from the structural issue:

"For the architect too, the separation of disciplines has not solely brought benefits. The growing difficulty of understanding how structures function definitely represents an impoverishment." (Muttoni·2005, page v)

This general list can be supplemented by the following issues specifically linked to finite-element models.

(8) The creation of adequate finite-element models can be rather complex, is often poorly mastered and might produce unexpected or misunderstood results.

(9) This creation and its computation can be time-consuming too and can slow down the design flow considerably, hinder the user's creativity and discourage important cross checks:

"It is usual that the modelling takes so much time and effort that little is left for verification and validation of results, alternative designs, writing reports and documentation, backups and reviews by other analysts." (Rodríguez·2010)

(10) Analysts might not devote their energy to the right place since they may be tempted to look for a neat model rather than an efficient mode of construction:

"Analyst can become so involved in FEA [Finite Element Analysis] *that the link between the real structure and the model may be forgotten."* (Rodríguez·2010)

(11) Analysts are required to have a precise idea of how they will conduct the modelling beforehand, otherwise they would be forced to start again due to unexpected discoveries:

"The geometry should be defined explicitly and clearly: it is absolutely impossible to calculate anything without a previous design. [...] *On the other hand, oversimplification can leave out critical load paths."* (Rodríguez·2010)

(12) Last but not least, tools of this kind offer very few opportunities to influence one particular structural behaviour over another. If they do, opportunities consist of detours that might have a significant impact on the structure's stability.

In conclusion, current computerised analysis, optimisation and form-finding tools do not easily allow the designer to be entirely successful in controlling the process of computation and the shaping of the structure.

associated risks · For most common basic projects developed in contemporary offices, these drawbacks will have little detrimental impact on the structure's quality. However, structures that require closer attention might suffer from weaknesses from various perspectives:

- safety — *e.g.* the structure might suffer from a lack of robustness; in the worst case scenarios, a misunderstanding of the structural behaviour might lead the designer to produce a mechanically unsafe construction
- the architectural project — *e.g.* the structure might not fit with initial design intentions or might not be relevant to other structural intentions
- the current economic situation — *e.g.* structural elements might be over-dimensioned or superfluous; the structural system might incur additional costs for costly, better performing materials, maintenance, repair *etc.*
- new materials being used perhaps — *e.g.* the designed structural behaviour might not fit with the rheological behaviour of the material, manufacturing processes and methods of construction
- environmental concerns — *e.g.* the structure might require greater use of raw materials than absolutely necessary or generate more pollution
- any other consideration that is non-quantifiable and requires ongoing and direct input from the designer's sensitivity and creativity.

Furthermore, the impact of these dangers is aggravated by the desire to minimise risk and increasingly stringent standards — *e.g.* those governing fire resistance, structural robustness and site safety.

the need for new tools · This observation does not mean in any sense that contemporary computerised analysis, optimisation and form-finding tools are unsuitable or dangerous for structural design. Rather it means that (a) they play a limited, precise role in the design process and (b) it is better not to use them alone.

In figure 1, page 4, these tools are indeed used during only one step of the structural design process — step c, after the initial typologies, geometric conditions and assumptions about structural behaviour have been determined and before their results are interpreted and reworked.

There are different ways to avoid the potential pitfalls set out above. Two main options emerge. The first enhances current computerised analysis, optimisation and form-finding tools until the drawbacks mentioned above are eliminated. The second improves the accomplishment of steps other than step c in order to consolidate the shaping process as a whole. This thesis explores the second option.

In contemporary practice, there is generally no tool to accompany the other steps — *i.e.* (step a) when the very first design assumptions are made by the architect and the engineer, (step b) when the structural behaviour is shaped for the first time, (step d) when it is amended following more detailed analysis, and (step e) when it is discussed with other members of the design team. Apart from a few exceptions, designers have confidence instead in their experience and intuition.

The few existing tools that might be used during these steps — *e.g.* exploratory physical models and graphical hand calculations — are generally avoided because (1) (it is thought that) they require more time and energy than is available and/or (2) it is difficult to implement them in the usual workflow since they rely on means that are too exotic for the computer.

Some might think that the current structural design situation is entirely satisfactory and that the designer does not need the assistance of computers during the other process steps. They may be right. However, this thesis is rooted in the principle that better results can always be attained and that the relevance and efficiency of a new tool cannot be judged before it exists or before it is given in-depth consideration.

These are all the motivations that stress the need for new tools specifically devised to assist steps a, b, d and e in figure 1, page 4.

The next sub-section seeks to identify what such tools might look like by examining strategies that have been used by renowned structural designers. In this respect, it will be demonstrated that some approaches even go so far as to render analysis tools — step c in figure 1, page 4 — unnecessary.

> “*There is an old saying which goes something like this:* ‘An engineer is a man who can do for a dollar what any fool can do for two.’ *Its emphasis on ingenuity is praiseworthy, but it has been seen too often as a justification for much that is cheap and nasty in engineering. It has been taken to mean that engineering is nothing more than the achievement of clearly specified technological objectives for the lowest possible cost in cash. This view has been reinforced for engineering students by the fact that with a few notable exceptions, text books entitled* 'Design of Structures' *are predominantly concerned with the techniques of computational analysis*” (Holgate·1986, page 6)

03 answers: exemplary practices

learning from the great masters · The previous sub-section urged the search for new tools devised to supplement computerised analysis, optimisation and form-finding tools. This sub-section highlights strategies developed by recognised structural engineers who have worked wonders in a context in which computers were not necessarily available. It is then assumed that these proven strategies are still relevant today and would be even more effective if they had the cautious benefit of a computer's speed.

clarity, speed and interactivity for creativity and intuition · From a general point of view, structural designers who care about quality projects seek to employ tools and methods that maximise the opportunity for their creativity and intuition to percolate through:

> "[...] *Creativity is necessary not just for issues around form, but also for purely technical aspects: processes, materials and static systems. This creativity is the difference between people who are happy to calculate and real engineers.*" (Jürg Conzett, in Conzett/Solt·2008, page 29)

> "*There is no method that enables us automatically to discover the most adequate structural type to fit a specific problem, as it is faced by the designer. The achievement of the final solution is largely a matter of habit, intuition, imagination, common sense and personal attitude. Only the accumulation of experience can shorten the necessary labour or trial and error involved in the selection of one among the different possible alternatives.*" (Eduardo Torroja, in Addis·1994)

Structural designers consequently prefer tools that are capable of clarity, speed and interactivity:

(1) clarity in order to understand and control the structural project, to be aware of every design choice and its impact

(2) speed because the process must be carried out in a continuous, fluid and relatively fast workflow so as not to inhibit the designer's creative energy

(3) interactive because structural designers must be able to place non-quantifiable data — from their own experience or sudden intuition — onto the computation at any time.

Clarity, speed and interactivity are the valued qualities in the process. The techniques that favour these qualities are many and varied. The paragraphs below highlight four of them:

- prior definition of the structural behaviour
- design-oriented use of simplifying assumptions
- problem reduction guaranteeing permanent control
- extensive use of graphical methods and geometry.

prior definition of the structural behaviour · Defining the structural behaviour before analysis is about knowing what role the material will play in the structural system before the system is even drawn. The intention is therefore to set the general course of the design process from the outset in order to avoid messing about unproductively. How can a designer guide the conception of his building in a considered and appropriate manner with regards to the material used if the structure's behaviour is only discovered after analysis?

This early definition is important because if the shape of the structure is determined independently of the structural behaviour, the resulting behaviour might be too complex to understand and hence modified after analysis. This might lead to an inefficient and sometimes unsafe use of the material.

On the other hand, the prior definition of the structural behaviour also allows the designer to employ design methods that are particularly suited to the structure and hence simpler and more rapid.

The requirement for this technique is evident when materials have certain limitations of strength. For example, it was partly due to the compression-only behaviour of stone being determined beforehand that medieval masters achieved efficient architectural forms in gothic cathedrals. Likewise, the architect and engineer Eladio Dieste (1917-2000) drew efficient forms because he established the structural behaviour of brick walls from the outset and designed them taking this into account (Dieste/…·2001, Anderson·2004). A similar observation may be made about traditional wood joints, the design of which must be preceded by considerations about the expected rheological behaviour of wood and its manufacture.

When materials have fewer restrictions concerning their strength, the requirement for the prior definition of the structural behaviour is unfortunately less obvious, but just as relevant. As an illustration, Swiss engineer Robert Maillart (1872-1940) made exemplary use of it (Zastavni·2009). Depending on the context — *e.g.* the soil properties, the required span and the required width of the deck — he selected an appropriate typology for the bridge — *e.g.* a three-hinged arch bridge (as for the Salginatobel Bridge in figure 2) or a deck-stiffened arch bridge. Depending on the typology selected, he chose the most appropriate design assumptions and subsequently the most direct design method that allowed him to make as few calculations as possible. These efficient methods led to efficient structures.

figure 2 the Salginatobel bridge designed by Robert Maillart, picture by Andrea Badrutt.

design-oriented use of simplifying assumptions · It is still impossible nowadays to predict the real behaviour of structures in its full complexity. Hopefully this is not a big issue since the goal of the engineer is to ensure the safety of the structure, not to model reality as accurately as possible.

> *"A real structure is, in fact, supported externally in a way which is unknown (and unknowable) to the engineer, who nevertheless is required to make a design"* (Heyman·2008a, page x)

In order to produce the design, engineers work with idealised representations that are known, understood and, above all, controllable. Consequently there is always a gap between the real material behaviour and its theoretical behaviour, whichever approach is chosen. These gaps are qualified by so called simplifying assumptions.

Each simplifying assumption encourages specific methods and tools and hence has different values of efficiency regarding the design process. Some suit the analysis of existing structures better, others the initial shaping of structures. The choice of an appropriate assumption will generally make the process more straightforward.

Of all the existing simplifying assumptions, those that allow the application of the lower-bound theorem of plasticity are probably the ones that have provided the most rapid methods for the initial shaping of structures. The explicit formulation of the lower-bound theorem has its origin in practical experiments conducted during the first half of the twentieth century. These experiments suggested that yielding of material involves higher inner stresses than those computed with elastic theory (Heyman·1996 pages 127-153). The lower-bound theorem, also called the safe theorem or static theorem of plastic theory, has been expressed as follows:

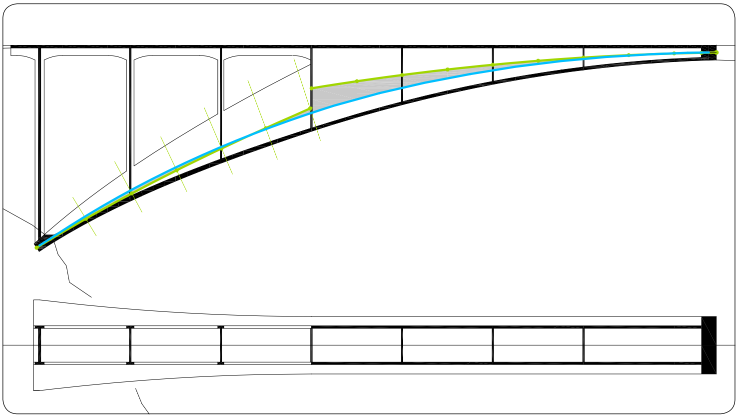
> *"A load calculated from an equilibrium state which satisfies the yield condition is a lower bound on the value of the collapse load."* (A.A. Gvozdev in 1938, translated by Heyman·1996, page 141)

The use of this theorem is subject to two conditions: (1) the set of plastic hinges produced by the structure cannot cause its cinematic collapse and (2) the material must present perfect plastic behaviour. Other formulations of this theorem have since been proposed to accommodate specific materials — see for example Heyman·1996 (page 144) or Heyman·2008a (appendix B, page 123) for reticular structures, Heyman·1995 for masonry structures and Muttoni/...·1997 (page 9) for stress fields in reinforced concrete.

This theorem represents an outstanding design tool (Zastavni·2008a, page 54) since it does not require the designer to find the actual state of stresses that will occur in the structure. The identification of one statically admissible stress field is sufficient to guarantee the stability of the structure, regardless of the real behaviour of the structure (Frey·2000). The entire dimensioning of the structure may therefore be based on a single adequate stress field.

This principle had already been widely employed intuitively before its scientific statement — examples can be found in Ochsendorf·2005. A remarkable working method is the one used by Robert Maillart for the design of the Salginatobel Bridge. By assuming concrete's plastic behaviour, Maillart designed the three-hinged arch bridge by graphically computing a single line of thrust passing through each hinge in equilibrium with the dead load of each segment of the discretised arch. By multiplying the load of the thrust by the eccentricity between this line of thrust and the line of centroids of the arch, Maillart directly computed the bending moments in the arch (figure 3). This single graphical result allowed him to modify the geometries of the arch efficiently in order to minimise its bending moments repeatedly.

figure 3
description of half of the Salginatobel bridge designed by Robert Maillart; the line of thrust is the continuous curve and the line of centroids consists of two disconnected curves.



problem reduction guaranteeing permanent control · A third technique of a design process promoting clarity, speed and interactivity is to reduce the design issue to a small number of critical parameters (or equations, variables or relationships) that alone control all the major questions of the design issue. This is achieved by making a set of parameters secondary and dependent on the controlling set of parameters, such that the complexity of these dependencies can be provisionally forgotten by the designer. This allows a close focus on a minimum set of data while handling the entire issue as a whole.

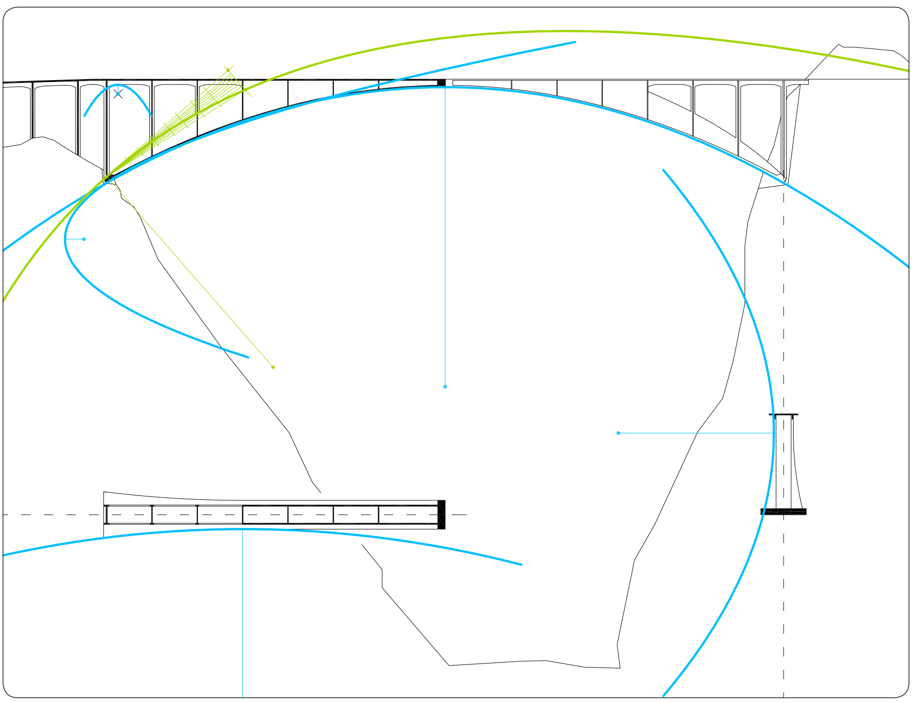
Applications of this technique have been accomplished in many different forms. The following paragraphs illustrate five of them.

(1) Antoni Gaudi (1852·1926) used hanging models to control both the shape of the buildings and their stability under dead loads (Krasny·2008 page 58, Huerta·2006a); in the design of vaults all that was required was to add and remove weights and ropes

(2) Robert Maillart (1872·1940) drew parabolas in order to correct repeatedly the apparently free-form bottom and upper chords of the Salginatobel Bridge (figure 4; the construction method of parabolas he used permitted him to handle the entire curve easily by moving just two crossing points and rotating the directrix — the crossing-points and the orientation of the directrix being the parameters (Fivet/Zastavni·2012)

(3) Felix Candela (1910·1997) synthesised the entire structural (and, to some extent, architectural) problem into a single equation describing hyperbolic paraboloids; the alteration of this single equation enabled him to accommodate the general shape and its boundary conditions freely (Faber/Candela·1963, Garlock/...·2008);

figure 4 reconstruction of some parabolas that Robert Maillart drew on the first working-drawings for the Salginatobel bridge; the bundle of lines depict the construction process of a parabola from two crossing-points and the orientation of a directrix.



(4) The algorithms of contemporary engineers dealing with optimisation issues also formulate the structural problem in a way that reduces to a minimum the number of parameters handled, as well as the number of different types of parameters. For example, Mutsuro Sasaki only altered the local altitude — *i.e.* the z coordinate — of each vertex of a triangulated mesh when he minimised the strain energy of the free-curved surface concrete shell for the crematorium in Gifu (Sasaki·2007, page 81);

(5) Laurent Ney, like others, had a similar concern around the design of the steel bowstring arch of the Nijmegen City Bridge (Ney/…·2010a, page 166): he worked on the topology of the arch with the aim of minimising the number of geometric parameters describing it, which then greatly simplified the control of the variation of these parameters for the weight minimisation process.

In summary, all these examples support the fact that the use of as few parameters as possible — provided that they are crucial to the global definition of the structure — is a guarantee of better control of the design process.

extensive use of graphical methods and geometry · The fourth technique highlighted concerns the extensive use of sketches, drawings, graphical methods and geometry. Although it appears that these are being used less and less in contemporary practices, they are still an important technique in the designer's toolbox:

> “[Graphical methods] *contribute to intuitive understanding and visualization of behavior. They greatly facilitate all statical operations. In early stages of design, they have significant advantages over numerical methods in their simplicity, speed, transparency, and ability to generate efficient forms for cables, arches, trusses, and other structural devices. They are also the source of most of the mathematical expressions used in structural analysis, and give the same answers.*” (Allen·2009, page xii)

Indeed, graphics and graphical methods confer the following benefits on the design process:

(1) sketches provide speed and synthesis
(2) drawings are the medium of a comprehensive memory
(3) graphics and geometry demonstrate visual expressiveness
(4) graphical methods act as open-box processing
(5) geometric reasoning allows the problem to be simplified
(6) drawings constitute a common ground for engineers and architects.

These six points are developed briefly as follows:

(1) Hand sketches provide speed and synthesis. This is a well-known fact for architects: the pen, like other precious tools, can act as an extension of the mind on paper (Clark/Chalmers·1998). First, sketches enable the rapid representation of an idea in the best possible way — *e.g.* regardless of whether it is abstract or concrete — since the only rules it has to obey are those specifically set by the mind with regard to the purpose at hand. They can then also be altered rapidly by the addition and removal of matter, compared rapidly with other proposals and then forgotten rapidly too.

> *"I don't just like sketching because of nostalgia about the hand. It is really about thinking fast, because you can test all sorts of complex relations very, very quickly. It is just a way* [...] *to think."* (Elizabeth Diller in Krasny·2008, page 45)

(2) Drawings — *i.e.* the precise description of a projected position — also provide an excellent means of keeping track of the design process and its product, at any scale and from any viewpoint. Drawings act as an additional memory for the designer, a memory that backs up every detail as well as every intention:

> *"Drawing is the engineer's language. It translates his thinking with a clarity that ordinary language would not have. The engineer first draws everything that he means to have executed. He fixes and therefore keeps the form sensitive to his ideas, the results of his calculations, all the way to useful traces of his trials and errors. He does not wait either for his calculations to be finished before starting to translate them into graphics."* (Favaro·1879, page xviii)

As opposed to the graphical means mentioned previously, the following ones may support pure engineering calculations:

(3) This is the case for graphics and geometry: they are able to express computations entirely visually. This very convenient feature for the designer can be illustrated with the following observation: it is far easier — *i.e.* faster and more intuitive — to characterise the particular properties of a curve — *e.g.* minimum and maximum points, inflexion points, radii of curvature — if the curve is drawn in a Cartesian coordinate system rather than described by

an algebraic equation. This is linked to the fact that graphics offer a stronger focus on topological properties than algebra. Antonio Favaro spelled all this out very well:

> “*Analysis excels, it is true, in arranging problems in equations, in disengaging, by a series of transformations, the combinations of symbols, which give the key to the question propounded, but its very perfection as a means of research neutralises its efficacy as a means of intellectual culture. Leading to the result by a procedure in some manner mechanical, the mind loses sight of the realities upon which it operates, it advances along a labyrinth of formulae, intent only that it lose not the conducting thread, obliged to be more confiding as the darkness becomes more profound, and nearly always unconscious of the path along which it has travelled. On the other hand,* [...] *it is not rare that the results to which analysis conducts, remain concealed under the generality of algebraic symbols, so far as to appear even with less clearness in the solution than in the enunciation.*
> *Geometry proceeds wholly otherwise; she presents the propositions under a sensible form, she removes the train of auxiliaries which hide them from our view, she puts in evidence the transformations which each problem undergoes, and when the solution appears we now perceive the truth under a form the most simple and the most attractive.*”
> (Favaro·1879, preface page i, english translation by Chalmers·1881, page viii)

(4) Graphical methods, and more specifically graphic statics, are also noteworthy for their ability to perform open-box processing. It means that data is not hidden inside intermediate cryptic algorithms. The geometric construction is simultaneously the resolution process and its own result. Hence, the understanding of its result is equivalent to the understanding of its resolution process. Moreover, successive geometric operations form a whole that is impossible to untangle. Tiny details and large trends have equal weight in the drawing. Consequently, graphics essentially take care of accuracy: no one has to deal with the number of digits after the decimal point in order to check the observable intersection of two lines:

> “[Graphic statics] *gives everyone simple and quick processes, substituting the clever and laborious calculations which our engineers do all the time. These processes also have the valuable advantage of still containing the principle of verification, in such a way that they can, like all graphic methods, leave some doubt about a decimal fraction which does not much matter in this*

kind of application. On the other hand they are free of opportunities for the kind of stupid mistake found in long arithmetical operations and algebraical formulae where nothing speaks to the eyes." (Lévy·1874, page xvi)

This open-box processing is reflected in the way Robert Maillart used graphic statics to compute the line of thrust shown in figure 3. After segmenting half the bridge and transposing its weight onto a straight line (figure 5, vertical line between the two bundles of rays), Maillart drew a first funicular polygon, the junction of the ends of which provide the position of the half-bridge's axis of gravity (figure 5, bottom curve and left bundle of rays). From that, he deduced the orientation of the thrusts passing through the hinge at the abutment. He then drew a second funicular polygon passing through that hinge and the one at the crown (figure 5, top curve and right bundle of rays). This last funicular polygon corresponds to the line of thrusts of the bridge under dead load. Comparing this line of thrust with the line of centroids (figure 3, page 21), he obtained the bending moment distribution and was able to make informed changes to the geometry and thus minimise bending moments.

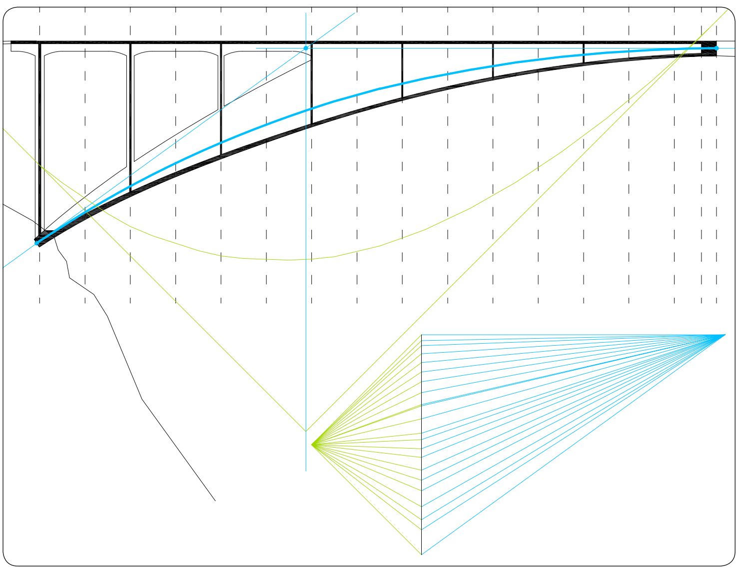


figure 5
description of half of the Salginatobel bridge designed by Robert Maillart; bottom curve and left bundle of rays present the funicular polyline providing the position of the half bridge's axis of gravity; top curve and right bundle of rays construct the line of thrust of the bridge.

Robert Maillart repeated this process three times. Once this had been done, he (or one of his partners) checked the geometry more precisely, but practically no changes had to be made and the bridge has been built as it is. In conclusion, he used graphic statics as a pre-design tool, but this tool was so powerful that it made further analyses completely superfluous. More details about Robert Maillart's drawing methods for the Salginatobel Bridge can be found in Fivet/Zastavni·2012.

(5) Geometric reasoning can also greatly simplify the problem. Not only does the geometric depiction of the problem allow the identification and execution of geometric shortcuts that might shorten the resolution process considerably, but efficient geometric syntheses can also simplify the definition of a structure while increasing and refining its essential capabilities:

> *"Geometry helps manage the multitude of forms and helps us to find new forms. It offers us the opportunity to take hold of a complex problem and then to work on it. Geometry is the basis of parametric design, in which a simple geometric model is adjusted countless times. The geometric study is an eternal promise of a better-adapted object, a potential system which repeatedly culminates in a different optimum solution. In this way, simple geometry can underlie complex and inconceivable forms, such like as a DNA helix."* (Laurent Ney, in Ney/…·2010a, page 39)

(6) Finally, graphics provide common ground for engineers and architects:

> *"The qualitative evaluation of the forces using an inductive process — for example, graphic statics — does not require exact calculation, just practice and experience. This method is understandable to architects too, and offers a good basis for working together.* […] *A common language needs to be learned — an indispensable prerequisite for a close dialogue between the architect and the engineer.* […] *This would be a culture in which the dialogue between architects and structural engineers can begin to grow — a culture that would enable the development of designs in which structural and formal needs merge."* (Joseph Schwartz in Flury/…·2012)

As a result of all these examples, it can be argued that graphics and geometry offer many benefits as regards speed, synthesis, comprehensive memory, visual expressiveness and open-box processing.

from exemplary practices to the definition of a new tool · This sub-section has shed light on four crucial design techniques that give the structural designer all the clarity, speed and interactivity needed to exercise creativity and intuition at their best:

- prior definition of the structural behaviour
- design-oriented use of simplifying assumptions

- problem reduction guaranteeing permanent control
- extensive use of graphical methods and geometry.

The following sub-section uses this inventory to define the main features of the tool being studied in this thesis. Some identified techniques will be implemented by the tool directly; others provide a more global context in which the tool is expected to be used.

04 proposal: a tool to accompany the construction of static equilibriums

defining a new tool · This sub-section outlines a tool whose theoretical foundations will be developed further in this thesis. It starts with a paragraph presenting the expected purposes of the tool. The next three paragraphs explain why the tool uses strut-and-tie models as a general structural abstraction and summarises how this model will be handled by graphic statics and interactive geometry. Two original concepts lie at the heart of the tool's main benefits: graphical regions defining design freedoms and operations altering equilibrium states. Introductory descriptions of these concepts are set out in the final two paragraphs.

general purposes · This thesis theorises a tool aimed at accompanying the definition of structural behaviour. More specifically, it would give the structural designer an opportunity to draw and modify statically equilibrated force paths interactively. The tool focuses on statically equilibrated force paths because they are the most elementary, yet most crucial reduction of the behaviour of almost every structure. Force paths are also directly linked to structural shape and its efficiency as regards stiffness.

"What, then, do engineers see when they imagine or look at a structure ? Broadly, they see patterns of loads which the structure must withstand; and they see load paths which conduct these loads through the structure to the foundations and the earth. The idea of the load path is very powerful, but it is perhaps a more nebulous concept than non-engineers might imagine. Sketches of load paths usually show lines and arrows, yet nothing actually flows." (Addis·1994)

This tool is mainly intended for early design stages — *i.e.* before any use of analysis tools (figure 1, page 4, step b) — but it is also suitable for handling the results of previous analysis (figure 1, step d), being a medium of choice between the architect and the engineer (figure 1, step a and e), or performing certain types of analyses and optimisations (figure 1, step c).

Since this tool would be for preliminary structural explorations, it must be simple to use and so on purpose has limited capabilities: only static equilibriums are managed; one model suits one load case (although this load case can be modified at any time and multiple models can be superimposed); and no kind of deformation — including buckling, seismic responses *etc.* — is taken into account, as least in a direct form. The objective is to offer a complement to classic analysis tools, not to replace them.

Exemplary practices highlighted in the previous sub-section suggest that this tool's main characteristic should be to give the user full control over the design process. Beyond the fact that each structural choice should be dealt with by the user rather than by the tool, it implies that:

- the user must remain in control of the interpretation given to the model; the computerised model should be as abstract as possible; the structural reality that it represents must belong to the user
- the user must remain in control of the chronology being pursued; the tool should not, as far as possible, impose a particular procedure or particular inputs or outputs
- the user must remain in control of the hierarchy he gives to design decisions as well as to structural parts
- the user must be able to interact with the model very quickly, to modify it as freely as possible
- the user must be informed of the consequences of his decisions as effectively as possible; this is the tool's main role.

The success of these objectives would ensure that the tool feeds the structural designer's creativity in an effective way and gives him more control over the structure being shaped, while avoiding being forced to make decisions too early. The following paragraphs explain how the tool would fulfil these objectives and how exemplary practices highlighted in sub-section 03 ("answers: exemplary practices", page 17) guide its definition.

the strut-and-tie model as high-level structural abstraction · First it is proposed that only strut-and-tie models are used. A strut-and-tie network is the skeleton of a structure's behaviour. It is a graphical depiction of the force path inside the structure. And this force path only considers pin-jointed axial stresses — *i.e.* free of bending moments. This model is composed of just four elements: pin joints, point forces applied on nodes, compression rods — *i.e.* struts — and traction rods — *i.e.* ties — linking pin joints together. This very small number of element types offers the great advantage that it is an abstraction of many precisely defined elements. For example:

- forces can be used as representations of applied loads as well as of reactions from the ground, with inner forces describing a load path inside the structure or pretension loads
- pin joints can be used as representations of actual structural hinges but equally they can describe eccentricities of forces acting on the fixed end of a beam
- struts and ties can be used as representations of axial loads in linear structural members, but their eccentricities with the line of centroids of a given beam also provide the bending moments found in this beam.

An example of a strut-and-tie network is given in figure 6 and two possible structural applications of this network are shown in figure 7. The strut-and-tie network is meant to be the abstraction of the real structure, while the actual role of each structural part is only in the designer's head. As a consequence there will always be a certain gap between the actual structure and its representation. This gap allows the designer to define and modify the role of each structural part freely at the precise moment deemed necessary — neither early nor late. Hence, this gap reduces the number of initial inputs required from the user and in a way unlocks the chronology inherent in certain formatted design processes. As long as the structure can be approached by discretised strut-and-tie networks, the way the design process is conducted by the user is therefore not predetermined by the tool.

A strut-and-tie model can be used as a generic abstraction for many types of structures: reticular systems, regardless of whether they are isostatic, indeterminate, pre-stressed, self-stressed — *e.g.* tensegrities — or mechanisms — *i.e.* linkages; depictions of bending moments; lines of thrusts in compression-only structures; and load paths in continuous plastic materials, thanks to

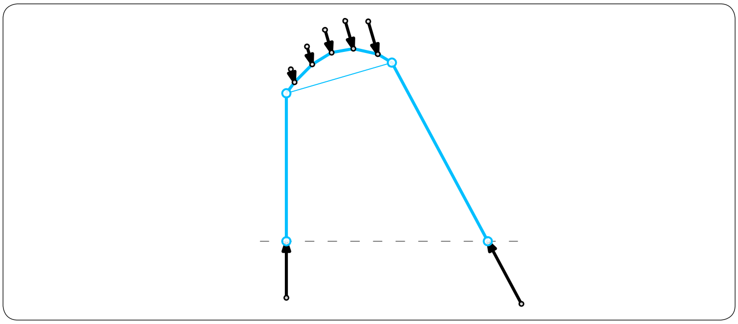


figure 6
Example of a strut-and-tie network.

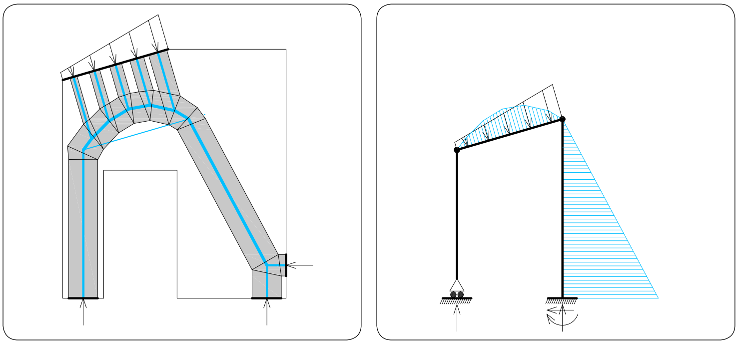


figure 7
Two possible structural applications of the strut-and-tie network shown in figure 6.

the safe theorem of plastic theory — see paragraph entitled "design-oriented use of simplifying assumptions" (page 19), *e.g.* reinforced concrete structures.

This covers a large range of materials — *e.g.* steel (beams and cables), wood, (reinforced) concrete, glass, ceramics and earth — and a large range of applications — *e.g.* roof structures, frameworks, beams, bridges, masonry works, and shear walls.

The major limitation of strut-and-tie models is their inability to address non-discrete representations of force paths. Each model must therefore consist of a finite number of elements. For example, each distributed load must be discretised into a finite number of point forces.

graphic statics · A decision is then taken to represent this strut-and-tie model in a diagram called a "form diagram", along with another diagram called a "force diagram". The role of the force diagram is to express the static equilibrium of the form diagram graphically. For example, the force diagram associated with the strut-and-tie network in figure 6 is shown in figure 8.

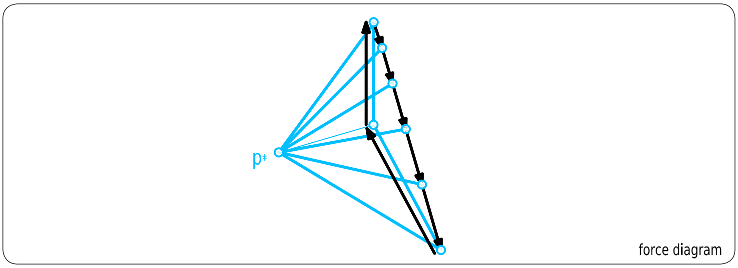


figure 8
Force diagram associated with the strut-and-tie network shown in figure 6.

Both diagrams have the same number of rods. A rod in the form diagram always has a corresponding rod in the force diagram and they both have identical orientations. Distances in the force diagram are measured in units of force magnitudes, *e.g.* Newtons or kilograms, so that the length of a force or a rod in the force diagram is equal to the magnitude of the corresponding force or rod in the form diagram.

In order for a network of struts and ties to be in static equilibrium, two conditions must be ensured: (1) translational equilibrium — *i.e.* the sum of all the forces applied on the network (including those inside the rods) must be zero — and (2) rotational equilibrium — *i.e.* the sum of the moments produced by all the forces applied on this network, with regard to a given point, must be zero. The first condition here is observed simply when, placed end-to-end, the representation of the forces (including those acting inside rods) applied on this network forms a closed polygon in the force diagram (figure 9). Since this first condition is observed for any sub-network, including those containing only one point, the second condition will always be satisfied.

A last rule linking form and force diagrams is finally applied in order to ease the recognition of corresponding force polygons and to guarantee systematically that no rod or force is represented twice in the force diagram. This rule establishes that the forces applied to any sub-network of struts and ties must be read in the form diagram on a cycle (a) that is always read either clockwise or anti-clockwise, and (b) that is identical to the order described by the corresponding closed polygon in the force diagram (figure 9).

Some properties of form and force diagrams are easier to explain if considered in the light of projective geometry — *i.e.* geometry in which two parallels always intersect at a point at infinity. For example, if point p* in figure 8 is placed at infinity, the rays it joins would be parallel, meaning that the funicular polyline supporting the loads would become a straight rod of infinite

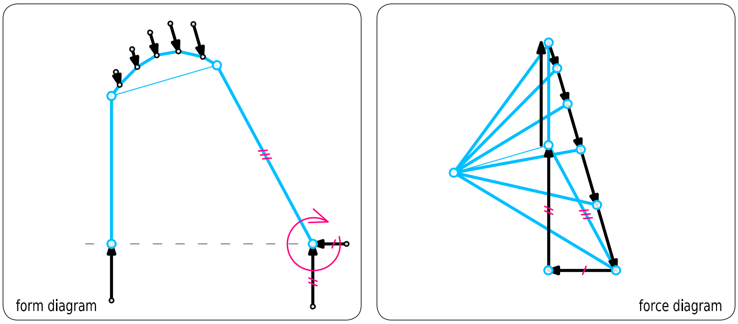
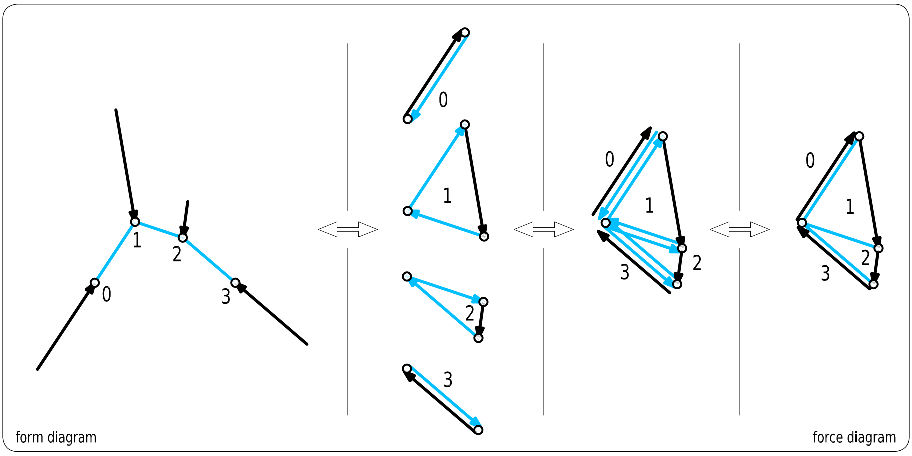


figure 9
three forces that are read clockwise in the form diagram are read in the same order in the force diagram.

magnitude. The choice is made here to focus on Euclidean geometry instead, since (1) this is the geometry with which non-mathematicians have the greatest affinity and (2) infinite magnitudes are never attainable in real life.

The second section of this thesis ("geometric axiomatisation of graphic statics", page 55) will provide a rigorous geometric description of the rules linking the form diagram and the force diagram. These rules can also be summarised with the successive alterations shown in figure 10. Starting from a strut-and-tie network in static equilibrium, the static equilibrium of the four nodes of this form diagram is represented graphically in the first instance by four closed polygons. Forces applied on each node are read clockwise in the form diagram and are represented according to a same order in each closed polygon. Because two forces related to a unique rod must have equal orientation and magnitude, the next step assembles the four closed polygons side by side, matching the pair of forces related to the same rod. Finally, the last step performs a graphical simplification: each pair of forces related to the same rod is replaced by the representation of the rod. The resulting figure is the

figure 10
Successive equivalences between the form diagram and the force diagram.



force diagram of the initial strut-and-tie network. These successive alterations can also be performed starting from the force diagram and building the form diagram.

When forces and rods superpose in the force diagram, each is drawn distinctively so that it is always visually expressed that "*Nothing is lost, nothing is created, everything is transformed*" (Antoine Lavoisier). This way of describing diagrams is not widespread in the literature. Authors generally tend to omit forces and rods when they are superimposed in the force diagram.

Form and force diagrams are necessary and adequate containers of graphic statics. Graphic statics include all the methods of structural computations that make use of form and force diagrams. The tool being sought makes use of graphic statics because it has all the benefits that have been highlighted in the paragraph entitled "extensive use of graphical methods and geometry" (page 23) of sub-section 03 ("answers: exemplary practices", page 17).

While, on the one hand, the form diagram depicts the geometry of the strut-and-tie network and, on the other, the force diagram expresses its inner stresses, the simultaneous understanding of both diagrams reveals the commonly hidden properties that govern the inner distribution of forces — *i.e.* their reciprocal influence and their respective role concerning static equilibrium.

interactive geometry · Graphic statics is here performed using a computer. This combination should be a good match both for the user and the computer. On the one hand, computerised graphic statics enhances the precision of the drawing, offers more visual expressiveness to the user, allows him to automate elementary constructions of diagrams and enables him to parameterise diagrams and handle them dynamically. On the other hand, computerised graphic statics simplifies the computational implementation and increases the speed of algorithmic resolutions thanks to geometric reasoning.

In order to take account of comments referred in sub-section 03 ("answers: exemplary practices", page 17), attention should be given to the interactivity of the tool so that the designer's speed of reflexion is not slowed down and his creativity is not hampered.

In order to benefit fully from the intrinsic geometric simplifications provided by graphic statics, both diagrams are expressed in an entirely geometric framework. This means that all the variables are defined geometrically — as positions of points — and controlled using geometric tools. It implies that the set of design freedoms is homogeneous, that their quantity and quality are well known and, hence, that they are mastered more effectively.

The proposal consequently employs drafting analogies such as straightedges and compasses. However, instead of constraining points on intersections of lines (imposed by straightedges) and circles (imposed by compasses), it uses Boolean combinations — *i.e.* unions, intersections and negations — of half-planes (figure 11, left), disc interiors (figure 11, right) and disc exteriors. This feature removes all the limitations of classical drafting, except that it has to remain discrete. An extensive review will be conducted in sub-section 17 ("examples of graphical computations", page 165).

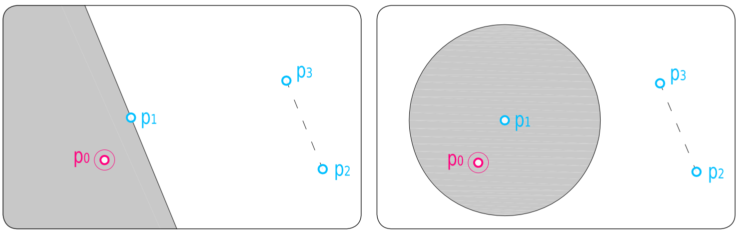


figure 11
(left) point p_0 is constrained on the left of or along the line passing through p_1 according to the direction going from p_2 to p_3; (right) point p_0 is constrained on a distance from p_1 that is smaller than or equal to the distance between p_2 and p_3.

The dynamic handling of these geometric variables is obtained by making points parameters of others. A current serious drawback of classical parameterisation is that the alteration of the hierarchy of dependences between variables cannot be performed as the process goes along and requires the complete reconstruction of the parameterisation. Thanks to the entirely geometric nature of the framework, this limitation can largely be overcome — see sub-section 18 ("switching constraint dependencies", page 195).

graphical regions of design freedoms · As a complementary concept to "geometrically computerised graphic statics" mentioned previously, each variable — *i.e.* each point — is constrained within a graphical region that is constructed so as to be the precise depiction of all the positions it can take and, hence, of all the design freedoms it symbolises.

This concept was originally introduced with the following successive considerations. Given a set of loads that have to be supported by a simply connected strut-and-tie network — *i.e.* a network in which each node joins two rods —, the geometry of the resulting funicular polyline is defined by the relative position of a point called the pole:

(1) if the funicular polyline has to pass through only one given point — p_a — in the form diagram, the pole can be moved anywhere in the force diagram (figure 12); to move the pole rotates the rays in the force diagram and updates the geometry of the funicular polyline in the form diagram

(2) if the funicular polyline has to pass through two given points — p_a and p_b — in the space diagram, the set of available positions of the pole is limited to a line whose fixed position and orientation is a function of the resultant of the given loads and of the two points p_a and p_b (figure 13 and figure 14); this property can be explained in figure 15 — the reaction force of the funicular polyline through p_a (respectively p_b) is divided into two components; the first one is directed towards p_b (respectively p_a) the other one is parallel to the resultant of the applied loads; because the funicular polyline is in equilibrium, the first components must be cancelled out; the magnitudes of the other two components are found by expressing their rotational equilibrium with the resultant, which depends on the position of this resultant; whatever the pole of the funicular polyline is, the magnitude of these two components must remain constant since the funicular polyline always passes through p_a and p_b and since the resultant remains equivalent; their representation in the force diagram consequently constrains the positions of the pole on a single line

(3) if the funicular polyline has to pass through two given points — p_a and p_b — and between two other points — p_c and p_d — in the space diagram, the set of available positions of the pole is limited to a line segment (figure 16); the ends of the segment are found using the intermediary construction of figure 17 — each segment of a funicular polyline passing through two points (such as p_a and p_b) has the property of pivoting around a constant point — p^* — positioned on the line passing through points p_a and p_b (Mayor·1909, pages 14 and 16; Pirard·1950, pages 54 to 56); the two extreme orientations of the second rod are then those passing through p^* and p_c and through p^* and p_d; once reproduced in the force diagram around p_x, these orientations delimit the extreme positions of the pole — p_y and p_z

(4) if the funicular polyline has to pass through three given points — p_a, p_b and p_c — in the space diagram, the set of available positions of the pole is limited to a single position, which is precisely the intersection of two lines; for instance one given by the condition ensuring that the funicular polyline passes through p_a and p_b, and the other given by the condition ensuring that

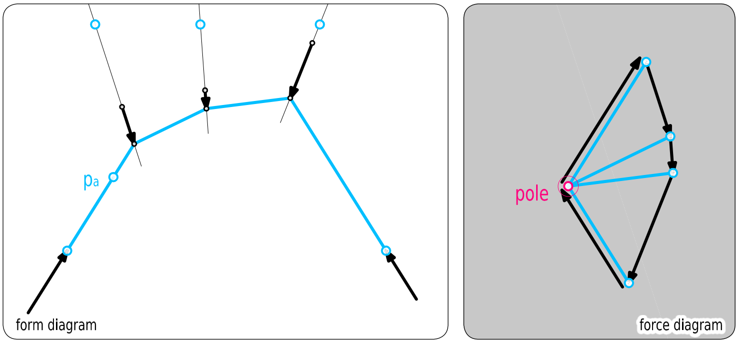


figure 12
the grey area in the force diagram shows the possible positions of the pole if the funicular polyline has to pass through the point p_a.

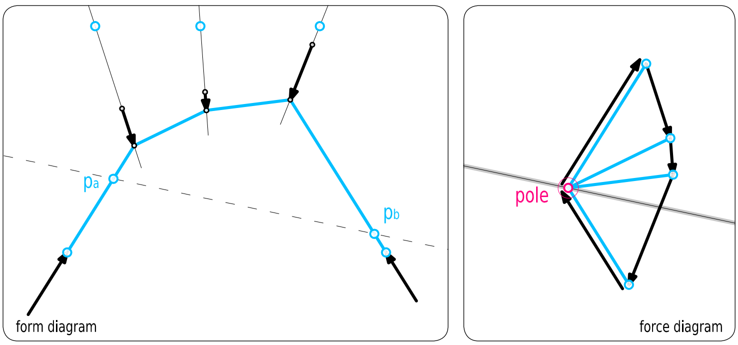


figure 13
the grey line in the force diagram shows the possible positions of the pole if the funicular polyline has to pass through the points p_a and p_b.

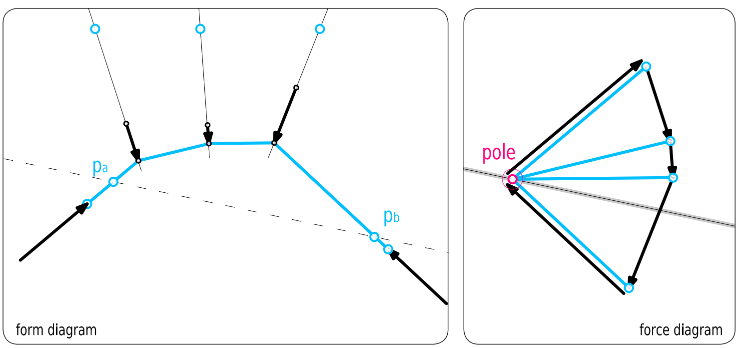


figure 14
the same construction as in figure 13 for another position of the pole.

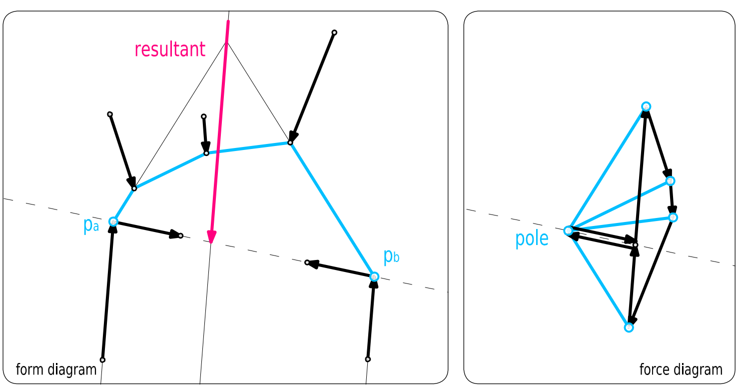


figure 15
explanation of the properties leading to the construction of the line of the poles presented in figure 13 and figure 14.

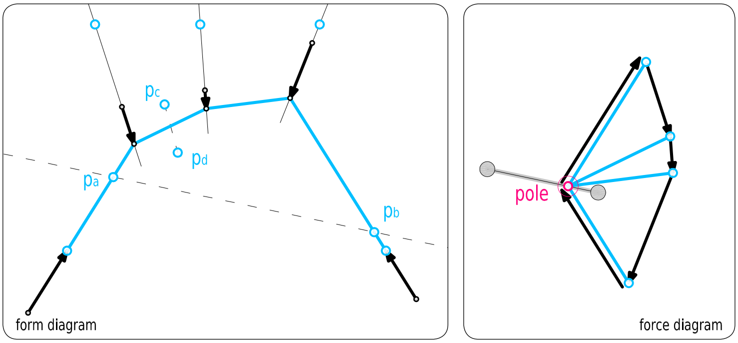


figure 16
the grey segment shows the possible positions of the pole if the funicular polyline has to pass through p_a and p_b and between p_c and p_d.

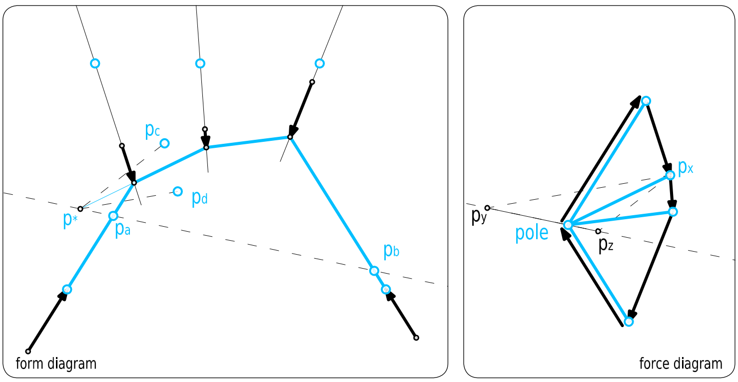


figure 17
explanation of the properties leading to the construction of the line of the pole presented in figure 16.

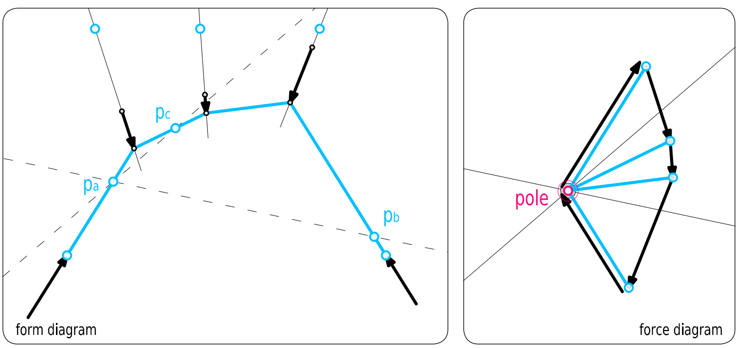


figure 18
the pole in the force diagram must stand on its position if the funicular polyline has to pass through the points p_a, p_b and p_c.

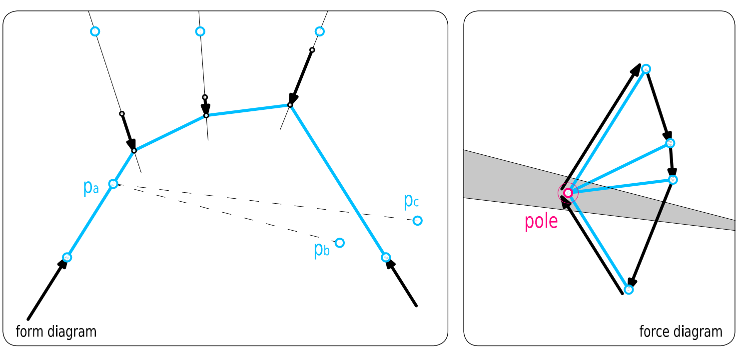


figure 19
the grey area in the force diagram shows the possible positions of the pole if the funicular polyline has to pass through the points p_a and between p_b and p_c.

the funicular polyline passes through p_a and p_c (figure 18). As a result, the pole will also pass through the line given by the condition ensuring that the funicular polyline passes through p_b and p_c

(5) if the funicular polyline has to pass through a given point — p_a — and between two other points — p_b and p_c — in the space diagram, the set of available positions of the pole (figure 19) is limited to an area that can be deduced using the same logic as in figure 15

(6) if the funicular polyline has to pass through a given point — p_a — and has to stay inside a certain shape given by new points, the set of available positions of the pole is harder to find, but is nevertheless still computable and representable in a graphical region (figure 20). The same construction in which the pole is moved to other positions is shown in figure 21 and figure 22.

In conclusion, this pole enjoys different freedoms regarding the conditions imposed on the funicular polyline, and these freedoms are synthesised inside a graphical region. Furthermore, not only does a condition on the funicular polyline constrain the pole, but it also constrains every other point whose position is used to define the construction. Graphical domains of solutions exist for any point and for any strut-and-tie network. For example, figure 23 shows the region in which point p* (defining the upper hole) must stay in order to ensure that the funicular polyline does not cross the hole that point p* defines. This feature is particularly interesting because it removes the common distinction between initial data and results.

Geometric constraints may be imposed manually in the form diagram to limit the spatial extent of the structure or in the force diagram to limit the magnitude or orientation of the forces. Other geometric constraints are created algorithmically (using more general methods than those presented in figure 15 and figure 17) in order to propagate those applied manually. If propagation is automated for every constraint, it is then ascertained that every point has a non-empty region in which it can stay. In other words, it means that there is at least one solution to the geometric construction. This subject is a matter of geometric solvers and will be further developed in the third section of this thesis ("dynamic handling of geometric constraints", page 135).

Thanks to the force diagram, mainly all of the needed quantifiable parameters are expressed by positions of points and are modified by dragging them. Each graphical domain provides all the solutions that a certain parameter can have, given the other applied geometric boundaries.

figure 20
the grey area in the force diagram shows the possible positions of the pole if the funicular polyline has to stay outside the shapes drawn in the form diagram.

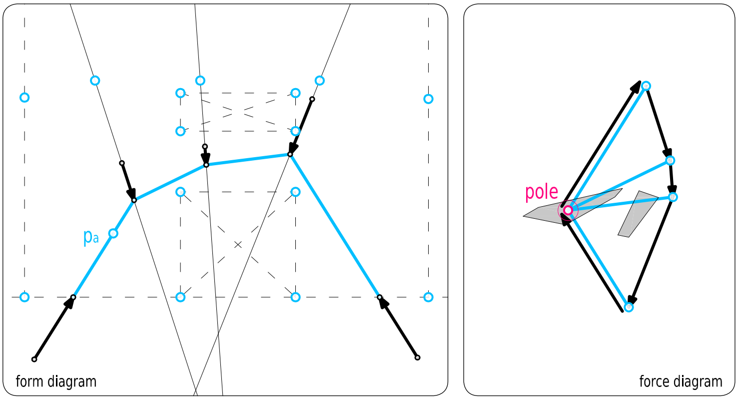


figure 21
reproduction of the construction in figure 20 in which the pole has been moved onto a particular position.

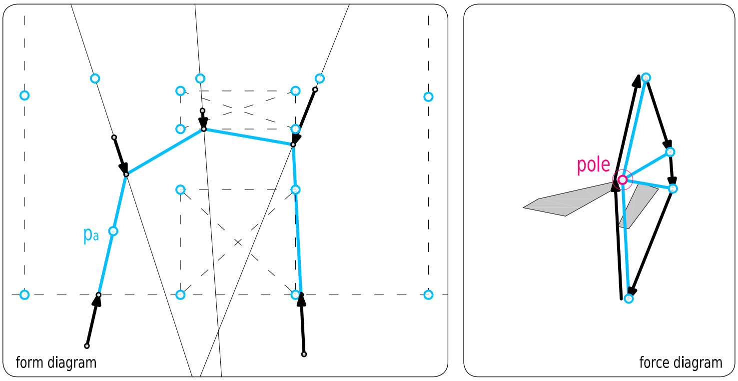
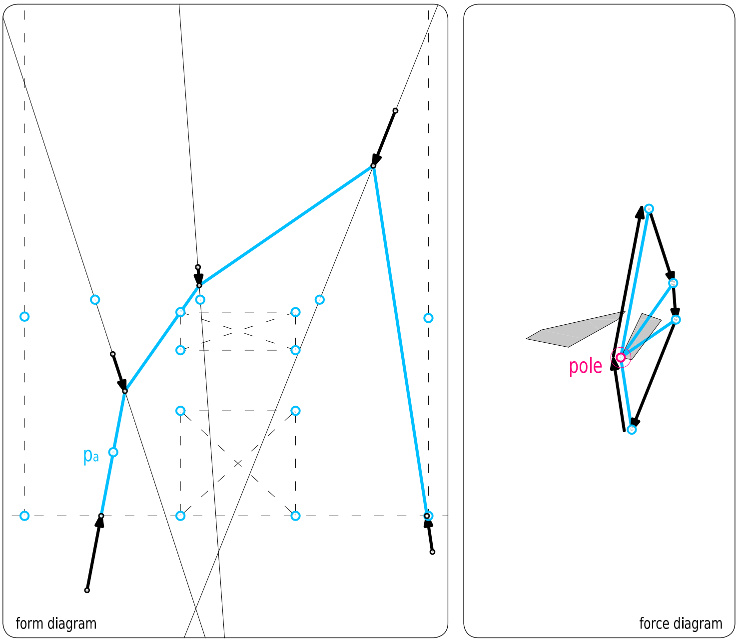


figure 22
reproduction of the construction in figure 20 in which the pole has been moved onto another particular position.



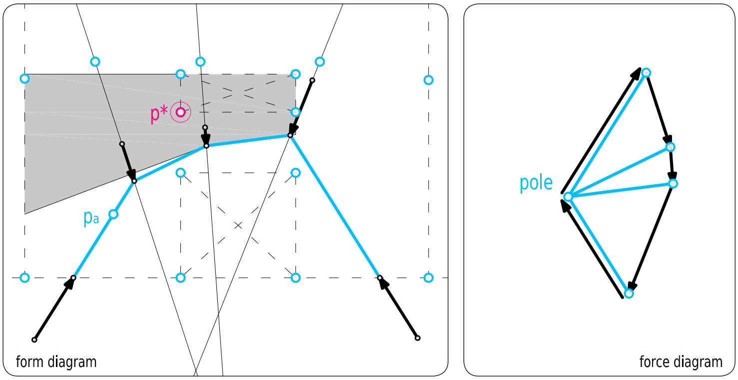


figure 23
reproduction of the construction in figure 20 in which the region in which point p* must stay is highlighted.

operations of equilibrium · The final new concept makes use of geometric operations to assemble different strut-and-tie networks in static equilibrium into a new network in static equilibrium. This concept can be understood by constructing a reticular network of struts and ties on the basis of two funicular polylines.

Starting from the previous funicular polyline (figure 24), the user wants, for example, to retain the geometry of the compression rods but, for one reason or another, has to decrease the magnitudes of these compression rods to a certain value given by the distance between points p_a and p_b. One way of

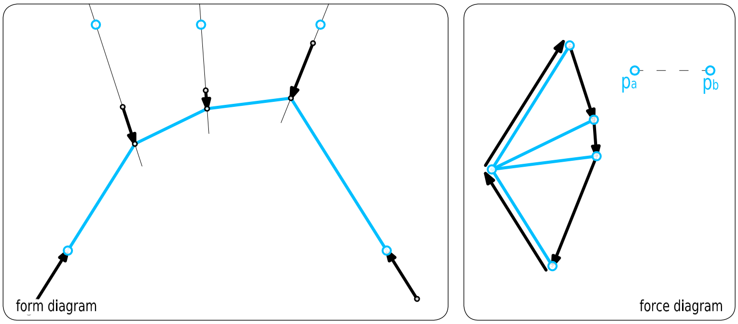


figure 24
initial situation: forces are supported by a funicular polyline but the magnitude of the compression rods are too high.

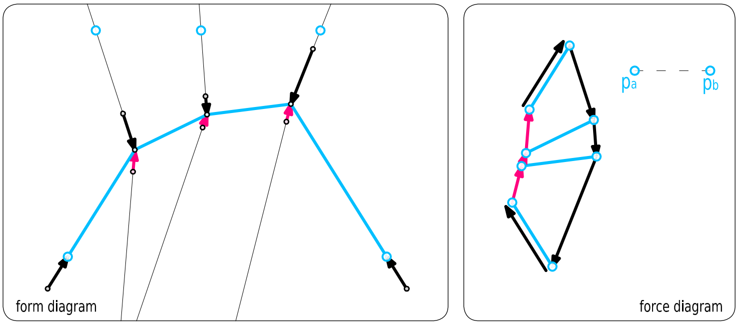


figure 25
new forces are added in order to reduce the magnitude of the compression rods to the value given by the distance between points p_a and p_b.

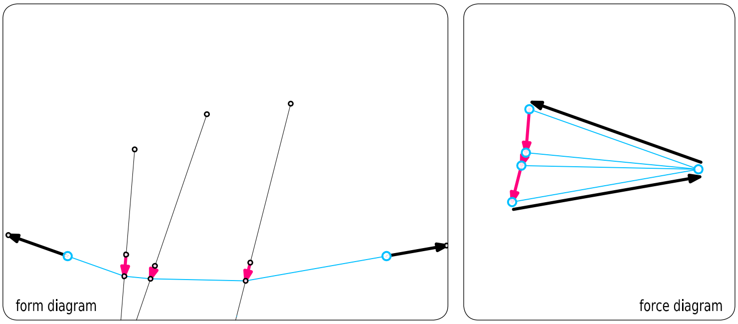


figure 26
a new strut-and-tie network is created to bear the new temporary forces created in figure 25.

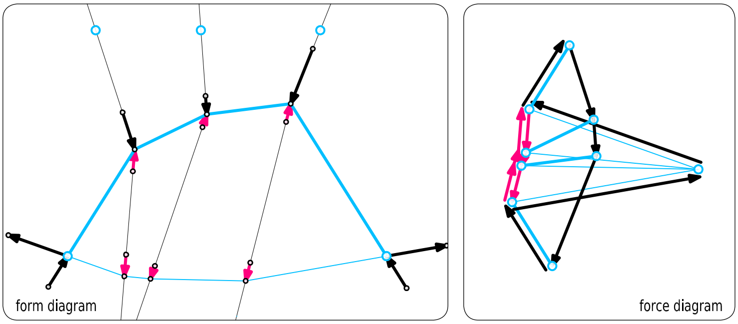


figure 27
diagrams in figure 25 and figure 26 are combined in order to form a new strut-and-tie network.

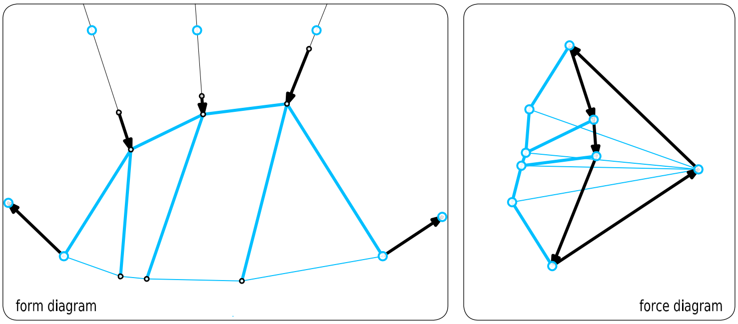


figure 28
opposite forces of equal magnitudes are transformed in rods and reaction forces are summed up.

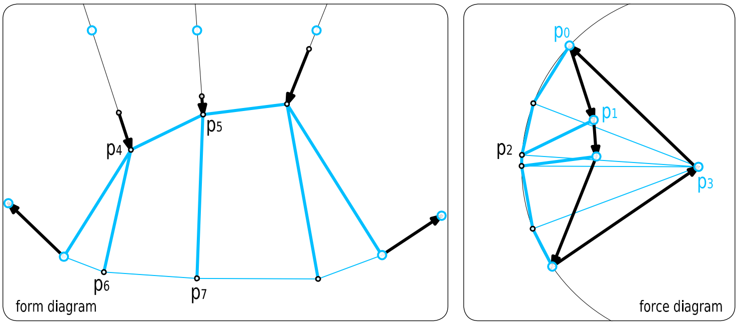


figure 29
points in the force diagram are further constrained on a circle centred in p_3 so that tension rods have constant magnitudes.

doing this, as suggested by the force diagram, is to create new intermediary forces on each loaded point (figure 25). But these new forces cannot be directly supported by the ground, they must themselves be directed to the supports by another network of struts and ties. A possible funicular polyline suited to bearing these temporary forces is illustrated in figure 26: it is made of tension rods and passes through the same supports. The sub-networks in figure 25 and figure 26 are then superimposed in figure 27. The displacement of the pole may be required for that reason. Since each pair of temporary forces are aligned in the form diagram, opposed and of equal magnitude in the force diagram, they act just like a rod. The user can then decide to transform each pair of forces into a new compression rod (figure 28). In addition, the two forces applied on each support can be summed up easily using the force diagram (same figure).

As a result, figure 28 shows a new strut-and-tie network that has been built from two separate funicular polylines. This new network can now be handled just like any other. For instance, the user may want to keep the tension rods constant. This can be done by constraining further the points of the force diagram that define the orientation and amplitude of these rods (figure 29). For example, point p_2 (that was already constrained along a line parallel to p_4p_5 and passing through p_1) would now also be constrained on a circle centred in p_3 whose radius is equal to p_0p_3, the consequence being to update the orientation of the rods p_4p_6, p_5p_7 and p_6p_7. The application of the constraints is controlled by the user; the update of the diagrams is controlled by the tool.

The fourth section of this thesis ("production rules for computer-aided graphic statics", page 265) will show that any operation used to build a strut-and-tie network in static equilibrium can be defined by a procedure that executes sequences of native operations.

The process begins with diagrams that have already been equilibrated and the operations available to the user automatically modify both diagrams using only geometric rules that never put the static equilibrium of the structure at risk. Each operation takes an equilibrated network as input and produces a new equilibrated network. The check for equilibrium is no longer the designer's duty; his job is just to focus on how forces flow through the structure.

05 precedents

The tool introduced in the previous sub-section does not have a full equivalent for comparison purposes. However parts of the concepts and theories it uses have precedents that can be clearly identified. It includes strut-and-tie models and their use with computers, geometric constructions and their use with computers, and graphic statics and its use by and with computers. The following paragraphs provide a recap of these precedents and highlight how the proposed tool exceeds them.

strut-and-tie networks · The explicit use of strut-and-tie networks goes back to the nineteenth century for the structural analysis of pin-jointed trusses — *e.g.* wood and steel frameworks — and suspension bridges (Charlton·2002). Since then, it has been shown that strut-and-tie analogies are also appropriate for the study of structures that present a plastic behaviour (page 20) — *e.g.* for the computation of the line of thrust in masonry (Ungewitter/...·1901) or for the study of the reinforcement required in concrete beams (Marti·1985).

Although the magnitude of forces within strut-and-tie networks can be computed very quickly by graphical means, many contemporary designers still compute them using algebraic trigonometry.

computerised strut-and-tie networks · The last few decades have seen the development of all kinds of software for building and/or analysing strut-and-tie networks, whether as part of multi-purpose structural analysis tools (Autodesk/...·2008, Risa·1987), stand-alone software (Latteur·1998) or plug-ins of geometric modellers (Piker·2010, Preisinger·2013)

Nearly all of them either compute the inner forces from a predetermined shape — *i.e.* the stresses that are displayed are the results of geometries previously drawn — or the other way round, but they offer no opportunity for reversing the deductive approach as it goes along. In other words, these programs impose a design chronology.

geometric constructions · Compass-and-straightedge constructions (figure 30) are as old as the geometry treatises of Ancient Greece (Euclid·2008). This consistent body of knowledge is able to perform many kinds of geometric, arithmetic and trigonometric constructions by hand (Ozanam·1691, Mascheroni·1797, Cousinery·1839, Reuleaux·1899 page 22 and Holme·2010 pages 54 and 422), as far as Euclidean geometry is concerned — *i.e.* the constructions are maintained on one plane and the intersection of two parallels does not exist (Stillwell·2005, Holme·2010).

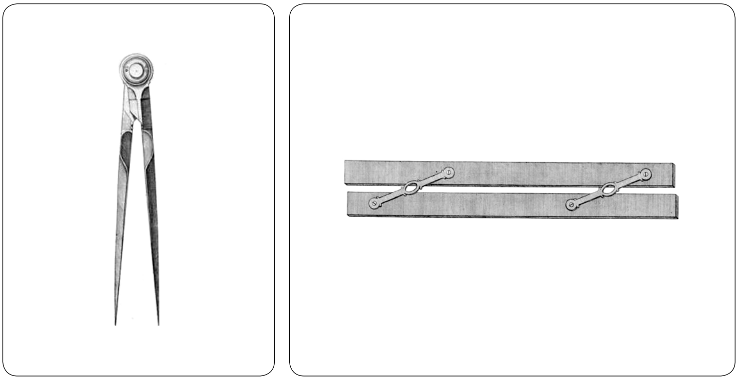


figure 30
a compass and a double straightedge (figures from Penther·1749).

These constructions have the following disadvantages: they may take a long time to produce; they require the user to start from scratch as soon as an initial parameter changes; they require constant accuracy; the drawing can soon become illegible when numerous lines and circles become overlaid. Computers remove these limitations.

In this thesis, consecutive operations of mechanical compasses and straightedges will be replaced by first-order logic (Schöning·2008, Rautenberg·2010, and Makinson·2012). This means that the geometric rules and constructions associated with graphic statics can be robustly expressed in rigorous and logical terms. As such, this approach is informed by the work of *Alfred Tarski* (Tarski·1959, Szczerba·1986, Tarski/Givant·1999 Schwabhäuser/...·2011, Narboux·2007) who axiomatised Euclidean geometry by means of first-order logic. His concern fitted within a broader movement (Peano·1889, Hilbert·1902, Pieri·1908, Birkhoff·1932, Birkhoff/…·1959 and Marchisotto/…·2007) initiated in response to the lack of rigorous demonstrations permitted by Euclid's original axioms.

computerised geometric constructions · Two types of computerised geometric constructions can be distinguished: static constructions and dynamic constructions. The first type is the subject of all computer-aided drafting software. Some of them have recently developed the ability to create dynamic geometric objects, but these are difficult to parameterise and/or have quite limited functionalities.

The second type is tools specifically designed to permit parameterised geometry and its dynamic handling. The best known of these are the ones developed for educational purposes — *e.g. GeoGebra* (Hohenwarter/…·2002 and Hohenwarter/…·2012), *Cinderella* (Richter-Gebert/…·1998, Kortenkamp·1999, Richter-Gebert/…·2012 and Kortenkamp·2013) and *Cabri II Plus* (Laborde/…·2002, Laborde/…·2007). These software applications are built on what literature calls *constraint-based geometric solvers*. This large field of research first appeared in the early 1980s and is still very active today. Reviews on this subject can be found in Dohmen·1995, Hoffmann/…·2005, Rossi/…·2006 and Bettig/…·2011.

They usually offer little support for geometric constructions allowing multiple solutions, graphical inequalities, relative directions, switch of dependencies hierarchy, interdependencies of constraints and union and negations of constraints. If they allow Boolean combinations of inequalities to be drawn and points to be constrained on it — *e.g.* in *Geogebra* —, these inequalities must be defined with algebraic expressions that cannot be dependent on positions of movable points.

This example highlights the main difference between the approach proposed in this thesis and classical geometric solvers. Classical solvers analyse systems of algebraic (in)equations in order to find the solution(s). Here, solutions are obtained by (1) calculating the smallest distance between orthogonal projections on lines or circles, and intersections between lines and circles and (2) checking the membership of points inside Boolean combinations of half-planes, insides of discs and outsides of discs.

Another difference is that, these constraints only take positions as parameters, whereas other solvers deal with constraints and parameters of various kinds (distances, angles, parallelism, tangency, proportionality *etc.*). This does not imply any restriction of application since, for instance, the size of an angle can still be defined by three positions and a tangency can still be defined by successive intersections of lines and circles.

The closest precedent to the approach proposed in this thesis seems to be that of Veltkamp·1995. Indeed, both approaches are logic-based and constructive and solve domains of solutions incrementally in graphical regions. However, the resemblance stops here since the primitive constraints in this thesis are modes of relative directions and graphical inequalities (leading to the possible formulation and handling of infinite domains of solutions).

Constructive constraint-based geometric solvers share many analogies with mechanical linkages (Schooten·1646, Reuleaux·1876, Kempe·1877 and McCarthy/...·2011). Indeed, the most used elementary linkages are nothing more than dynamic strut-and-tie networks. Some constructions developed in this thesis are direct implementations of linkages presented in Yates·1941, Yates·1959 and Artobolevski·1964.

hand-drawn graphic statics · Methods of graphic statics are the practical developments of geometric drawings made by, inter alia, Simon Stevin (Stevin/Girard·1634), Pierre Varignon (Varignon·1725) and Gaspard Monge (Monge·1788) for assessing static equilibrium — read Zastavni·2008a (pages 84 to 86) and Duhem·1905 for deeper review.

According to William John Macquorn Rankine (Rankine·1870), Rankine himself was the first to publish the concept of force polygons — *i.e.* showing "*how to combine in one diagram a system of lines representing the directions and magnitudes of all the forces acting in a given frame*" — in 1856 in a synopsis of lectures he gave at the University of Glasgow. Indeed, the first published diagrams can be found in Rankine·1858 (figure 31). The same year, Earnshaw·1858 (figure 76 plate 4) also published explicit force polygons, but these did not consider trusses. According to Jenkin·1869 (page 441), similar diagrams were used earlier by a certain draughtsman by the name of Taylor and, according to Bow·1873 (page 46), similar diagrams has been published before 1854 in a paper by Mr. C. H. Wild. But there appears to be no remaining publication of these.

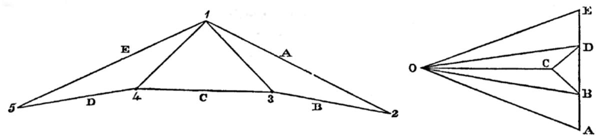


figure 31
diagram of forces of a frame in Rankine·1858, page 143.

The first general definition was expressed in 1858 by William John Macquorn Rankine in his manual of applied mechanics (Rankine·1858, entry 150 page 139). James Clerk Maxwell gave it a geometric background, the same year, in his celebrated paper *"On Reciprocal Figures and Diagrams of Forces"* (Maxwell·1864). Both papers consider pure mathematical reciprocal figures in which forces are not represented in the form diagram (figure 32).

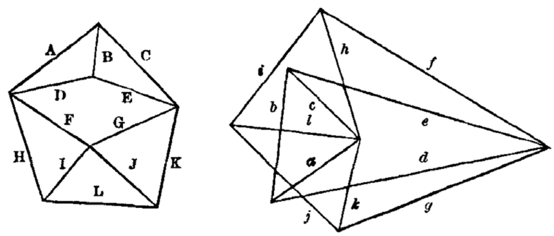


figure 32
reciprocal diagrams by Maxwell·1864, page 253, figure 2.

A couple of years later, Carl Culmann (Culmann·1866) and Luigi Cremona (Cremona·1868) provided the first books entirely devoted to what they called for the first time *"graphic statics"*. Although these books include several new practical methods, they are restricted to the use of form diagrams in which nodes only connect two rods — *i.e.* they do not deal with trusses and other reticular frameworks. Interest in reciprocal figures for the practical computation of forces inside reticular frameworks was only generated subsequently in two papers by Maxwell·1867 and Jenkin·1869 (figure 33).

These contributions seem to be the only published premises of graphic statics. There then followed myriad of books enhancing the range of applications of graphic statics, increasing the number of methods and simplifying them: Bauschinger·1871, Cremona·1872, Bow·1873, Bow·1874, Lévy·1874, DuBois·1875, Eddy·1878, Favaro·1879, Cremona·1885, Lévy·1886, Mohr·1886, Culmann/Ritter·1888, Herrmann·1892, Daubresse·1904, Fairman/...·1932, Pirard·1950 *etc.* Some of them extended graphic statics to the study of spatial frameworks: Rankine·1864, Daubresse·1904, Henneberg·1911, Mayor·1926, Foulon·1969, *etc.*

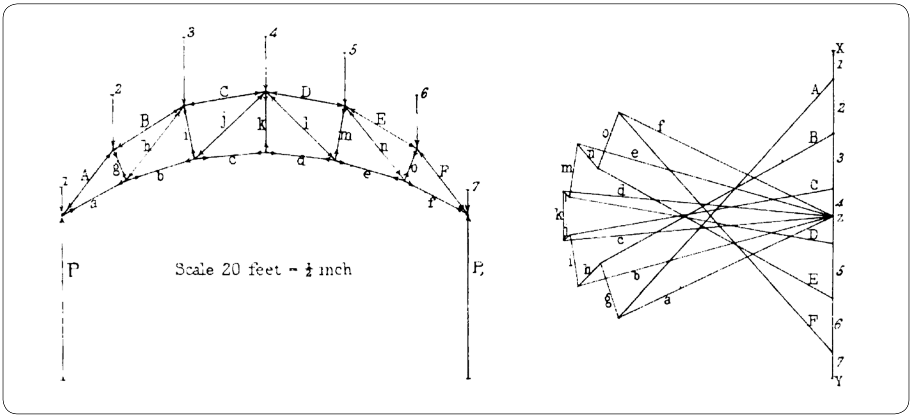


figure 33 uniformly loaded roof by Jenkin·1869, plate XIX, figure 8.

Other historical developments in graphic statics can be found in Jenkin·1869, Maxwell·1876, Chalmers·1881, Scholz·1994, Charlton·2002 (pages 56 to 66) and Zastavni·2008a (pages 84 to 98).

The vast majority of methods presented in this literature is essentially aimed at the analysis of reticular shapes already drawn. Books that use the force diagram as a structural shaping engine are fairly rare. Zalewski·1997 and Allen·2009 are perhaps the greatest counterexamples, but also the most recent. Most accounts on graphic design methods should be found in practice, such as with Robert Maillart for instance.

The main drawback of manual graphic statics is that users build the force diagram out of nothing, line by line until it becomes complete and closed — *i.e.* until the structure is in equilibrium — meaning that the user is only sure of the correctness of his construction at the very end of the process. The process can soon become slow and tedious for complex structures, as was the case reported by Jürg Conzett for the Traversina Bridge design (Mostafavi/…·2003 and personal communication). As for manual geometric constructions, a change of an initial parameter requires the entire drawing to be recomputed.

Graphic statics had its glory days between the 1870s and the 1950s. Different factors brought about its decline, but the main one might be the great developments in numerical methods, encouraged by electronic calculators and, later, by computers. Since it was far easier for electronic engineers to design machines that produce numerical calculus than to design machines capable of drawing, it soon became easier for structural engineers to produce long and complex calculus with calculators than to do them graphically. As a result, graphic statics has now unfortunately been relegated to being an old-fashioned tool solely of educational interest in schools of architecture.

However, computers are nowadays very well suited to graphical design — *e.g.* computers are powerful enough to compute real-time complex graphics and touch-sensitive displays remove the interface between users and their computed drawing like never before — which suggests that the age for computer-aided graphic statics has come or, at least, that it deserves optimistic research.

computer-aided graphic statics · Any research about form and force diagrams that has been conducted in the past decade is related to their use with computers. Two types must be recognised. The first builds on the simplifying properties of reciprocal figures to achieve purposes beyond graphic statics — *e.g.* for the form-finding of masonry vaults (Block/…·2007 and Block·2009) or the shaping of optimal trusses (Beghini·2013, Baker/…·2013).

The second, closer to the purpose of this thesis, follows the objective of making form and force diagrams dynamically modifiable, which would enable the interactive use of a whole host of methods which are just waiting to be re-used. The only current implementations serve pedagogical purposes through didactic examples. These contributions have been made in various forms: ActiveStatics, developed at MIT (Greenwold·2009), is an original web-applet; eQUILIBRIUM, developed at ETHZ (VanMele/Block·2011, VanMele/…·2012), is a web-applet built on the dynamic geometry software GeoGebra; other implementations, as developed in Lachauer/…·2011a and Lachauer/…·2011b, are components of the parametric modelling software Grasshopper for Rhino (Khabazi·2010 and Payne/Issa·2009).

Unfortunately, all these implementations are, for the time being, capable of only dynamic displacements of nodes on preassembled diagrams. In other words, they do not yet allow the interactive construction of graphic statics diagrams.

The user who wants to build custom graphic statics diagrams has to establish beforehand the underlying geometric parameterization of these diagrams. The main issue of this process — which in addition is extraneous to the origi-

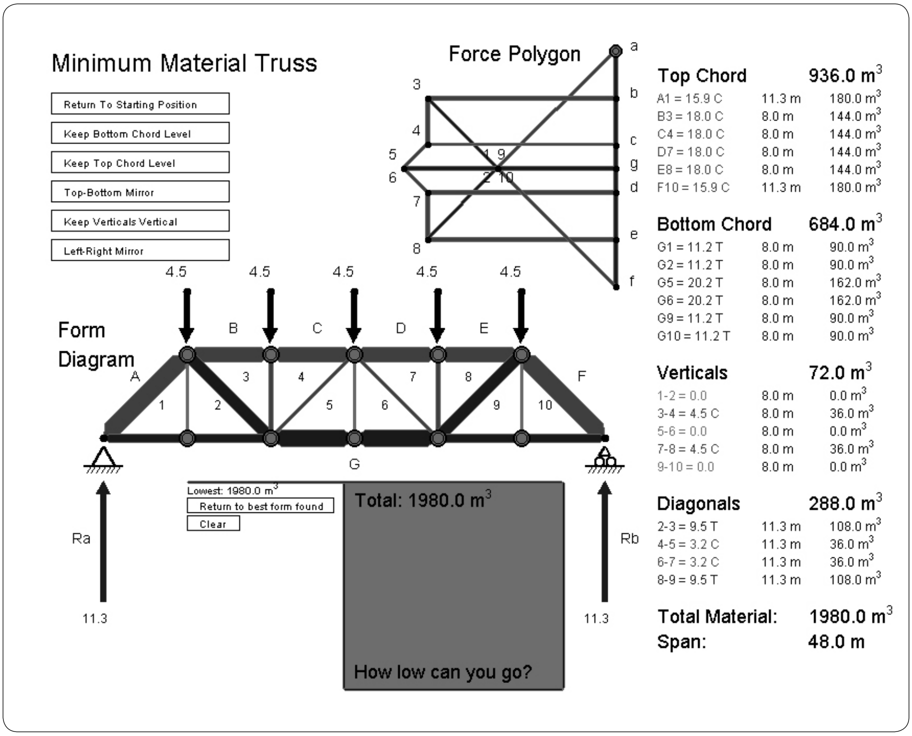


figure 34
minimal material truss, snapshot of ActiveStatic; Greenwold·2009.

nal structural concern — is the same as if the reciprocal diagrams were to be built by hand: the inner geometric properties of the final force diagram are only known and fully understood once it is completed. This generally involves numerous attempts and leads to the dependence of the force diagram on the form diagram, rather than the opposite.

06 organisation of the content

summary of the introduction · This paragraph concludes the introductory section. Sub-section 01 ("fact: contemporary structural design practice", page 3) has provided insight into the shaping of structures: the role of designers, their habits, their tools and their methods for structural design. Sub-section 02 ("critique: the lack of adequate tools for the initial shaping of structures", page 11) has then shown that the common process that gives the dimensions of the structural parts following analysis of a predetermined shape suffers from drawbacks that cannot be overcome by current analysis tools. In response, sub-section 03 ("answers: exemplary practices", page 17) has highlighted past structural design approaches that have produced exemplary structures in the near absence of analysis tools. Inspired by these approaches, sub-section 04 ("proposal: a tool to accompany the construction of static equilibriums", page 29) has defined the purpose and main features of the tool that will be developed in the sections that follows. This tool is aimed at assisting the construction of form and force diagrams interactively and benefits from two original concepts: geometrical domains of solutions and equilibrium operations. Finally, sub-section 05 ("precedents", page 45) looked for precedents, highlighted the theoretical fields it uses, placed them in a historical context and presented some of their current limitations.

organisation of the following sections · The main body of this thesis consists of three sections. The first builds an axiomatic definition of immobile force and form diagrams and of the geometric relations they share. The second uses this axiomatisation to develop the inner working of the tool related to the dynamic displacements of points within these diagrams and to the automated construction of geometric domains of solutions. The third section develops the inner rules of the tool related to the dynamic modifications of diagrams using equilibrium operations.

GEOMETRIC AXIOMATISATION OF GRAPHIC STATICS

This section develops a first-order axiomatisation of graphic statics. More than just a theoretical grounding on which the sections that follow are built, it proposes a precise vocabulary with its own grammar capable of characterising graphic statics diagrams in full.

Sub-section 07 (“positions of points and first-order logic”, page 57) recalls the main connections of Boolean logic. The next sub-sections introduce each fundamental relationship one after the other, together with the axioms defining them:

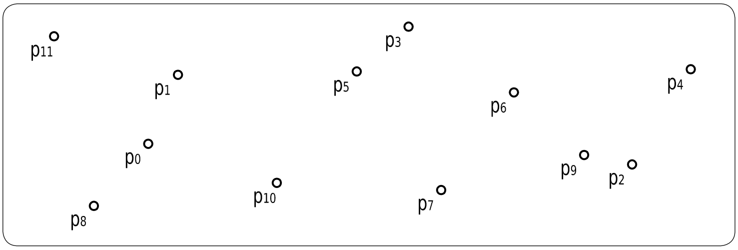
- sub-section 08 (“relationships of proximity and laterality”, page 61) defines one metric and one affine relationship to compare distances and directions
- sub-section 09 (“form diagram and force diagram”, page 77) differentiates the form diagram and the force diagram
- sub-section 10 (“geometrical definition of forces”, page 81) proposes an axiomatisation of the concept of force using only relationships between positions of points

With the help of these fundamental relationships, the concept of rod is defined in sub-section 11 (“rods and other objects”, page 87). The next two sub-sections supplement the set of axioms in order to compel each strut-and-tie network to be in equilibrium within two reciprocal diagrams — see sub-section 12 (“static equilibrium”, page 99) and sub-section 13 (“uniform reading cycle”, page 107). Finally, a summary of all these fundamental relationships and their main axioms is provided in sub-section 14 (“recapitulation”, page 133).

07 positions of points and first-order logic

relationships between positions · Let p_0, p_1, p_2, p_3, … be a set of positions in the plane. Let φ_0, φ_1, φ_2, φ_3, … be a set of geometric relationships that these positions maintain (figure 35). A geometric relationship is either true or false — the law of excluded middle — and a geometric relationship can never be both true and false at the same time — the law of noncontradiction.

figure 35
a cloud of points.



These relationships will be characterised and used with first-order logic (Schöning·2008, Rautenberg·2010 and Makinson·2012). The logical connectives on which first-order logic is based are recalled in the following paragraphs.

implication · A relationship φ_0 implies a relationship φ_1 when φ_1 is true or when φ_0 and φ_1 are simultaneously false. It is written "$\varphi_0 \longrightarrow \varphi_1$", can be read "*if* φ_0 *then* φ_1", and is considered a relationship in itself. The truth table of this logical connective is as follows:

true	⟶ true	*is a true relationship*
false	⟶ true	*is a true relationship*
true	⟶ false	*is a false relationship*
false	⟶ false	*is a true relationship*

equivalence · Two relationships φ_0 and φ_1 are equivalents when φ_1 implies φ_0 and φ_1 implies φ_0. It is written "$\varphi_0 \longleftrightarrow \varphi_1$", can be read "$\varphi_0$ *if and only if* φ_1" and is considered as a relationship in itself. The equivalence of φ_0 and φ_1 does not mean that φ_0 and φ_1 have the same meaning, it just means that they are simultaneously true or false. The truth table of this logical connective is as follows:

true	⟷	true	*is a true relationship*
false	⟷	true	*is a false relationship*
true	⟷	false	*is a false relationship*
false	⟷	false	*is a true relationship*

negation · The negation of a true relationship is false and the negation of a false relationship is true. The negation of a relationship φ_0 is written "$\neg\varphi_0$", is considered as a relationship itself and is defined such that the two following relationships are always observed:

true	⟷	¬false	*is a true relationship*
false	⟷	¬true	*is a true relationship*

conjunction · A logical conjunction, *i.e.* an *intersection*, synonym of "*and*", between two relationships φ_0 and φ_1 is only verified when both φ_0 and φ_1 are true. This logical connective is written "$\varphi_0 \wedge \varphi_1$", is considered as a relationship in itself and is defined by the following truth table:

true	∧	true	*is a true relationship*
true	∧	false	*is a false relationship*
false	∧	true	*is a false relationship*
false	∧	false	*is a false relationship*

disjunction · A logical conjunction, *i.e.* an *union*, synonym of "*or*", between two relationships φ_0 and φ_1 is only verified when either φ_0 or φ_1 is true. This logical connective is written "$\varphi_0 \vee \varphi_1$", is considered a relationship in itself and is defined by the following truth table:

true	∨	true	*is a true relationship*
true	∨	false	*is a true relationship*
false	∨	true	*is a true relationship*
false	∨	false	*is a false relationship*

Boolean logic · Conjunction, disjunction and negation constitute a Boolean algebra (Givant/Halmos·2009). This means that they are ruled so that the following relationships are always true, for any relationships φ_0, φ_1, φ_2 and φ_3:

annihilation:

$$\varphi_0 \wedge \text{false} \longleftrightarrow \text{false}$$
$$\varphi_0 \vee \text{true} \longleftrightarrow \text{true}$$

identity:

$$\varphi_0 \wedge \text{true} \longleftrightarrow \varphi_0$$
$$\varphi_0 \vee \text{false} \longleftrightarrow \varphi_0$$

complementation:

$$\varphi_0 \wedge \neg\varphi_0 \longleftrightarrow \text{false}$$
$$\varphi_0 \vee \neg\varphi_0 \longleftrightarrow \text{true}$$

double negation:

$$\neg\neg\varphi_0 \longleftrightarrow \varphi_0$$

idempotence:

$$\varphi_0 \wedge \varphi_0 \longleftrightarrow \varphi_0$$
$$\varphi_0 \vee \varphi_0 \longleftrightarrow \varphi_0$$

De Morgan:

$$\neg(\varphi_0 \wedge \varphi_1) \longleftrightarrow \neg\varphi_0 \vee \neg\varphi_1$$
$$\neg(\varphi_0 \vee \varphi_1) \longleftrightarrow \neg\varphi_0 \wedge \neg\varphi_1$$

commutativity:

$$\varphi_0 \wedge \varphi_1 \longleftrightarrow \varphi_1 \wedge \varphi_0$$
$$\varphi_0 \vee \varphi_1 \longleftrightarrow \varphi_1 \vee \varphi_0$$

associativity:

$$\varphi_0\wedge(\varphi_1\wedge\varphi_2) \longleftrightarrow (\varphi_0\wedge\varphi_1)\wedge\varphi_2$$
$$\varphi_0\vee(\varphi_1\vee\varphi_2) \longleftrightarrow (\varphi_0\vee\varphi_1)\vee\varphi_2$$

distributivity:

$$\varphi_0\wedge(\varphi_1\vee\varphi_2) \longleftrightarrow (\varphi_0\wedge\varphi_1)\vee(\varphi_0\wedge\varphi_2)$$
$$\varphi_0\vee(\varphi_1\wedge\varphi_2) \longleftrightarrow (\varphi_0\vee\varphi_1)\wedge(\varphi_0\vee\varphi_2)$$

equivalence between implication and conjunction or disjunction :

$$(\varphi_0 \vee \varphi_1) \longleftrightarrow (\neg\varphi_0 \longrightarrow \varphi_1)$$
$$\longleftrightarrow (\neg\varphi_1 \longrightarrow \varphi_0)$$

$$\neg(\varphi_0 \wedge \varphi_1) \longleftrightarrow (\varphi_0 \longrightarrow \neg\varphi_1)$$
$$\longleftrightarrow (\varphi_1 \longrightarrow \neg\varphi_0)$$

if/else equivalences:

$$(\ (\varphi_0\longrightarrow\varphi_1) \wedge (\neg\varphi_0\longrightarrow\varphi_2)\ ) \longleftrightarrow (\ (\varphi_0 \wedge \varphi_1) \vee (\neg\varphi_0 \wedge \varphi_2)\ )$$
$$\neg\quad (\ (\varphi_0\longrightarrow\varphi_1) \wedge (\neg\varphi_0\longrightarrow\varphi_2)\ ) \longleftrightarrow (\ (\varphi_0 \wedge \neg\ \varphi_1) \vee (\neg\varphi_0 \wedge \neg\ \varphi_2)\ )$$

$$\neg\quad (\ (\varphi_0\longrightarrow\varphi_1) \wedge (\neg\varphi_0\longrightarrow\varphi_2)\ ) \longleftrightarrow (\ (\varphi_0\longrightarrow \neg\ \varphi_1) \wedge (\neg\varphi_0 \longrightarrow \neg\ \varphi_2)\ )$$

other equivalences:

$$((\varphi_0 \longrightarrow \varphi_1) \wedge (\varphi_2 \longrightarrow \varphi_3)) \quad \longrightarrow ((\varphi_0 \wedge \varphi_2) \longrightarrow (\varphi_1 \wedge \varphi_3))$$
$$((\varphi_0 \vee \varphi_1) \longrightarrow (\varphi_2 \vee \varphi_3)) \quad \longrightarrow ((\varphi_0 \longrightarrow \varphi_2) \vee (\varphi_1 \longrightarrow \varphi_3))$$
$$((\varphi_0 \wedge \varphi_1) \longleftrightarrow (\varphi_0 \wedge \varphi_2)) \longleftrightarrow (\varphi_0 \longrightarrow (\varphi_1 \longleftrightarrow \varphi_2))$$

existential quantifier · The existential quantifier means that at least one position exists that satisfies the given relationship. If p_0 is that position and φ_0 that relationship, it is written "$\exists p_0 : (\varphi_0)$".

general quantifier · The general quantifier means that a given relationship is verified whatever the indicated position(s). If φ_0 is that relationship and p_0 that position, it is written "$\forall p_0 : (\varphi_0)$".

When no quantifier is specified with a relationship, it is assumed that this position is true whatever the indicated positions of this relationship. Consequently, if the relationship φ_0 does not engage any existential quantifier on p_0, this relationship is equivalent to the relationship "$\forall p_0 : (\varphi_0)$"

identities related to quantifiers · Various identities relating to quantifiers can be identified :

$$\exists p_0 : (\varphi_0) \longleftrightarrow \neg \forall p_0 : (\neg \varphi_0)$$
$$\forall p_0 : (\varphi_0) \longleftrightarrow \neg \exists p_0 : (\neg \varphi_0)$$

$$\forall p_0 : (\forall p_1 : (\varphi_0)) \longleftrightarrow \forall p_1 : (\forall p_0 : (\varphi_0))$$
$$\exists p_0 : (\exists p_1 : (\varphi_0)) \longleftrightarrow \exists p_1 : (\exists p_0 : (\varphi_0))$$

$$\forall p_0 : (\varphi_0) \wedge \forall p_0 : (\varphi_1) \longleftrightarrow \forall p_0 : (\varphi_0 \wedge \varphi_1)$$
$$\forall p_0 : (\varphi_0) \vee \forall p_0 : (\varphi_1) \longleftrightarrow \forall p_0 : (\varphi_0 \vee \varphi_1)$$

The first two equivalences are very interesting since they help avoid the use of all existential quantifiers or, if need be, all general quantifiers.

08 relationships of proximity and laterality

definition of geometric relationships · Having recall the principal notions of mathematical logic, this sub-section introduces the definition of three new fundamental relationships between positions: Proximity, Laterality and UnitDistance. They are called fundamental because they can be admitted as true without being demonstrated. They will constitute the geometrical base of the axiomatisation undertaken in the following sub-sections.

This sub-section also shows how Proximity and Laterality are two fundamental relationships that are sufficient for the axiomatisation of plane elementary Euclidean geometry and how they extend classical compass and straight-edges constructions. Some examples of non-fundamental relationships will ultimately be constructed.

the fundamental relationship of proximity · The relationship of proximity depicts the metric nature of geometry. Proximity[p_0 p_1 p_2 p_3] can be understood as a quaternary predicate that is only verified if the distance from p_0 to p_1 is less than or equal to the distance from p_2 to p_3 (figure 36). In other words, it is only verified if the distance from p_2 to p_3 is greater than or equal to the distance from p_0 to p_1.

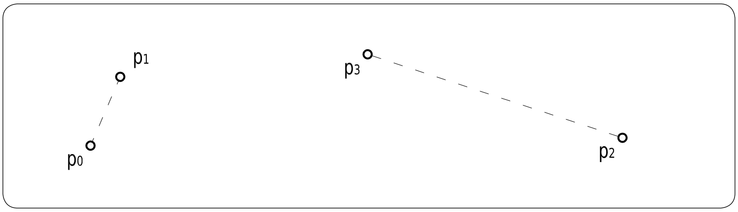


figure 36 four points holding a Proximity[$p_0p_1p_2p_3$] relationship.

The negation ¬Proximity[p_0 p_1 p_2 p_3] says that the distance from p_0 to p_1 is strictly greater than the distance from p_2 to p_3.

axioms related to proximity · This fundamental relationship has numerous properties. The first concerns its symmetry: to exchange the first and second terms or to exchange the third and fourth terms does not alter the existence of this relationship at all. In other words, it means that the following equivalences are always true — whatever the positions p_0, p_1, p_2 and p_3:

Ax.1 — *symmetry of proximity:*
$\text{Proximity}[p_0\ p_1\ p_2\ p_3]$
$\longleftrightarrow \text{Proximity}[p_1\ p_0\ p_2\ p_3]$
$\longleftrightarrow \text{Proximity}[p_0\ p_1\ p_3\ p_2]$
$\longleftrightarrow \text{Proximity}[p_1\ p_0\ p_3\ p_2]$

The Proximity relationship also presents two types of reflexivity. The first is due to equal distances: the distance between two points is always less than or equal to itself. The following relationship (and all its symmetries) is consequently always true:

Ax.2 — *reflexivity of proximity due to equal distances:*
$\text{Proximity}[p_0\ p_1\ p_0\ p_1]$

The second reflexivity is due to the existence of a zero distance: if the first two terms are identical positions, the zero distance separating them is always less than or equal to any two other positions. The following relationship is consequently always verified:

Ax.3 — *reflexivity of proximity due to a zero distance:*
$\text{Proximity}[p_0\ p_0\ p_1\ p_2]$

Because the previous two axioms remain valid when some positions are identical, the following properties are also always verified:

$\text{Proximity}[p_0\ p_0\ p_1\ p_1]$

$\text{Proximity}[p_0\ p_0\ p_0\ p_0]$

A fourth axiom for Proximity relationship can be expressed according to transitivity, *i.e.* the following relationship is always true:

Ax.4 — *transitivity of proximity:*
$(\ \text{Proximity}[p_0\ p_1\ p_2\ p_3] \wedge \text{Proximity}[p_2\ p_3\ p_4\ p_5]\ ) \longrightarrow \text{Proximity}[p_0\ p_1\ p_4\ p_5]$

Finally, the Proximity relationship also develops the following property: if a distance p_0p_1 is strictly greater than a distance p_2p_3, it implies that the distance p_2p_3 is less than or equal to the distance p_0p_1. Strictly speaking, this means that the following relationship is always verified:

Ax.5 — *inclusion of proximity in its negation:*
$\neg$Proximity[p_0 p_1 p_2 p_3] $\longrightarrow$ Proximity[p_2 p_3 p_0 p_1]

the fundamental relationship of laterality · The relationship of laterality depicts the affine nature of geometry. Laterality[p_0 p_1 p_2 p_3] can be understood as a quaternary predicate that is only verified if (1) p_2 is coincident with p_3 (figure 39) or (2) p_2 is not coincident with p_3 but p_0 is on the left of (figure 37) or in line with (figure 38) p_1 according to the direction from p_2 to p_3.

figure 37
four points holding a Laterality[$p_0p_1p_2p_3$] relationship.

figure 38
four points holding a Laterality[$p_0p_1p_2p_3$] relationship.

figure 39
four points holding a Laterality[$p_0p_1p_2p_3$] relationship.

As a consequence, the negation $\neg$Laterality[p_0 p_1 p_2 p_3] informs that (1) p_2 is not coincident with p_3 and (2) p_0 is on the right but not in line with p_1 according to the direction from p_2 to p_3.

axioms related to laterality · This fundamental relationship also has numerous properties. The first concerns its symmetry and is due to the fact that (1) this relation corresponds to a left/right opposition, (2) the second couple of positions correspond to a front/back opposition and (3) a direction going from a first position to a second one is opposed to the direction going from the second to the first one. This means that the following equivalences are always true — whatever the positions p_0, p_1, p_2 and p_3 (figure 40, figure 41, figure 42 and figure 43):

Ax.6 — *symmetry of laterality:*
Laterality[p_0 p_1 p_2 p_3]
⟷ Laterality[p_1 p_0 p_3 p_2]
⟷ Laterality[p_3 p_2 p_0 p_1]
⟷ Laterality[p_2 p_3 p_1 p_0]

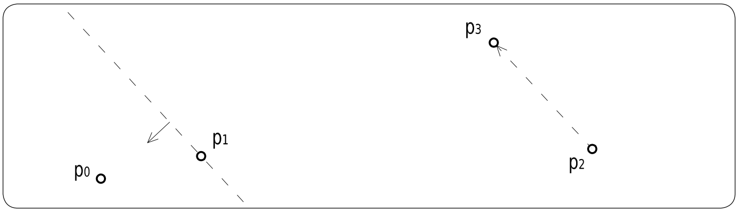


figure 40
four points holding a Laterality[$p_0p_1p_2p_3$] relationship.

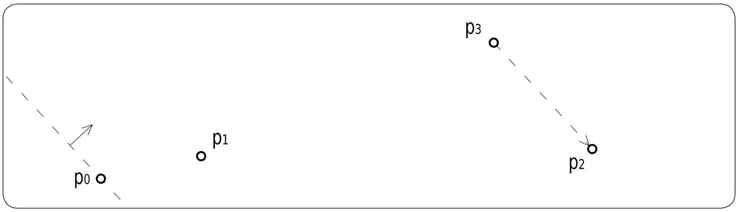


figure 41
four points holding a Laterality[$p_1p_0p_3p_2$] relationship.

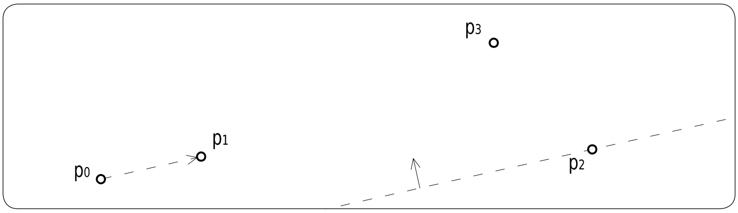


figure 42
four points holding a Laterality[$p_3p_2p_0p_1$] relationship.

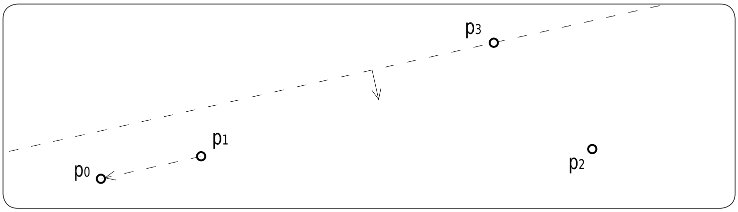


figure 43
four points holding a Laterality[$p_2p_3p_1p_0$] relationship.

The Laterality relationship also presents two types of reflexivity. The first is due to parallelism: a position is always in line with another position according to the direction they give themselves. The following relationship (and all its symmetries) is consequently always true:

Ax.7 — *reflexivity of laterality due to parallelism:*
Laterality[p_0 p_1 p_0 p_1]

The second reflexivity is due to the existence of an undefined direction: if the last two terms are identical positions, the laterality relationship will always be true. The following relationship is consequently always verified:

Ax.8 — *reflexivity of laterality due to an undefined direction:*
Laterality[p_0 p_1 p_2 p_2]

By symmetry, the following relationship is also true:

Laterality[p_0 p_0 p_1 p_2]

Because the previous two axioms remain valid when all the positions are identical, the following properties are also always verified:

Laterality[p_0 p_1 p_1 p_1]

Laterality[p_0 p_0 p_0 p_0]

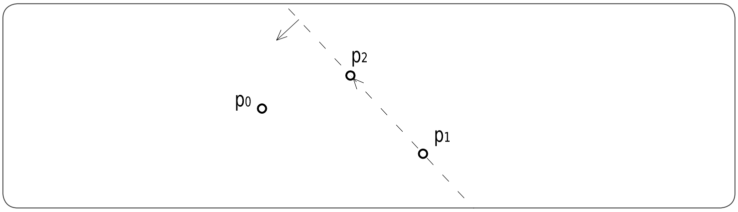
When a position of the first two terms is also a position of the last two terms, the permutation of one of these two positions is allowed, according to the following equivalence (figure 44):

Ax.9 — *permutation of laterality if two equal terms:*
Laterality[p_0 p_1 p_1 p_2] ⟷ Laterality[p_0 p_2 p_1 p_2]

Axioms Ax.6 and Ax.9 consequently give:

Laterality[p_0 p_1 p_1 p_2] ⟷ Laterality[p_0 p_1 p_0 p_2]

figure 44
three points holding the Laterality[$p_0p_1p_1p_2$] and Laterality[$p_0p_2p_1p_2$] relationships.



A fifth axiom of the Laterality relationship can be expressed according to transitivity, *i.e.* the following relationship is always true (figure 45):

Ax.10 — *transitivity of laterality:*

$$(\text{Laterality}[p_0\ p_1\ p_2\ p_3] \wedge \text{Laterality}[p_4\ p_0\ p_2\ p_3]) \longrightarrow \text{Laterality}[p_4\ p_1\ p_2\ p_3]$$

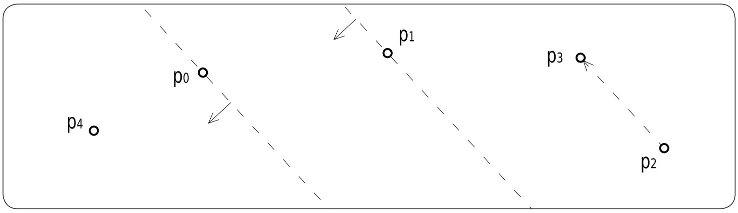


figure 45
five points showing the transitivity of Laterality.

Finally, the Laterality relationship also develops the following property: if a position p_0 is on the right of but not in line with p_1 according to the direction going from p_2 to p_3, it implies that p_0 is on the left or in line with p_1 according to the same direction. Strictly speaking, this means that the following relationship is always verified:

Ax.11 — *inclusion of laterality in its negation:*

$$\neg\text{Laterality}[p_0\ p_1\ p_2\ p_3] \longrightarrow \text{Laterality}[p_0\ p_1\ p_3\ p_2]$$

two sufficient relationships for the axiomatisation of plane elementary Euclidean geometry · The Proximity and Laterality relationships are two sufficient relationships for the axiomatisation of plane elementary Euclidean geometry. This can easily be demonstrated on the basis of Alfred Tarski's first-order axiomatisation. This paragraph provides proof of it.

A geometry is said to be *Euclidean* if it meets the five postulates of Euclid (Euclid·2008, page 7), rewritten here in contemporary language by Holme·2010 (page 253):

(1) *Through two different points there passes one and only one line.*

(2) *If two points on a line are in a plane, then the line lies in the plane.*

(3) *Given two points in a plane. Then there may be drawn a circle with the first point as centre, passing through the second point.*

(4) *All right angles are equal.*

(5) *Given a straight line and a point outside it. Then there is one and only one other line passing through the point which does not intersect the first line.*

This geometry is said to be *elementary* as long as no absolute value is defined, *i.e.* as long as no numerical value is said to be equal to a certain distance between two points. If the fifth postulate — commonly named the parallelism axiom — is omitted, this geometry is no longer Euclidean but *absolute*. If, on the contrary, the fifth postulate is replaced by its inverse, this geometry is non-Euclidean.

A first-order axiomatisation is a system (composed of fundamental relationships, *i.e.* predicates, and axioms) that is "(1) *complete — every assertion is either provable or refutable,* (2) *decidable — there is a mechanical procedure for determining whether or not any given assertion is provable, and* (3) *there is a constructive consistency proof for the theory*" (Tarski/Givant·1999, page 175).

Tarski built a first-order axiomatisation of elementary Euclidean geometry by postulating three fundamental relationships: (1) coincidence (or equality), (2) equidistance (or congruence), and (3) betweenness (Tarski·1959, Szczerba·1986, Tarski/Givant·1999 and Schwabhäuser/...·2011). These three relationships are written here as follows:

(1) Coincidence[p_0 p_1] means that p_0 and p_1 share the same position;

(2) Equidistance[p_0 p_1 p_2 p_3] means that the distance between p_0 and p_1 is equal to the distance between p_2 and p_3;

(3) Betweenness[p_0 p_1 p_2] means that (a) p_0, p_1 and p_2 are collinear and (b) p_1 is between p_0 and p_2, inclusive (figure 46).

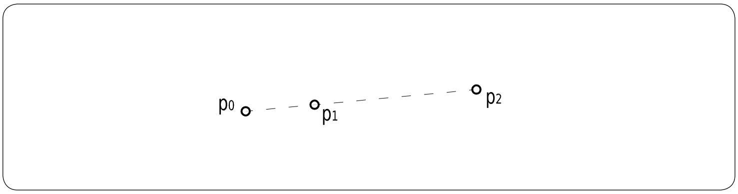


figure 46 three points holding a Betweenness [$p_0p_1p_2$] relationship.

The axiomatic system consists of a set of logical sentences that always has to be verified and that, put together, constitute a sufficient definition of the three fundamental relationships of Coincidence, Equidistance and Betweenness. Many versions of Tarski's axiomatisation have been developed since it was first stated (Tarski/Givant·1999). The one reproduced here is one of the latest version, provided by Haragauri Narayan Gupta (Tarski/Givant·1999, page 190). Given that, unlike Tarski's set of axioms, the coincidence relationship is not seen here as a logical connexion — *i.e.* an equality symbolised by the sign = — but as a fundamental relationship, two axioms ruling its behaviour are added at the beginning of that list. Moreover, Tarski's set of axioms has the particularity of remaining valid for any dimension n. Since this thesis only considers the plane, the two axioms that deal with the lower and upper bounds of this dimension are modified so that the dimension n is 2. The resulting set of axioms is as follows:

T.1 — *reflexivity axiom for coincidence:*
$\text{Coincidence}[p_0\ p_0]$

T.2 — *substitution axiom for coincidence:*
$\text{Coincidence}[p_0\ p_1] \longrightarrow (\varphi_0 \longrightarrow \varphi_1)$
where the relationship φ_1 *is obtained by replacing in any relationship* φ_0 *all the occurrences of* p_0 *by* p_1

T.3 — *inner transitivity for betweenness:*
$\text{Betweenness}[p_0\ p_1\ p_2] \wedge \text{Betweenness}[p_1\ p_3\ p_2] \longrightarrow \text{Betweenness}[p_0\ p_1\ p_3]$

T.4 — *reflexivity axiom for equidistance:*
$\text{Equidistance}[p_0\ p_1\ p_1\ p_0]$

T.5 — *transitivity axiom for equidistance:*
$\text{Equidistance}[p_0\ p_1\ p_2\ p_3] \wedge \text{Equidistance}[p_0\ p_1\ p_4\ p_5] \longrightarrow \text{Equidistance}[p_2\ p_3\ p_4\ p_5]$

T.6 — *identity axiom for equidistance:*
$\text{Equidistance}[p_0\ p_1\ p_2\ p_2] \longrightarrow \text{Coincidence}[p_0\ p_1]$

T.7 — *segment axiom construction:*
$\exists p_0\text{: (Betweenness}[p_1\ p_2\ p_0] \wedge \text{Equidistance}[p_2\ p_0\ p_3\ p_4]\text{)}$

T.8 — *five-segment axiom (similar triangles):*
$(\neg\ \text{Coincidence}[p_0\ p_1]$
$\wedge\ \text{Betweenness}[p_0\ p_1\ p_2] \wedge \text{Betweenness}[p_3\ p_4\ p_5]$
$\wedge\ \text{Equidistance}[p_0\ p_1\ p_3\ p_4] \wedge \text{Equidistance}[p_1\ p_2\ p_4\ p_5]$
$\wedge\ \text{Equidistance}[p_0\ p_6\ p_3\ p_7] \wedge \text{Equidistance}[p_1\ p_6\ p_4\ p_7]\ )$
$\longrightarrow \text{Equidistance}[p_2\ p_6\ p_5\ p_7]$

T.9 — *outer Pasch axiom:*
$\text{Betweenness}[p_0\ p_1\ p_2] \wedge \text{Betweenness}[p_3\ p_2\ p_4]$
$\longrightarrow \exists p_5\text{: (Betweenness}[p_0\ p_5\ p_3] \wedge \text{Betweenness}[p_4\ p_1\ p_5]\text{)}$

T.10 — *lower 2-dimensional axiom:*
$\exists p_0\text{: }\exists p_1\text{: }\exists p_2\text{: (}\neg\text{Betweenness}[p_0\ p_1\ p_2]$
$\wedge\ \neg\text{Betweenness}[p_1\ p_2\ p_0]$
$\wedge\ \neg\text{Betweenness}[p_2\ p_0\ p_1]\text{)}$

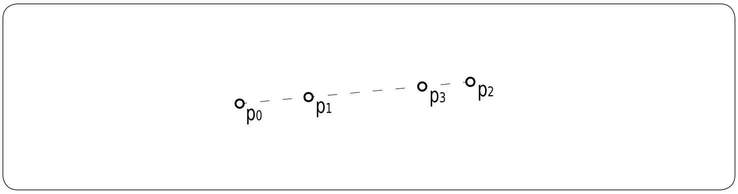


figure 47
axiom T.3,
inner transitivity
for Betweenness.

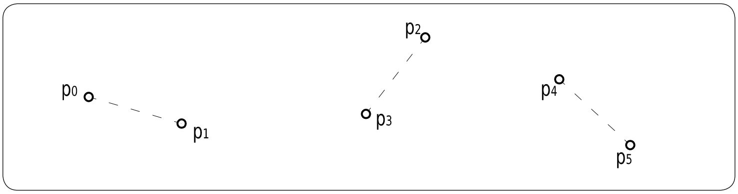


figure 48
axiom T.5,
transitivity for
Equidistance.

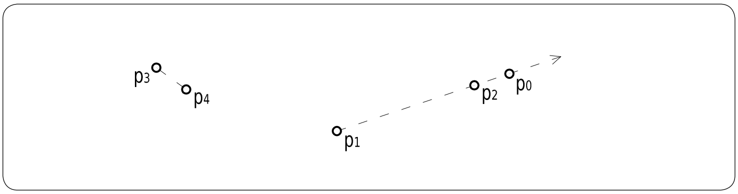


figure 49
axiom T.7,
segment
construction.

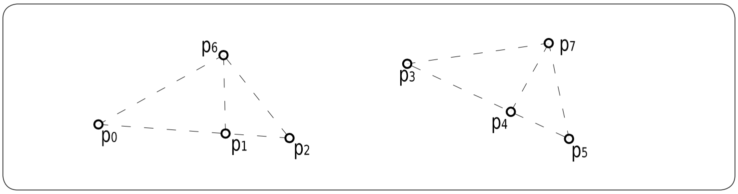


figure 50
axiom T.8,
five segments.

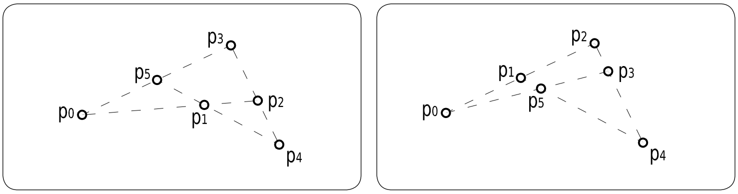


figure 51
axiom T.9,
outer Pasch (two
examples).

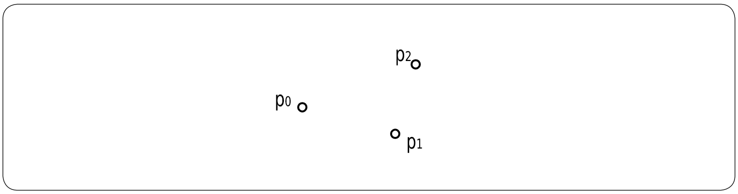


figure 52
axiom T.10,
lower dimension.

T.11 — *upper 2-dimensional axiom:*

$$\begin{aligned} \neg\text{Coincidence}[p_0\ p_1] \ \wedge\ & \text{Equidistance}[p_0\ p_2\ p_1\ p_2] \\ \wedge\ & \text{Equidistance}[p_0\ p_3\ p_1\ p_3] \\ \wedge\ & \text{Equidistance}[p_0\ p_4\ p_1\ p_4] \\ \longrightarrow\ & \text{Betweenness}[p_2\ p_3\ p_4] \\ \vee\ & \text{Betweenness}[p_3\ p_4\ p_2] \\ \vee\ & \text{Betweenness}[p_4\ p_2\ p_3] \end{aligned}$$

T.12 — *Euclid's axiom:*

$$\begin{aligned} &\text{Betweenness}[p_0\ p_1\ p_2] \wedge \text{Betweenness}[p_3\ p_1\ p_4] \wedge \neg\text{Coincidence}[p_0\ p_1] \\ &\quad \longrightarrow \exists p_5\text{: } \exists p_6\text{: } (\ \text{Betweenness}[p_0\ p_3\ p_5] \\ &\qquad\qquad \wedge \text{Betweenness}[p_0\ p_4\ p_6] \\ &\qquad\qquad \wedge \text{Betweenness}[p_6\ p_2\ p_5]\ ) \end{aligned}$$

T.13 — *Axiom schema of continuity:*

$$\begin{aligned} &\exists p_0\text{: } \forall p_1\text{: } \forall p_2\text{: } (\ \varphi_0 \wedge \varphi_1 \longrightarrow \text{Betweenness}[p_0\ p_1\ p_2]\ ) \\ &\quad \longrightarrow \exists p_3\text{: } \forall p_1\text{: } \forall p_2\text{: } (\ \varphi_0 \wedge \varphi_1 \longrightarrow \text{Betweenness}[p_1\ p_3\ p_2]\ ) \end{aligned}$$

where φ_0 *and* φ_1 *first-order formulas so that*
there is no free occurrence of p_0, p_2 *or* p_3 *in* φ_0
and there is no free occurrence of p_0, p_1 *or* p_3 *in* φ_1

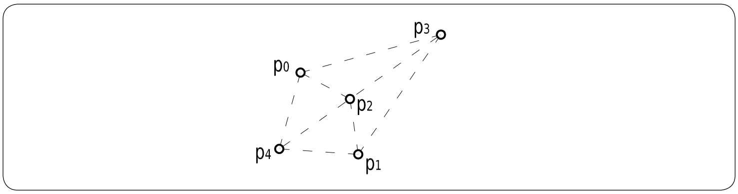


figure 53
axiom T.11,
upper dimension.

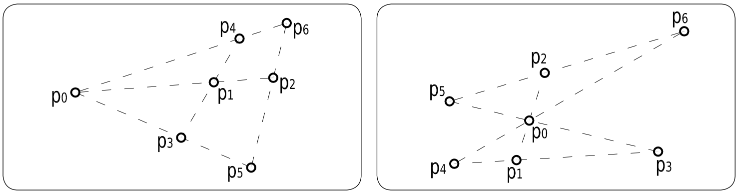


figure 54
axiom T.12,
Euclid's
parallelism
(two examples).

In order to prove that Laterality and Proximity relationships are sufficient to be used as a basis for first-order axiomatisation of plane Euclidean geometry, Tarski's Coincidence, Equidistance and Betweenness relationships must be rewritten in terms of Laterality and Proximity, which can easily be done as follows:

(1) $\text{Coincidence}[p_0\ p_1] :\longleftrightarrow \text{Proximity}[p_0\ p_1\ p_1\ p_1]$

(2) $\text{Equidistance}[p_0\ p_1\ p_2\ p_3] :\longleftrightarrow \text{Proximity}[p_0\ p_1\ p_2\ p_3]$
$\qquad \wedge \text{Proximity}[p_2\ p_3\ p_0\ p_1]$

$$\text{(3a) Betweenness}[p_0\ p_1\ p_2] :\longleftrightarrow \text{Proximity}[p_0\ p_1\ p_0\ p_2] \land \text{Proximity}[p_1\ p_2\ p_0\ p_2] \land \text{Laterality}[p_1\ p_0\ p_0\ p_2] \land \text{Laterality}[p_1\ p_0\ p_2\ p_0]$$

The redefinition of Betweenness can also be performed without using Proximity (figure 55):

$$\text{(3b) Betweenness}[p_0\ p_1\ p_2] :\longleftrightarrow (\text{Coincidence}[p_0\ p_2] \land \text{Coincidence}[p_0\ p_1]) \lor (\text{Laterality}[p_1\ p_0\ p_0\ p_2] \land \text{Laterality}[p_1\ p_2\ p_2\ p_0] \land \exists p_3: (\neg\ \text{Laterality}[p_3\ p_2\ p_2\ p_0] \land \text{Laterality}[p_1\ p_0\ p_3\ p_0] \land \text{Laterality}[p_1\ p_2\ p_2\ p_3]))$$

or without using Laterality (figure 56):

$$\text{(3c) Betweenness}[p_0\ p_1\ p_2] :\longleftrightarrow \forall p_3: (\text{Proximity}[p_3\ p_0\ p_0\ p_1] \land \text{Proximity}[p_3\ p_2\ p_2\ p_1] \longrightarrow \text{Proximity}[p_3\ p_1\ p_3\ p_3])$$

figure 55
redefinition of Betweenness without Proximity.

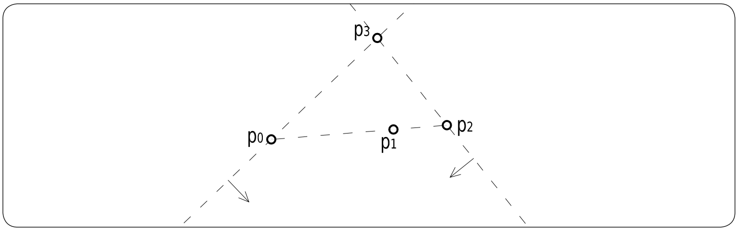
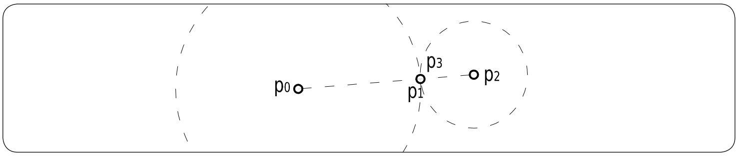


figure 56
redefinition of Betweenness without Laterality.



The existence of these three equivalences concludes the demonstration since it proves that Tarski's entire axiomatisation can be rewritten in terms of Proximity and Laterality. The corollary of this is that Proximity and Laterality are two relationships that are sufficient for describing plane elementary Euclidean geometry.

examples of definitions and demonstrations in plane Euclidean relationships · This statement is illustrated here by defining some non-fundamental relationships — *e.g.* Parallelism, Collinearity, MidPoint, Orthogonality and LineCircleTangency — and by using them to demonstrate a simple theorem of plane geometry.

The first example checks whether two lines are parallel or not. Parallelism[p_0 p_1 p_2 p_3] is proved to be true when the line passing through p_0 and p_1 is parallel to the line passing through p_2 and p_3; if one or both lines have no direction, *i.e.* points p_0 and p_1 are coincident or points p_2 and p_3 are coincident, the relationship remains true. The Parallelism relationship can be defined as follows (figure 57):

Parallelism[p_0 p_1 p_2 p_3] :⟷ Laterality[p_0 p_1 p_2 p_3] ∧ Laterality[p_0 p_1 p_3 p_2]

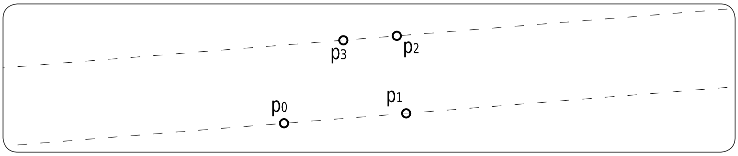


figure 57 four points holding a Parallelism [$p_0p_1p_2p_3$] relationship.

Since this definition remains valid when the parallels are superimposed, three points are collinear if the following statement is verified:

Collinearity[p_0 p_1 p_2] :⟷ Parallelism[p_0 p_1 p_1 p_2]

Furthermore, the existence of two distinct parallels can be defined as follows:

DistinctParallelism[p_0 p_1 p_2 p_3] :⟷ Parallelism[p_0 p_1 p_2 p_3] ∧ ¬Collinearity[p_0 p_2 p_3]

To check whether two points define a real line or not can be done using the following statement:

ValidLine[p_0 p_1] :⟷ ¬Coincidence[p_0 p_1]

The third example uses this Collinearity relationship. It takes three points as parameters and is proved to be true when the second point is the middle of the segment joining the first and the last point (figure 58):

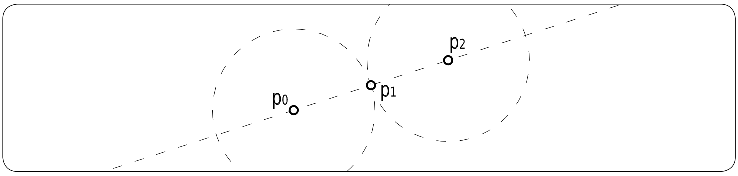


figure 58 three points holding a MidPoint[$p_0p_1p_2$] relationship.

$$\text{MidPoint}[p_0\ p_1\ p_2] :\longleftrightarrow \text{Collinearity}[p_0\ p_1\ p_2] \wedge \text{Equidistance}[p_0\ p_1\ p_1\ p_2]$$

This definition remains valid in the particular case where p_0, p_1 and p_2 are all coincident.

The fourth example checks whether two lines are orthogonal or not. Orthogonality[p_0 p_1 p_2 p_3] is proved to be true when the line passing through p_0 and p_1 is perpendicular to the line passing through p_2 and p_3. If one or both lines have no direction, *i.e.* points p_0 and p_1 are coincident or points p_2 and p_3 are coincident, the relationship is always true. This definition is illustrated in figure 59:

$$\begin{aligned} \text{Orthogonality}[p_0\ p_1\ p_2\ p_3] :\longleftrightarrow \exists p_4 p_5:& \\ & \text{Equidistance}[p_0\ p_4\ p_0\ p_1] \wedge \text{Equidistance}[p_1\ p_4\ p_0\ p_1] \\ \wedge\ & \text{Equidistance}[p_0\ p_5\ p_0\ p_1] \wedge \text{Equidistance}[p_1\ p_5\ p_0\ p_1] \\ \wedge\ & (\neg\text{Coincidence}[p_4\ p_5] \vee \text{Coincidence}[p_0\ p_4]) \\ \wedge\ & \text{Parallelism}[p_2\ p_3\ p_4\ p_5] \end{aligned}$$

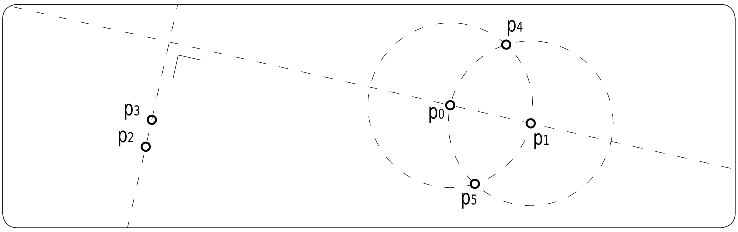


figure 59
four points holding a Orthogonality [$p_0p_1p_2p_3$] relationship.

The last example is written LineCircleTangency[p_0 p_1 p_2 p_3 p_4] and is proved to be true when the line passing through two points p_0 and p_1 and the circle centred on p_2 and passing through p_3 are tangent on a point p_4 (figure 60):

$$\begin{aligned} \text{LineCircleTangency}[p_0\ p_1\ p_2\ p_3\ p_4] :\longleftrightarrow\ & \text{Collinearity}[p_0\ p_1\ p_4] \\ \wedge\ & \text{Equidistance}[p_2\ p_3\ p_2\ p_4] \\ \wedge\ & \text{Orthogonality}[p_0\ p_1\ p_2\ p_4] \end{aligned}$$

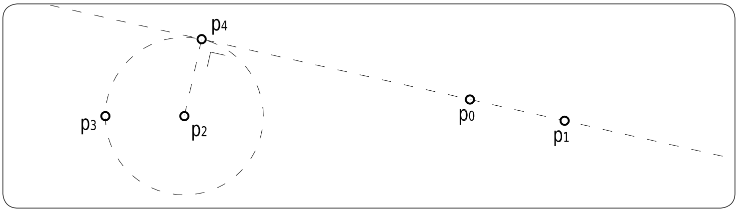


figure 60
five points holding a LineCircleTangency [$p_0p_1p_2p_3p_4$] relationship.

These various definitions can then be employed jointly with the axioms T.1 to T.13 (once they have been rewritten in terms of Laterality and Proximity) in order to prove theorems of plane elementary Euclidean geometry.

As a very simple illustration, the following example proves that two tangents to the same circle are equally distant from the centre of that circle (figure 61). If this circle is centred on point p_0 and passes through p_1, if the first line passes through p_2 and p_3 and is tangent to the circle at point p_4, and if the second line passes through p_5 and p_6 and is tangent to the circle at point p_7, then, it has to be proved that the distances p_0p_4 and p_0p_7 are equal.

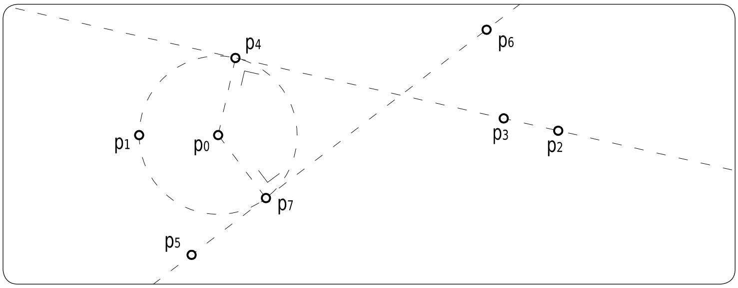


figure 61
demonstration of equal distances between tangencies.

In other words, this means the following implication has to be demonstrated:

$$\begin{aligned}&\text{LineCircleTangency}[p_2\ p_3\ p_0\ p_1\ p_4]\\ \wedge\ &\text{LineCircleTangency}[p_5\ p_6\ p_0\ p_1\ p_7] \longrightarrow \text{Equidistance}[p_0\ p_4\ p_0\ p_7]\end{aligned}$$

Using the definition of the LineCircleTangency relationship, the previous implication is equivalent to the following one:

$$\begin{aligned}&\text{Collinearity}[p_2\ p_3\ p_4] \wedge \text{Equidistance}[p_0\ p_1\ p_0\ p_4] \wedge \text{Orthogonality}[p_2\ p_3\ p_0\ p_4]\\ \wedge\ &\text{Collinearity}[p_5\ p_6\ p_7] \wedge \text{Equidistance}[p_0\ p_1\ p_0\ p_7] \wedge \text{Orthogonality}[p_5\ p_6\ p_0\ p_7]\\ &\longrightarrow \text{Equidistance}[p_0\ p_4\ p_0\ p_7]\end{aligned}$$

Thanks to the axiom T.5 (page 68), it is known that

$$\text{Equidistance}[p_0\ p_1\ p_0\ p_4] \wedge \text{Equidistance}[p_0\ p_1\ p_0\ p_7] \longrightarrow \text{Equidistance}[p_0\ p_4\ p_0\ p_7]$$

is always true — which had to be demonstrated.

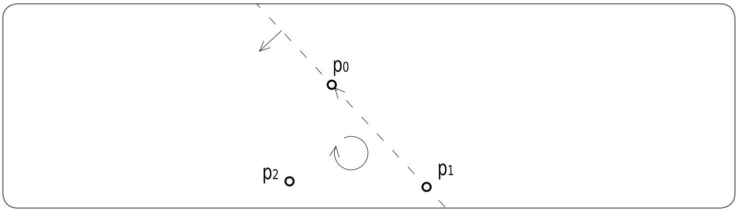
beyond classical compass-and-straightedge constructions · Because the Proximity relationship brings into play the concept of inequality of distances and because the Laterality relationship brings into play the concept of relative direction they allow the precise description of positions of points that do not necessarily lie on circles and lines. A point can be here defined as being inside or outside a circle and as being on the left or on the right of a line. This ability is extraneous to classical compass-and-straightedge constructions.

In the plane, the concept of relative direction relates the front/back opposition with the left/right opposition. For example, if a point moves from one position to another one, the infinite line linking these two positions divides the plane into two distinct parts; the concept of relative direction allows these two parts to be distinguished by defining one on the left and the other on the right of the direction of movement.

The existence of this concept in the definition of the Laterality relationship is of prime importance because it is directly related to the concept of clockwiseness which is omnipresent in graphic statics, *e.g.* for the reading of the forces acting on a point. A ClockWiseness[p_0 p_1 p_2] relationship is verified if the three positions p_0, p_1 and p_2, browsed in this particular order, are clockwise or collinear (figure 62):

ClockWiseness[p_0 p_1 p_2] :⟷ Laterality[p_2 p_1 p_1 p_0]

figure 62 three points holding a ClockWiseness [$p_0p_1p_2$] relationship.



It might be desirable to have a first-order axiomatisation of this new geometry capable of inequalities and relative directions. This axiomatisation — which might be inspired by Tarski's — would allow the rigorous demonstration of any theorem that use Proximity and Laterality relationships. Unfortunately, it is beyond the scope of this thesis and from now on, it will be assumed that it exists without it being explicitly stated.

the fundamental relationship of unit distance · The numerous constructions that will be undertaken in this thesis — see sub-section 17 ("examples of graphical computations", page 165) — assess the need for a new fundamental relationship capable of identifying a unit distance, *i.e.* the need to use metric geometry instead of elementary geometry. This new fundamental relationship is written UnitDistance[p_0 p_1] and is verified if the distance between p_0 and p_1 is equal to a given unit distance. The only axiom that defines this relationship so far is the following one — it prevents the unit distance from being zero:

Ax.12 — *non-zero unit distance:*
UnitDistance[p_0 p_1] ⟶ ¬Proximity[p_0 p_1 p_1 p_1]

Other fundamental relationships describing a transcendental distance between two points — *e.g.* π, e, … — may also be introduced — *e.g.* PiDistance[p_0 p_1], eDistance[p_0 p_1] — since distances of this kind cannot be obtained precisely by geometric constructions. However, geometric constructions exist to approximate most of them to a sufficiently high level of accuracy — equivalent constructions will be given in Sub-section 17 ("examples of graphical computations", page 165).

09 form diagram and force diagram

two fundamental relationships to differentiate two diagrams · Graphic statics establishes geometric constructions on two distinct planes: the form diagram and the force diagram. Two new fundamental unary relationships are therefore introduced to reflect the fact that a point can either belong to the form diagram or to the force diagram. The first is written FormDiagramMembership[p_0] and is proved to be true if p_0 belongs to the form diagram. The second is written ForceDiagramMembership[p_0] and is proved to be true if p_0 belongs to the force diagram.

In order to characterise precisely the role of these two new relationships, five new axioms are added to the existing set. The following paragraphs present these axioms.

axioms guaranteeing the unicity of both diagrams · The next two axioms force a point to belong to at least one of the two diagrams — the law of excluded middle — and at most to only one of the two diagrams — the law of noncontradiction. These two axioms are always true:

Ax.13 — *law of excluded middle for diagram membership:*
FormDiagramMembership[p_0] ∨ ForceDiagramMembership[p_0]

Ax.14 — *law of noncontradiction for diagram membership:*
FormDiagramMembership[p_0] ⟷ ¬ForceDiagramMemberhsip[p_0]

Comparing the membership of two points can then be done by defining the new non-fundamental relationship SameDiagramMembership as follows:

SameDiagramMembership[p_0 p_1] :⟷ (FormDiagramMembership[p_0]
∧ FormDiagramMembership[p_1])
∨ (ForceDiagramMembership[p_0]
∧ ForceDiagramMembership[p_1])

The SameDiagramMembership[p_0 p_1] is consequently true only if p_0 and p_1 belong to the same diagram, regardless of whether it is the form diagram or the force diagram.

three axioms to limit the use of laterality, proximity and unit distance · The points taken as arguments for the Laterality, Proximity and UnitDistance relationships must satisfy some specific rules regarding the diagram in which they are positioned. These rules are encompassed in three axioms.

(1) The first specifies that a Laterality[p_0 p_1 p_2 p_3] relationship only exists if p_0 and p_1 belong to a same diagram and if points p_2 and p_3 also belong to a same diagram, which is the consequence of the fact that a direction does not exist if it is defined by two points that belong to different planes:

Ax.15 — *diagram membership for laterality:*

$$\text{Laterality}[p_0\ p_1\ p_2\ p_3] \longrightarrow (\ \text{SameDiagramMembership}[p_0\ p_1] \land \text{SameDiagramMembership}[p_2\ p_3])$$

This rule does not mean the first two points have to be in the same diagram as the last two points (figure 63). This is another great advantage of the Laterality relationship regarding its use in graphic statics: it allows parallels to be identified in different diagrams.

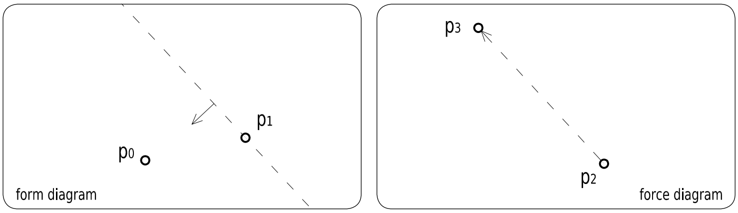


figure 63 four points holding a Laterality [$p_0p_1p_2p_3$] relationships on two different diagrams.

(2) The second new axiom specifies that a Proximity[p_0 p_1 p_2 p_3] relationship only exists if p_0, p_1, p_2 and p_3 belong to a same diagram, which guarantees that any Proximity relationship compares two lengths of an equal unit:

Ax.16 — *diagram membership for proximity:*

$$\text{Proximity}[p_0\ p_1\ p_2\ p_3] \longrightarrow (\ \text{SameDiagramMembership}[p_0\ p_1] \land \text{SameDiagramMembership}[p_1\ p_2] \land \text{SameDiagramMembership}[p_2\ p_3])$$

The existence of this axiom therefore allows form and force diagrams to be of different units of lengths.

(3) The third new axiom specifies that a UnitDistance[p_0 p_1] relationship only exists if both p_0 and p_1 belong to a same diagram:

Ax.17 — *diagram membership for unit distance:*

UnitDistance[p_0 p_1] $\longrightarrow$ SameDiagramMembership[p_0 p_1]

Again, this last axiom does not mean form and force diagrams have to be of same units of lengths.

10 geometrical definition of forces

the fundamental relationship of force · Let F_0, F_1, F_2, ... be a set of forces. This sub-section defines two new fundamental relationships: Force and Equipollence. The first relates the concept of forces with positions of points and the second inform whether two forces are the same.

A force here is meant to be any expression of action that exerts a pull or a push onto a point. A force is commonly defined by:

(a) a point of application
(b) a magnitude
(c) a direction
(d) a type of application (a pull or a push)

To bring the definition of force into line with the purely geometric framework identified so far, these four properties are defined here using four positions of points. The quaternary fundamental relationship Force[F_0 p_0 p_1 p_2 p_3] is said to be true only if there is a force F_0 that:

(a) is applied on the point p_0 lying in the form diagram,
(b) has an equal magnitude to the distance between p_2 and p_3 lying in the force diagram,
(c) has the same direction as the one going from p_2 to p_3, and

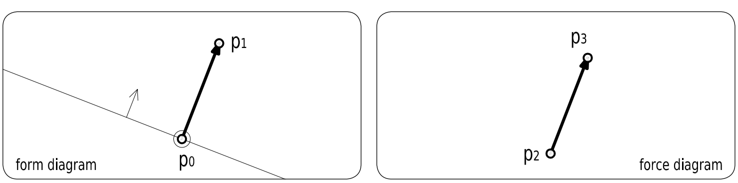


figure 64 four points holding a Force [$f_0p_0p_1p_2p_3$] relationship.

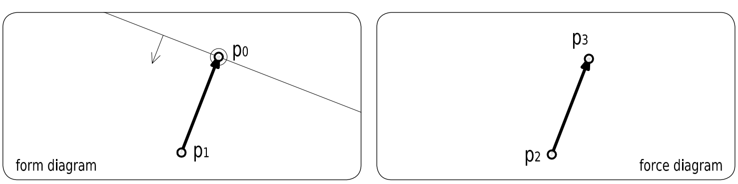


figure 65 four points holding a Force [$f_0p_0p_1p_2p_3$] relationship.

(d) pulls on p_0 if the directions going from p_0 to p_1 and from p_2 to p_3 are equal (figure 64) or pushes on p_0 if these directions are different (figure 65), p_1 being in the form diagram.

As a result, the negation $\neg\mathsf{Force}[F_0\ p_0\ p_1\ p_2\ p_3]$ of this relationship means that such a force does not exist. The $\mathsf{Force}[F_0\ p_0\ p_1\ p_2\ p_3]$ relationship will be defined in more detail by the next three axioms.

an axiom to relate force and diagrams · The concept of force is linked to two different units: on the one hand, a distance between two points of application is measured in units of lengths, *e.g.* metres or feet; on the other hand, the magnitude of a force is measured in units of force, *e.g.* Newtons. Therefore, if the four points p_0, p_1, p_2 and p_3 hold a $\mathsf{Force}[F_0\ p_0\ p_1\ p_2\ p_3]$ relationship, it is expected that p_0 belongs to the form diagram and that both p_2 and p_3 belong to the force diagram. Consequently, the following relationship must be always satisfied:

Ax.18 — *diagram membership for force:*

$$\mathsf{Force}[F_0\ p_0\ p_1\ p_2\ p_3] \longrightarrow \mathsf{FormDiagramMembership}[p_0] \land \mathsf{ForceDiagramMembership}[p_2] \land \mathsf{ForceDiagramMembership}[p_3]$$

an axiom to constrain the point that defines the application type of force · The points p_0, p_2 and p_3 of a $\mathsf{Force}[F_0\ p_0\ p_1\ p_2\ p_3]$ relationship can be placed anywhere within their respective diagram, but this is not the case for p_1. The main role of p_1 is to define the application type of the force F_0, *i.e.* a pull or a push. The need for this distinction is not common practice in literature. It is essential here because it allows the geometric characterisation of the order in which the forces applied on the same point are read — the necessity to characterise this order will be presented in the paragraph entitled "why imposing a uniform reading cycle locally involves a uniform reading cycle globally" (page 114) and the paragraph entitled "the need for uniform reading cycles towards reciprocal diagrams" (page 122).

Since it is proposed that the pull or the push is defined according to whether or not the direction from p_0 to p_1 is equal to the direction from p_2 to p_3, point p_1 should fulfil the following properties:

(1) points p_0 and p_1 should lie in the same diagram in order to define a direction; the following sentence should be verified:

$$\mathsf{SameDiagramMembership}[p_0\ p_1]$$

(2) it is expected that, for the same reason, the direction from p_0 to p_1 is not zero when the magnitude of the force F_0 is not zero, *i.e.* when the direction from p_2 to p_3 exists. The following sentence should be verified:

$$\text{Force}[F_0\ p_0\ p_1\ p_2\ p_3] \wedge \neg\text{Proximity}[p_2\ p_3\ p_2\ p_2] \longrightarrow \neg\text{Proximity}[p_0\ p_1\ p_0\ p_0]$$

(3) in order to avoid any ambiguity of interpretation, it is also advisable that the direction p_0p_1 is zero when the direction p_2p_3 is zero. The following should be true:

$$\text{Force}[F_0\ p_0\ p_1\ p_2\ p_3] \wedge \text{Proximity}[p_2\ p_3\ p_2\ p_2] \longrightarrow \text{Proximity}[p_0\ p_1\ p_0\ p_0]$$

(4) finally, the orientations p_0p_1 and p_2p_3 should be equal in order to express opposite or equal directions clearly. The following should consequently be satisfied:

$$\text{Force}[F_0\ p_0\ p_1\ p_2\ p_3] \longrightarrow \text{Laterality}[p_0\ p_1\ p_2\ p_3] \wedge \text{Laterality}[p_0\ p_1\ p_3\ p_2]$$

The addition of these four conditions are summarised in the following axiom which is always expected to be satisfied:

Ax.19 — *type of force application:*

$$\begin{aligned}\text{Force}[F_0\ p_0\ p_1\ p_2\ p_3] \longrightarrow\ & (\ \text{Laterality}[p_0\ p_1\ p_2\ p_3] \wedge \text{Laterality}[p_0\ p_1\ p_3\ p_2] \\ & \quad \wedge \neg\text{Proximity}[p_2\ p_3\ p_2\ p_2] \wedge \neg\text{Proximity}[p_0\ p_1\ p_0\ p_0]\ ) \\ & \vee (\ \text{Proximity}[p_2\ p_3\ p_2\ p_2] \wedge \text{Proximity}[p_0\ p_1\ p_0\ p_0]\ )\end{aligned}$$

an axiom to ascertain the univocal definition of a force · Each force should be defined by only one Force relationship. The following axiom therefore holds:

Ax.20 — *univocal definition of a force:*

$$\begin{aligned}&(\ \text{Force}[F_0\ p_0\ p_1\ p_2\ p_3] \wedge \text{Force}[F_0\ p_4\ p_5\ p_6\ p_7]\ ) \\ &\quad \longrightarrow (\ \text{Proximity}[p_0\ p_4\ p_4\ p_4] \\ &\qquad \wedge \text{Proximity}[p_1\ p_5\ p_5\ p_5] \\ &\qquad \wedge \text{Proximity}[p_2\ p_6\ p_6\ p_6] \\ &\qquad \wedge \text{Proximity}[p_3\ p_6\ p_6\ p_6]\ )\end{aligned}$$

On the contrary, having equivalent parameters is not a sufficient condition for two forces to be equivalent. For instance, the following statement does not ensure that F_0 and F_1 are a same force:

$$\text{Force}[F_0\ p_0\ p_1\ p_2\ p_3] \wedge \text{Force}[F_1\ p_0\ p_1\ p_2\ p_3]$$

The forces F_0 and F_1 could for instance be two distinct forces sharing the same properties.

two axioms to define equipollence · In order to describe whether two forces are distinct or not, the Equipollence relationship is introduced. Two forces F_0 and F_1 are said equipollent if they actually refer to a same force, meaning that $F_0=F_1$. Comparable to the Coincidence relationship (page 68), the Equipollence relationship is defined by the following axioms:

Ax.21 — *reflexivity axiom for equipollence:*
Equipollence[F_0 F_0]

Ax.22 — *substitution axiom for equipollence:*
Equipollence[F_0 F_1] $\longrightarrow$ ($\varphi_0 \longrightarrow \varphi_1$)
where the relationship φ_1 *is obtained by replacing in any relationship* φ_0 *all the occurrences of* F_0 *by* F_1

the half-line bearing the force · The opposition between pushing and pulling forces is related to another concept, the need for which will also be emphasized later — see the paragraph entitled "why a uniform reading cycle imposes the absence of almost any intersection of rods in the space diagram" (page 124). This is the concept of the half-line bearing the force. If a force F_0 is represented by the Force[F_0 p_0 p_1 p_2 p_3] relationship, it is said that this force is borne by a half-line whose orientation is given by p_2 and p_3, whose extremity is given by p_0 and whose direction is such that p_1 is always on it (figure 66).

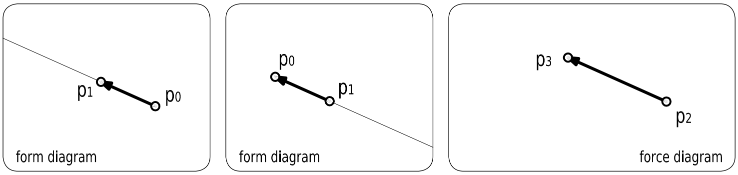


figure 66
two different half-lines bearing a force.

A point p_4 would belong to the half-line bearing the force F_0 represented by Force[F_0 p_0 p_1 p_2 p_3] if the following truth exists (figure 67):

Laterality[p_4 p_0 p_A p_B]
∧ Equidistance[p_0 p_A p_0 p_1] ∧ Equidistance[p_1 p_A p_0 p_1]
∧ Equidistance[p_0 p_B p_0 p_1] ∧ Equidistance[p_1 p_B p_0 p_1]
∧ Laterality[p_A p_0 p_0 p_1] ∧ Laterality[p_B p_0 p_1 p_0]

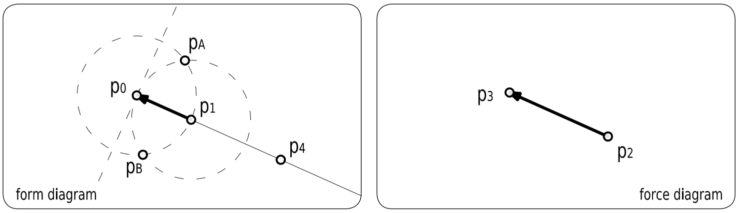


figure 67
four points holding a Force [$f_0p_0p_1p_2p_3$] relationship.

particular forces · Some non-fundamental relationships can be defined in order to characterise particular type of forces: zero forces, pulling forces and pushing forces.

$\mathrm{ZeroForce}[F_0\ p_0\ p_1\ p_2\ p_3] :\longleftrightarrow \mathrm{Force}[F_0\ p_0\ p_1\ p_2\ p_3] \wedge \mathrm{Coincidence}[p_2\ p_3]$

$$\begin{aligned}\mathrm{PullingForce}[F_0\ p_0\ p_1\ p_2\ p_3] :\longleftrightarrow \exists p_A,p_B:\ & \mathrm{Force}[F_0\ p_0\ p_1\ p_2\ p_3] \wedge \mathrm{Laterality}[p_1\ p_0\ p_A\ p_B]\\ \wedge\ & \mathrm{Equidistance}[p_2\ p_A\ p_2\ p_3] \wedge \mathrm{Equidistance}[p_3\ p_A\ p_2\ p_3]\\ \wedge\ & \mathrm{Equidistance}[p_2\ p_B\ p_2\ p_3] \wedge \mathrm{Equidistance}[p_3\ p_B\ p_2\ p_3]\\ \wedge\ & \mathrm{Laterality}[p_A\ p_2\ p_2\ p_3] \wedge \mathrm{Laterality}[p_B\ p_2\ p_3\ p_2]\end{aligned}$$

$$\begin{aligned}\mathrm{PushingForce}[F_0\ p_0\ p_1\ p_2\ p_3] :\longleftrightarrow \exists p_A,p_B:\ & \mathrm{Force}[F_0\ p_0\ p_1\ p_2\ p_3] \wedge \mathrm{Laterality}[p_1\ p_0\ p_B\ p_A]\\ \wedge\ & \mathrm{Equidistance}[p_2\ p_A\ p_2\ p_3] \wedge \mathrm{Equidistance}[p_3\ p_A\ p_2\ p_3]\\ \wedge\ & \mathrm{Equidistance}[p_2\ p_B\ p_2\ p_3] \wedge \mathrm{Equidistance}[p_3\ p_B\ p_2\ p_3]\\ \wedge\ & \mathrm{Laterality}[p_A\ p_2\ p_2\ p_3] \wedge \mathrm{Laterality}[p_B\ p_2\ p_3\ p_2]\end{aligned}$$

The following theorems are deduced from these definitions:

$$\begin{aligned}\mathrm{Force}[F_0\ p_0\ p_1\ p_2\ p_3] \longleftrightarrow (&\mathrm{PullingForce}[F_0\ p_0\ p_1\ p_2\ p_3]\\ \vee\ &\mathrm{PushingForce}[F_0\ p_0\ p_1\ p_2\ p_3])\end{aligned}$$

$$\begin{aligned}\mathrm{ZeroForce}[F_0\ p_0\ p_1\ p_2\ p_3] \longleftrightarrow (&\mathrm{PullingForce}[F_0\ p_0\ p_1\ p_2\ p_3]\\ \wedge\ &\mathrm{PushingForce}[F_0\ p_0\ p_1\ p_2\ p_3])\end{aligned}$$

$$\begin{aligned}&(\mathrm{PullingForce}[F_0\ p_0\ p_1\ p_2\ p_3] \wedge \neg\mathrm{ZeroForce}[F_0\ p_0\ p_1\ p_2\ p_3])\\ &\qquad\longleftrightarrow \neg\mathrm{PushingForce}[F_0\ p_0\ p_1\ p_2\ p_3]\end{aligned}$$

force networks · Each set of forces $\{F_0, F_1, F_2, \ldots, F_n\}$ is said to be a "*force network*" if, at the same time, no point is a point of application of a force belonging to this set and a point of application of a force not belonging to this set. On the basis of this definition, it can be deduced that:

- for any point p_0, if the two relationships $\mathrm{Force}[F_0\ p_0\ p_1\ p_2\ p_3]$ and $\mathrm{Force}[F_1\ p_0\ p_4\ p_5\ p_6]$ are verified, then the two forces F_0 and F_1 belong to the same force network
- the set of all forces is a network of forces

Some further definitions: a force network is said to be a "*sub-network*" of another (different) force network if all the forces inside the former network are also within the latter one. A force network is said to be "*minimum*" if it contains no sub-network other than itself. Consequently, a force network is minimum if it only contains forces applied on a same unique point. If p_0 is that point, the network is said to be a"force network of p_0".

11 rods and other objects

geometric definition of a rod · A rod is here meant to be a massless rectilinear element that links two points and exerts two forces on these points; these forces are (1) aligned on the axis of the rod, (2) of equal magnitude, (3) of opposite direction and (4) of an identical type of application. A strut is meant to be a rod that pushes on both points that it links. A tie is meant to be a rod that pulls on both points that it links. Hence, the rod is entirely defined by:

- the two points that it links
- the type of force that it exerts on these points
- the magnitude of the force that it exerts.

There is a slight difference between this definition and the one usually found in literature. The rod is indeed traditionally defined as a rectilinear element exerted on by two axial opposing forces (Frey·2005, point 4.6.1, page 62). Here, the rod is equal to two opposing forces — the minimum element is a rod or two forces. This difference will allow interesting conceptual simplifications in the demonstration held in the following sub-sections.

The logical flow of this thesis requires a fully geometric definition of the rod, *i.e.* using only positions of points and Laterality and Proximity relationships. This definition is constructed here on the basis of two forces and height points holding the relationships Force[F_0 p_0 p_1 p_2 p_3] and Force[F_1 p_4 p_5 p_6 p_7] (figure 68).

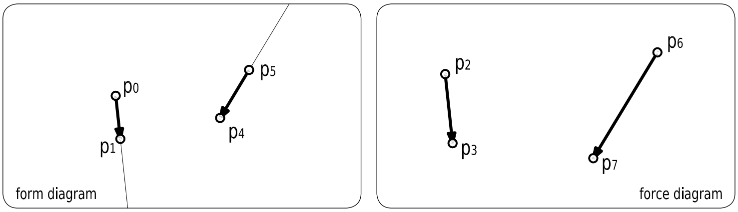


figure 68 two random forces.

In order to form a rod, these eight points must observe the following five geometrical rules.

(1) The two forces must be aligned with the axis of the rod. In other words, the orientations of the forces must be equal to the orientation given by the points of application p_0 and p_4. The following relationship must therefore be satisfied (figure 69):

Parallelism[p_2 p_3 p_0 P_4] ∧ Parallelism[p_6 p_7 p_0 p_4]

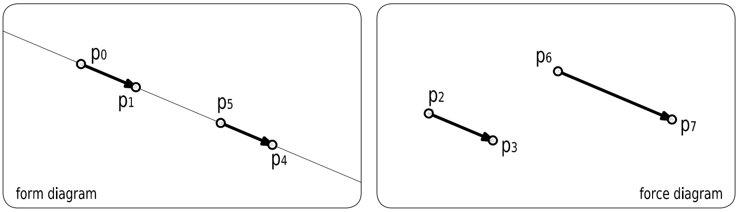


figure 69
application of the first geometric condition leading to a rod.

In view of the above axiom Ax.19 (page 83), points p_1 and p_5 are consequently expected to be in the axis p_0p_4.

(2) Since both force magnitudes must be equal, the following property must also hold (figure 70):

Equidistance[p_6 p_7 p_2 p_3]

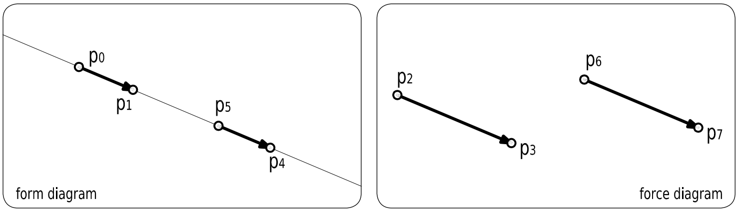


figure 70
application of the second geometric condition leading to a rod.

(3) The two force directions must be opposite. If points p_A and p_B are constructed such that p_A is on the left and p_B on the right of the direction going from p_0 to p_4, the two directions p_2p_3 and p_6p_7 are then distinguished and constrained as follows (figure 71):

(Laterality[p_2 p_3 p_A p_B] ∧ Laterality[p_6 p_7 p_B p_A])
∨ (Laterality[p_2 p_3 p_B p_A] ∧ Laterality[p_6 p_7 p_A p_B])

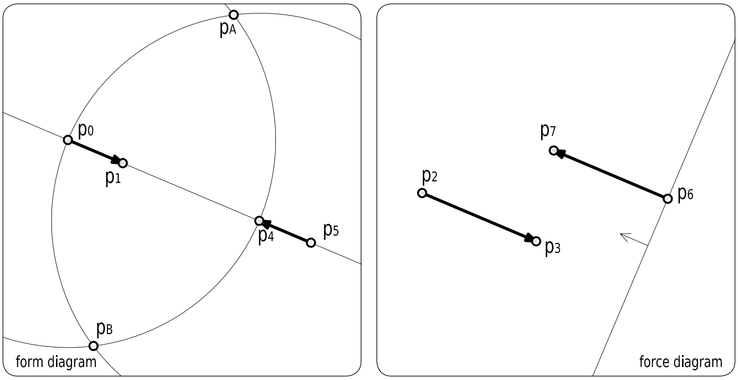


figure 71
application of the third geometric condition leading to a rod.

where p_A and p_B are defined as follows:

Equidistance[p_A p_0 p_0 p_4] ∧ Equidistance[p_A p_4 p_0 p_4]
∧ Equidistance[p_B p_0 p_0 p_4] ∧ Equidistance[p_B p_4 p_0 p_4]
∧ Laterality[p_A p_0 p_0 p_4] ∧ Laterality[p_B p_4 p_4 p_0]

This writing remains consistent when p_2 and p_3 are coincident, since p_6 and p_7 would be coincident as well because of condition (2).

(4) The two forces must be of equal application type. Since the two forces are opposite — condition (3) — this means that points p_1 and p_5 are both either inside or outside the segment p_0p_4. This condition is written using the same points p_A and p_B as in condition (3) (figure 72 and figure 73):

(Laterality[p_0 p_1 p_A p_B] ∧ Laterality[p_4 p_5 p_B p_A])
∨ (Laterality[p_0 p_1 p_B p_A] ∧ Laterality[p_4 p_5 p_A p_B])

As will be seen later in the paragraph entitled “why the definition of rod does not invalidate reciprocal rules” (page 123), only the second case — where the half-lines bearing the forces are inside the segment p_0p_4 — is eligible, which deletes the first line of this geometric rule. Condition (4) is subsequently rewritten as:

Laterality[p_0 p_1 p_B p_A] ∧ Laterality[p_4 p_5 p_A p_B]

(5) The fifth condition is specific to graphic statics. It stems from the desire to depict a rod in the same way in both form and force diagrams, *i.e.* by a line segment. Geometrically, this means that points p_2, p_3, p_6 and p_7 must be aligned.

The necessity of this condition will be explained later in the paragraph entitled “why the definition of rod does not invalidate reciprocal rules” (page 123).

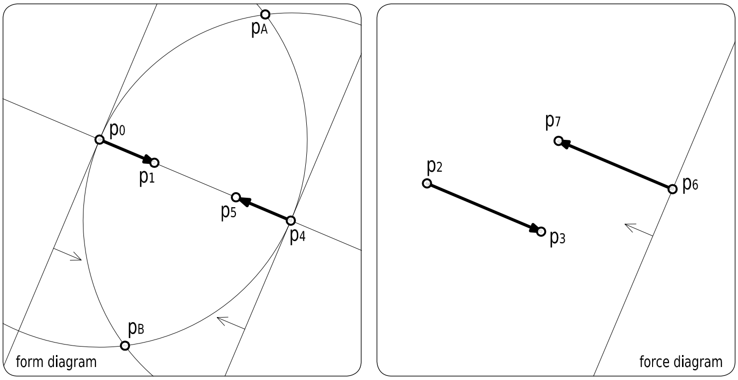


figure 72
application of the fourth geometric condition leading to a rod; first case.

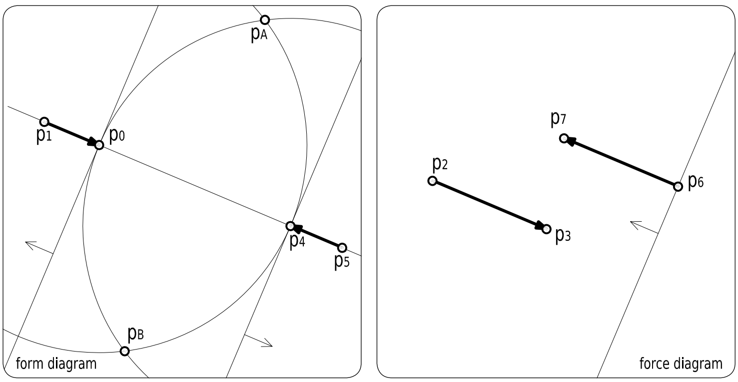


figure 73
application of the fourth geometric condition leading to a rod; second case.

These four points are now so constrained that moving one of them involves moving the two points that bound the other force in the force diagram, *e.g.* moving p_3 leads to the displacement of p_6 and p_7. For practical reasons, it seems logical to make p_3 and p_6 coincident, such that (figure 74):

Proximity[p_6 p_3 p_3 p_3]

figure 74
application of the fourth geometric condition leading to a rod; first case.

Combined with conditions (1), (2) and (3), condition (5) implies that p_2 and p_7 are always coincident:

Proximity[p_7 p_2 p_2 p_2]

As soon as condition (5) is imposed, conditions (2) and (3) become superfluous and condition (1) can be simplified as follows:

(1) $(\text{Laterality}[p_2\ p_3\ p_0\ p_4] \wedge \text{Laterality}[p_2\ p_3\ p_4\ p_0]$
$\wedge\ \text{Laterality}[p_6\ p_7\ p_0\ p_4] \wedge \text{Laterality}[p_6\ p_7\ p_4\ p_0])$
$\longleftrightarrow \text{Laterality}[p_2\ p_3\ p_0\ p_4] \wedge \text{Laterality}[p_2\ p_3\ p_4\ p_0]$

(2) $\text{Proximity}[p_6\ p_7\ p_2\ p_3] \wedge \text{Proximity}[p_2\ p_3\ p_6\ p_7] \longleftrightarrow \text{TRUE}$

(3) $((\text{Laterality}[p_2\ p_3\ p_A\ p_B] \wedge \text{Laterality}[p_6\ p_7\ p_B\ p_A])$
$\vee\ (\text{Laterality}[p_2\ p_3\ p_B\ p_A] \wedge \text{Laterality}[p_6\ p_7\ p_A\ p_B])) \longleftrightarrow \text{TRUE}$

After this simplification and after making p_3 equal to p_6 and p_2 equal to p_7, only two conditions remain: conditions (1) and (4). They define the new non-fundamental relationship $\text{Rod}[F_0\ F_1\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5]$. It is said to be true when the relationships $\text{Force}[F_0\ p_0\ p_1\ p_4\ p_5]$ and $\text{Force}[F_1\ p_2\ p_3\ p_5\ p_4]$ designate two forces F_0 and F_1 that are sufficiently compatible to form a rod. Rigorously, the Rod relationship is defined as follows (figure 75):

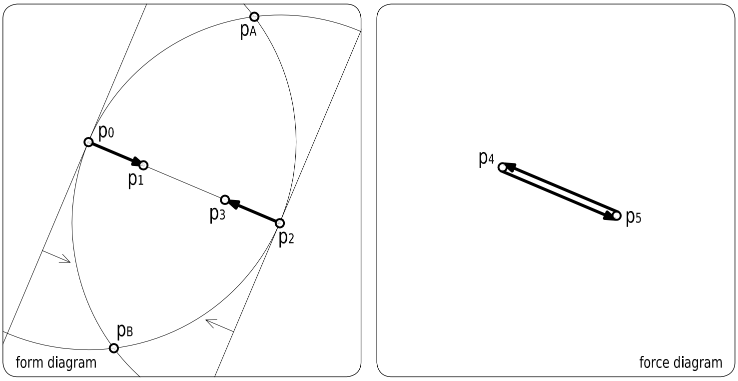
$$\begin{aligned}\text{Rod}[F_0\ F_1\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5] :\longleftrightarrow \exists p_A p_B:&\\ &\text{Force}[F_0\ p_0\ p_1\ p_4\ p_5] \wedge \text{Force}[F_1\ p_2\ p_3\ p_5\ p_4]\\ &\wedge \text{Laterality}[p_4\ p_5\ p_0\ p_2] \wedge \text{Laterality}[p_4\ p_5\ p_2\ p_0]\\ &\wedge \text{Laterality}[p_0\ p_1\ p_B\ p_A] \wedge \text{Laterality}[p_2\ p_3\ p_A\ p_B]\\ &\wedge \text{Equidistance}[p_0\ p_A\ p_0\ p_2] \wedge \text{Equidistance}[p_2\ p_A\ p_0\ p_2] \wedge \text{Laterality}[p_A\ p_0\ p_0\ p_2]\\ &\wedge \text{Equidistance}[p_0\ p_B\ p_0\ p_2] \wedge \text{Equidistance}[p_2\ p_B\ p_0\ p_2] \wedge \text{Laterality}[p_B\ p_2\ p_2\ p_0]\end{aligned}$$

Hence, the following theorem always holds:

$$\text{Rod}[F_0\ F_1\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5] \longleftrightarrow \text{Rod}[F_1\ F_0\ p_2\ p_3\ p_0\ p_1\ p_5\ p_4]$$

The important fact about this definition is that the Rod relationship is non-fundamental, *i.e.* it can be defined entirely using fundamental Laterality, Proximity and Force relationships. Since graphic statics does not involve objects other

figure 75
two forces fulfilling all the prerequisite conditions in a way that they can be considered as a rod.



than points, forces and rods, it is here shown that an axiomatisation of graphic statics using only Laterality, Proximity and Force fundamental relationships is possible.

Moreover, as soon as two forces fulfil sufficient geometric conditions to be considered a rod, it makes no difference talking about this rod (figure 76 and figure 77) or about the two forces (figure 75).

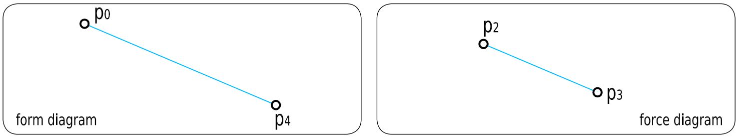


figure 76
a corresponding rod in tension.

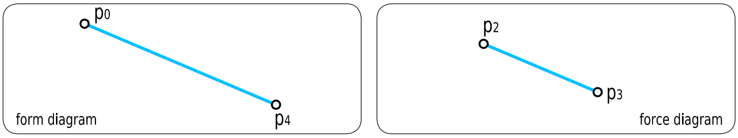


figure 77
another rod, in compression.

For example, the following relationship (figure 78) :

Force[F_0 p_1 p_0 p_{10} p_{11}] ∧ Force[F_1 p_6 p_8 p_{12} p_{10}] ∧ Force[F_2 p_7 p_9 p_{11} p_{12}]

∧ Rod[F_3 F_4 p_1 p_2 p_6 p_4 p_{12} p_{10}]
∧ Rod[F_5 F_6 p_1 p_3 p_7 p_5 p_{11} p_{12}]

∧ Laterality[p_6 p_1 p_{10} p_{12}] ∧ Laterality[p_6 p_1 p_{12} p_{10}]
∧ Laterality[p_7 p_1 p_{11} p_{12}] ∧ Laterality[p_7 p_1 p_{12} p_{11}]

is completely equivalent to this one (figure 79):

Force[F_0 p_1 p_0 p_{10} p_{11}] ∧ Force[F_1 p_6 p_8 p_{12} p_{10}] ∧ Force[F_2 p_7 p_9 p_{11} p_{12}]

∧ Force[F_3 p_1 p_2 p_{12} p_{10}] ∧ Force[F_4 p_6 p_4 p_{10} p_{12}]
∧ Force[F_5 p_1 p_3 p_{11} p_{12}] ∧ Force[F_6 p_7 p_5 p_{12} p_{11}]

∧ Laterality[p_6 p_1 p_{10} p_{12}] ∧ Laterality[p_6 p_1 p_{12} p_{10}]
∧ Laterality[p_7 p_1 p_{11} p_{12}] ∧ Laterality[p_7 p_1 p_{12} p_{11}]

figure 78
two rods and three forces.

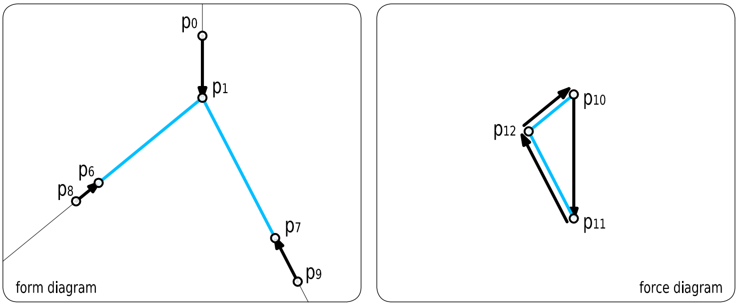
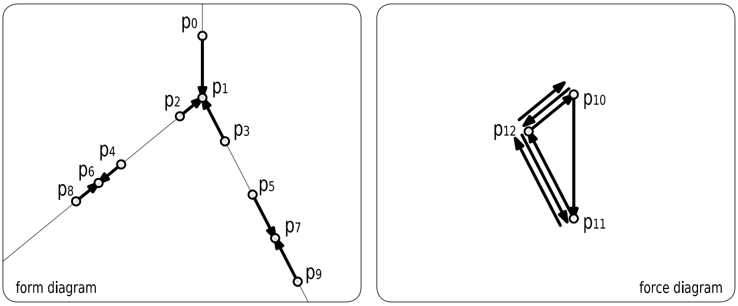


figure 79
equivalent seven forces.



particular rods and some theorems · Some non-fundamental relationships can be defined in order to characterise particular type of rods: zero rods, struts and ties.

$$\mathrm{ZeroRod}[F_0\ F_1\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5] :\longleftrightarrow \mathrm{Rod}[F_0\ F_1\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5] \wedge \mathrm{Coincidence}[p_4 p_5]$$

$$\begin{aligned}\mathrm{Strut}[F_0\ F_1\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5] :\longleftrightarrow\ & \mathrm{Rod}[F_0\ F_1\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5] \\ \wedge\ & \mathrm{PushingForce}[F_0\ p_0\ p_1\ p_4\ p_5] \\ \wedge\ & \mathrm{PushingForce}[F_1\ p_2\ p_3\ p_5\ p_4]\end{aligned}$$

$$\begin{aligned}\mathrm{Tie}[F_0\ F_1\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5] :\longleftrightarrow\ & \mathrm{Rod}[F_0\ F_1\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5] \\ \wedge\ & \mathrm{PullingForce}[F_0\ p_0\ p_1\ p_4\ p_5] \\ \wedge\ & \mathrm{PullingForce}[F_1\ p_2\ p_3\ p_5\ p_4]\end{aligned}$$

The following theorems are deduced from these definitions:

$$\begin{aligned}\mathrm{Rod}[F_0\ F_1\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5] \longleftrightarrow (\ & \mathrm{Strut}[F_0\ F_1\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5] \\ \vee\ & \mathrm{Tie}[F_0\ F_1\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5]\ )\end{aligned}$$

$$\begin{aligned}\mathrm{ZeroRod}[F_0\ F_1\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5] \longleftrightarrow (\ & \mathrm{Strut}[F_0\ F_1\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5] \\ \wedge\ & \mathrm{Tie}[F_0\ F_1\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5]\ )\end{aligned}$$

$$\begin{aligned}&(\mathrm{Strut}[F_0\ F_1\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5] \wedge \neg\mathrm{ZeroRod}[F_0\ F_1\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5]) \\ &\qquad \longleftrightarrow \neg\mathrm{Tie}[F_0\ F_1\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5]\end{aligned}$$

strut-and-tie networks · Each set of forces {F_0, F_1, F_2, ..., F_n} is said to be a "*strut-and-tie network*" if

(a) no point is both a point of application of a force belonging to this set and a point of application of a force not belonging to this set;

(b) no rod — *i.e.* a Rod relationship — is simultaneously formed by a force that belongs to this set and by another force that does not belong to this set.

Since the two forces composing a rod exist as soon as this rod exists, these two forces must also be taken into account in (a). On the basis of this definition, it can be deduced that:

- any strut-and-tie network is a force network
- for any point p_0, if the relationship Rod[F_0 F_1 p_0 p_1 p_2 p_3 p_4 p_5] is verified, then the two relationships Force[F_0 p_0 p_1 p_4 p_5] and Force[F_1 p_2 p_3 p_5 p_4] belong to a same strut-and-tie network

sub-network hulls · The identification of particular force sub-networks inside strut-and-tie networks can easily be performed by listing the set of application points of the forces concerned. Hence the hull that includes all these points of application characterise the force sub-network graphically. The use of sub-network hulls will be of great help in the next sub-section. Some examples are shown in figure 80.

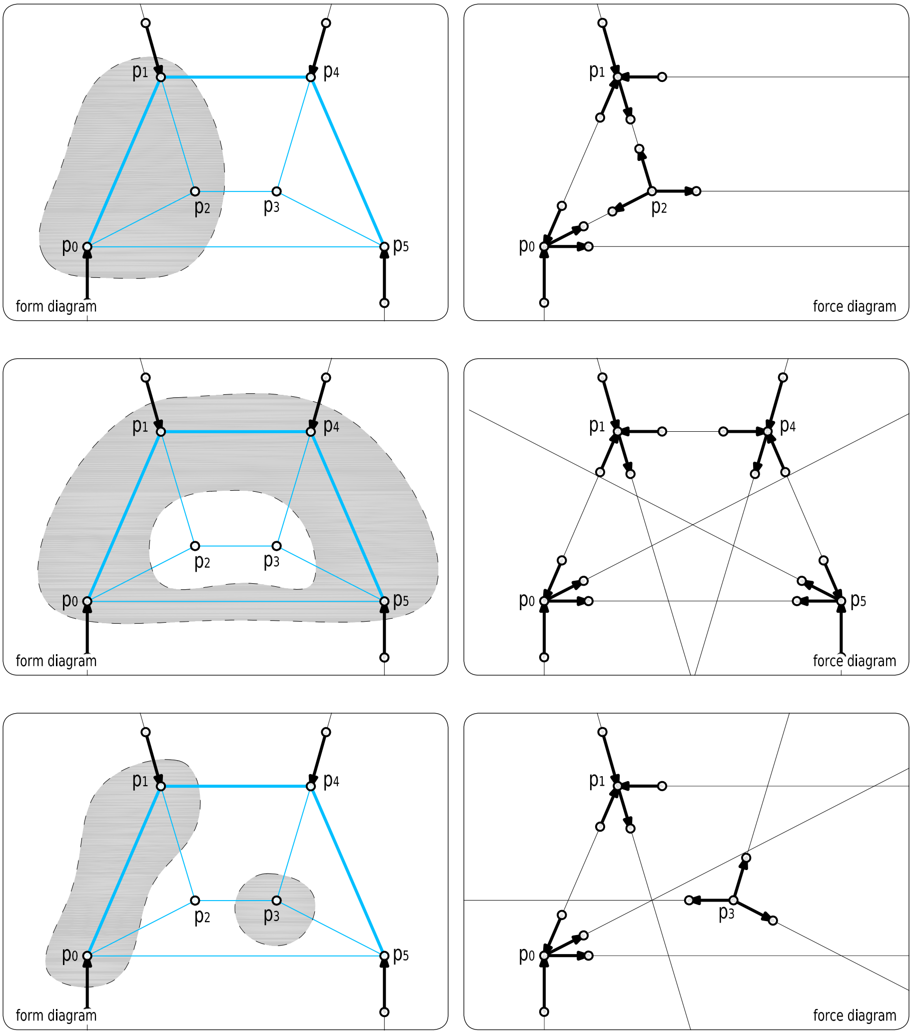


figure 80
(left) sub-network hulls and (right) their corresponding force sub-networks.

definition of other physical objects · Like rods and the Rod relationship, many structural objects can be defined as non-fundamental relationships implementing geometric rules of force networks in static equilibrium. Two examples are shown here: a basic shear-panel and a simple pulley. Once defined, these relationships can be used directly as objects whose internal functioning does not have to be known. In other words, these relationships build a layer of abstraction over the elementary Force, Proximity and Laterality relationships. Moreover, new abstract objects can themselves be combined into more complex objects, adding an additional layer of abstraction to the geometric construction.

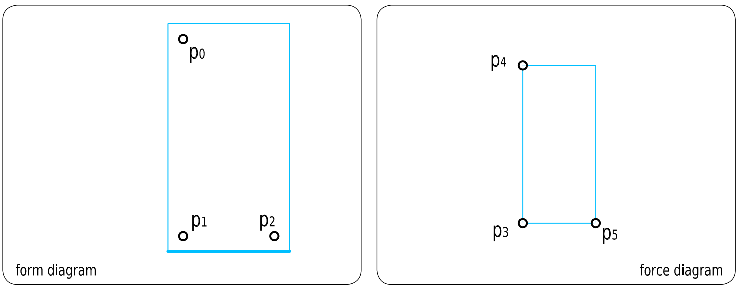


figure 81
a basic shear-panel object ready to be used.

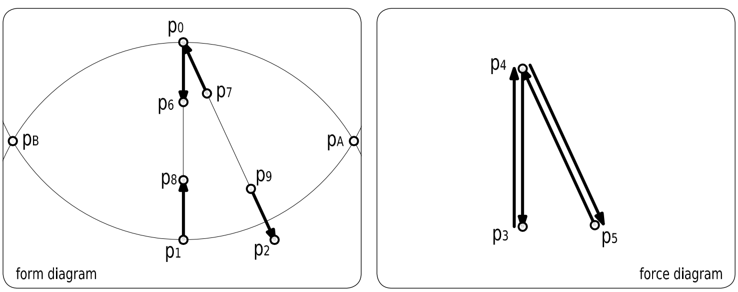


figure 82
the geometric pattern of the BasicShearPanel relationship.

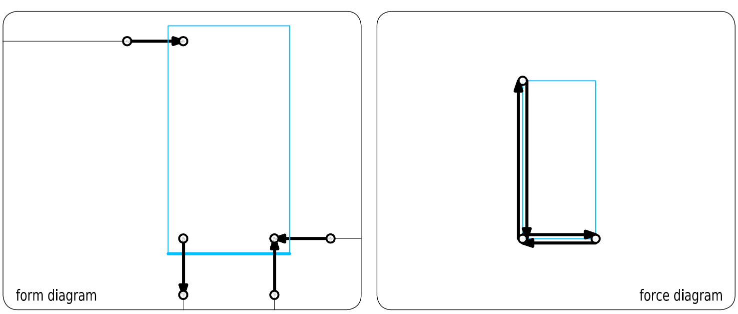


figure 83
application example of a BasicShearPanel object.

The basic shear panel (figure 81) here is equivalent to four forces holding the following geometric properties (figure 82):

$$
\begin{aligned}
&\text{BasicShearPanel}[F_0\ F_1\ F_2\ F_3\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5\ p_6\ p_7\ p_8\ p_9] : \\
&\quad \leftrightarrow \exists p_A\ p_B : \quad \text{Rod}[F_0\ F_1\ p_0\ p_6\ p_1\ p_8\ p_4\ p_3] \wedge \text{Rod}[F_2\ F_3\ p_0\ p_7\ p_2\ p_9\ p_5\ p_4] \\
&\qquad\qquad \wedge \text{Laterality}[p_1\ p_2\ p_3\ p_5] \wedge \text{Laterality}[p_1\ p_2\ p_5\ p_3] \\
&\qquad\qquad \wedge \text{Laterality}[p_1\ p_2\ p_A\ p_B] \wedge \text{Laterality}[p_1\ p_2\ p_B\ p_A] \\
&\qquad\qquad \wedge \text{Proximity}[p_0\ p_A\ p_0\ p_1] \wedge \text{Proximity}[p_0\ p_1\ p_0\ p_A] \\
&\qquad\qquad \wedge \text{Proximity}[p_A\ p_1\ p_0\ p_1] \wedge \text{Proximity}[p_0\ p_1\ p_A\ p_1] \\
&\qquad\qquad \wedge \text{Laterality}\ [p_A\ p_0\ p_0\ p_1] \\
&\qquad\qquad \wedge \text{Proximity}[p_0\ p_B\ p_0\ p_1] \wedge \text{Proximity}[p_0\ p_1\ p_0\ p_B] \\
&\qquad\qquad \wedge \text{Proximity}[p_B\ p_1\ p_0\ p_1] \wedge \text{Proximity}[p_0\ p_1\ p_B\ p_1] \\
&\qquad\qquad \wedge \text{Laterality}\ [p_B\ p_1\ p_1\ p_0]
\end{aligned}
$$

An example of the most direct application of this object is illustrated in figure 83.

In the same way, a pulley can be defined as three forces that hold particular geometric conditions. An object pulley is shown in both diagrams in figure 84 and figure 85 shows the forces equivalent to this geometrical object. An example of an application is shown in figure 86. The definition of the Pulley relationship is as follows:

$$
\begin{aligned}
&\text{Pulley}[F_0\ F_1\ F_2\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5\ p_6\ p_7\ p_8] : \\
&\quad \leftrightarrow \exists p_A\ p_B : \text{Force}[F_0\ p_0\ p_1\ p_6\ p_7] \wedge \text{Force}[F_1\ p_2\ p_3\ p_7\ p_8] \wedge \text{Force}[F_2\ p_4\ p_5\ p_8\ p_6] \\
&\qquad\qquad \wedge \text{Laterality}[p_0\ p_2\ p_2\ p_4] \wedge \text{Laterality}[p_0\ p_2\ p_4\ p_2] \\
&\qquad\qquad \wedge \text{Proximity}[p_0\ p_2\ p_0\ p_4] \wedge \text{Proximity}[p_0\ p_4\ p_0\ p_2] \\
&\qquad\qquad \wedge \text{Laterality}[p_8\ p_6\ p_6\ p_7] \wedge \text{Laterality}[p_8\ p_6\ p_7\ p_6] \\
&\qquad\qquad \wedge \text{Proximity}[p_6\ p_8\ p_7\ p_8] \wedge \text{Proximity}[p_7\ p_8\ p_6\ p_8] \\
&\qquad\qquad \wedge \text{Laterality}[p_6\ p_7\ p_A\ p_B] \wedge \text{Laterality}[p_6\ p_7\ p_B\ p_A] \\
&\qquad\qquad \wedge \text{Proximity}[p_0\ p_A\ p_0\ p_2] \wedge \text{Proximity}[p_0\ p_2\ p_0\ p_A] \\
&\qquad\qquad \wedge \text{Proximity}[p_A\ p_2\ p_0\ p_2] \wedge \text{Proximity}[p_0\ p_2\ p_A\ p_2] \\
&\qquad\qquad \wedge \text{Laterality}\ [p_A\ p_0\ p_0\ p_2] \\
&\qquad\qquad \wedge \text{Proximity}[p_0\ p_B\ p_0\ p_2] \wedge \text{Proximity}[p_0\ p_2\ p_0\ p_B] \\
&\qquad\qquad \wedge \text{Proximity}[p_B\ p_2\ p_0\ p_0] \wedge \text{Proximity}[p_0\ p_2\ p_B\ p_2] \\
&\qquad\qquad \wedge \text{Laterality}\ [p_B\ p_2\ p_2\ p_0]
\end{aligned}
$$

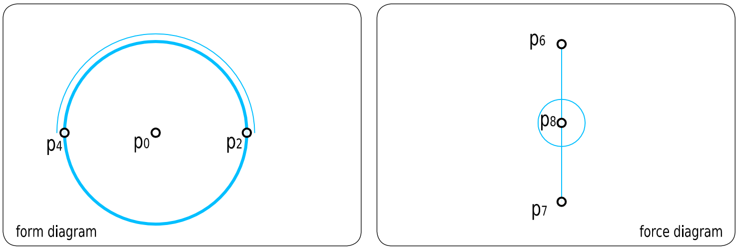


figure 84
a basic pulley object ready to be used.

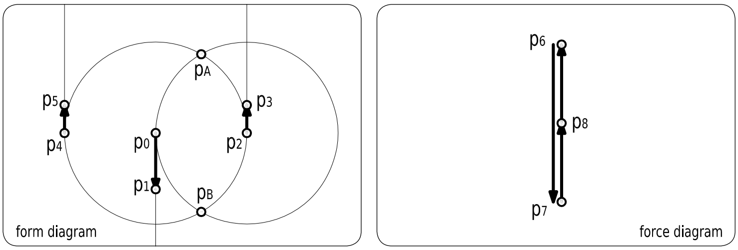


figure 85
the geometric pattern of the Pulley relationship.

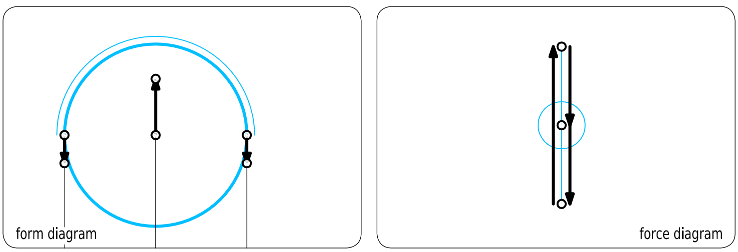


figure 86
application example of a Pulley object.

12 static equilibrium

This sub-section augments the axioms developed in the previous sub-sections in order to ensure static equilibrium. First it identifies the minimum condition required to guarantee the static equilibrium of strut-and-tie networks. It then defines four new axioms that reflect this condition geometrically. Lastly, the use of these axioms is illustrated using a brief example.

classical definition of static equilibrium · The concept of static equilibrium is usually defined in literature as follows: *"a system of coplanar forces is in equilibrium if, and only if, (a) its resultant is zero, and (b) the algebraic sum of the moments of all its forces is zero about any point in its plane."* (Ziwet/Field·1912, page 165). In an abbreviated fashion, it means that a body is in static equilibrium if the following two conditions are satisfied:

(a) *translational equilibrium:* ∑(Forces applied on the body) = 0
(b) *rotational equilibrium:* ∑(Moments applied on the body) = 0

simplified definition of static equilibrium for strut-and-tie networks · Since bodies are exclusively strut-and-tie networks, the definition can be simplified as follows:

> *"A strut-and-tie network is in static equilibrium if the sum of all the forces applied on each point of this network is zero."*

Using the fact that a sum of forces is zero if its drawing in the force diagram is a closed polygon, using the definition of "*minimum force network*" presented in the paragraph entitled "force networks" (page 85), and using the fact that rods can be entirely defined with just forces and geometric rules, this definition can be explained further:

> *"A strut-and-tie network is in static equilibrium if each minimum force network it contains produces a closed polygon in the force diagram."*

It is possible to provide a quick proof of these simplifications by recalling the following corollary of the classical definition of static equilibrium: "*If a body is in static equilibrium, every part of this body is also in equilibrium*".

Since (1) the only way to modify the equilibrium of a body is to change the forces applied on it and since (2) a force here is always applied on a point, then, to guarantee the equilibrium of a strut-and-tie network is tantamount to ensuring the equilibrium of each point of this network. Considering each point individually, its rotational equilibrium is systematically satisfied because the lever arm of applied forces about the point itself is zero. In other words, moments do not have to be checked. The only condition to satisfy is to ensure that the vectorial sum of forces applied on each point of the strut-and-tie network, *i.e.* each minimum force network, is zero.

To secure this condition for any kind of minimum force network, *i.e.* for any quantity, magnitude and orientation of forces, the choice has been made to introduce four complementary axioms: Ax.23 describes the rule guaranteeing the static equilibrium of a simple minimum force network and Ax.24, Ax.25 and Ax.26 propose a method to transform this simple network into any kind of minimum force network in static equilibrium iteratively. Forces applied on various points in the space diagram can then be linked by rods using the Rod relationship. The resulting strut-and-tie network is guaranteed to be in static equilibrium.

existence of a force network in equilibrium · The first axiom (Ax.23) describes a simple case of equilibrium that is always proved to be true. This case is chosen to be the zero force (figure 87):

Ax.23 — *existence of a zero force:*

$$(\text{FormDiagramMembership}[p_0] \wedge \text{ForceDiagramMembership}[p_1]) \longrightarrow \exists\, F_0: \text{Force}[F_0\ p_0\ p_0\ p_1\ p_1]$$

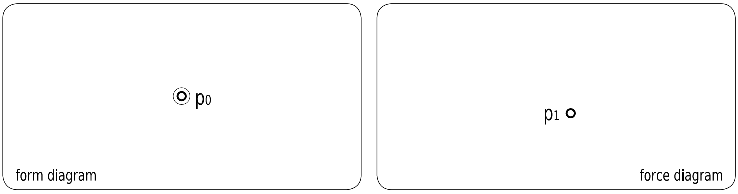


figure 87
axiom Ax.23, existence of a zero force.

Other basic cases of static equilibrium that can be used instead of a zero force are, for instance, two forces in static equilibrium around a point (figure 88) or three forces in static equilibrium around a point (figure 89). The following variants are therefore always true:

Ax.23 (variant a) — *existence of two forces in static equilibrium:*

$$\begin{aligned} (\quad & \text{FormDiagramMembership}[p_0] \\ \wedge\ & \text{ForceDiagramMembership}[p_2] \\ \wedge\ & \text{ForceDiagramMembership}[p_3]) \\ & \longrightarrow \exists\, F_0F_1p_1p_4\text{: Force}[F_0\ p_0\ p_1\ p_2\ p_3] \wedge \text{Force}[F_1\ p_0\ p_4\ p_3\ p_2] \end{aligned}$$

Ax.23 (variant b) — *existence of three forces in static equilibrium:*

$$\begin{aligned} (\quad & \text{FormDiagramMembership}[p_0] \\ \wedge\ & \text{ForceDiagramMembership}[p_2] \\ \wedge\ & \text{ForceDiagramMembership}[p_3] \\ \wedge\ & \text{ForceDiagramMembership}[p_5]\ ) \\ & \longrightarrow \exists\, F_0F_1F_2p_1p_4p_6 : \\ & \quad \text{Force}[F_0\ p_0\ p_1\ p_2\ p_3] \wedge \text{Force}[F_1\ p_0\ p_4\ p_3\ p_5] \wedge \text{Force}[F_2\ p_0\ p_6\ p_5\ p_2] \end{aligned}$$

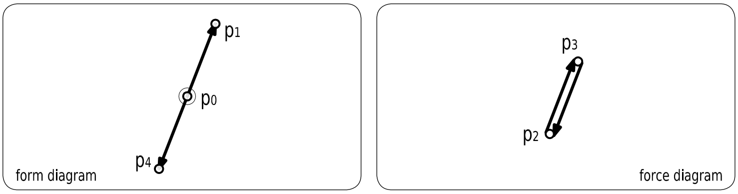


figure 88
two forces in static equilibrium around a point — axiom Ax.23 variant a.

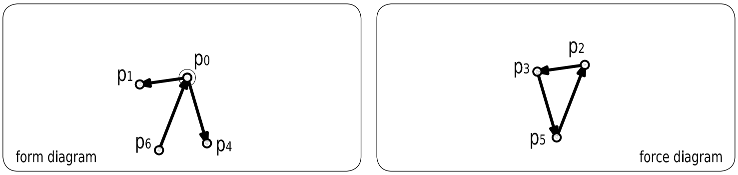


figure 89
three forces in static equilibrium around a point— axiom Ax.23 variant b.

parallelogram of forces · The second axiom (Ax.24) describes how to resolve a force (figure 90) into two components (figure 91) with the help of the parallelogram rule (Benvenuto·1985):

Ax.24 — *parallelogram of forces:*

$$\text{Force}[F_0\ p_0\ p_1\ p_2\ p_3] \longleftrightarrow \text{Force}[F_1\ p_0\ p_4\ p_2\ p_5] \wedge \text{Force}[F_2\ p_0\ p_6\ p_5\ p_3]$$

This axiom is the only device that allows a force network to be transformed into a new one. The force F_0 is the resultant and the forces F_1 and F_2 are the two force components.

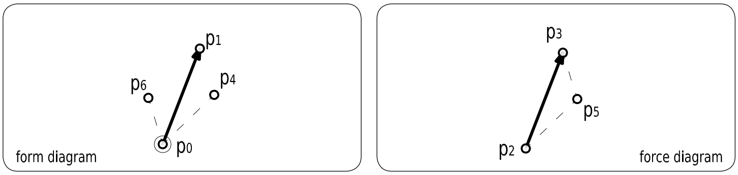


figure 90
a force before being resolved into two components.

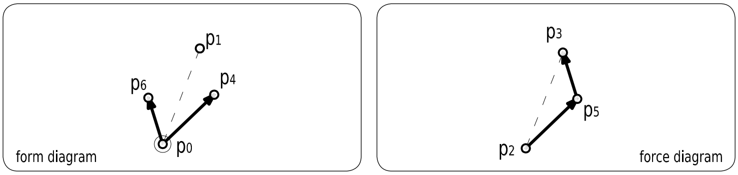


figure 91
two components of the force in figure 90.

Ax.24 could have been defined more generally. However, a more general definition would not be compatible with the axiom Ax.27 (page 109) developed in the next sub-section.

The transformation rule defined by Ax.24 has to be channelled by two new axioms in order to be used properly. Firstly, a force cannot be regarded as two different resultants, or in other words, a force cannot be the resultant of two different pairs of forces:

Ax.25 — *univocal definition of resultants:*

$$\begin{aligned}(\quad & (\mathrm{Force}[F_0\ p_0\ p_1\ p_2\ p_3] \longleftrightarrow \mathrm{Force}[F_1\ p_0\ p_4\ p_2\ p_5] \wedge \mathrm{Force}[F_2\ p_0\ p_6\ p_5\ p_3]) \\ \wedge\ & (\mathrm{Force}[F_0\ p_0\ p_1\ p_2\ p_3] \longleftrightarrow \mathrm{Force}[F_3\ p_0\ p_7\ p_2\ p_8] \wedge \mathrm{Force}[F_4\ p_0\ p_9\ p_8\ p_3])\ ) \\ & \qquad \longrightarrow (\mathrm{Equipollence}[F_1\ F_3] \wedge \mathrm{Equipollence}[F_2\ F_4])\end{aligned}$$

Secondly, a component cannot be used two form two different resultants, or in other words, two forces cannot be resolved into a same force:

Ax.26 — *univocal definition of force components:*

$$\begin{aligned}(\quad & (\mathrm{Force}[F_0\ p_0\ p_1\ p_2\ p_3] \longleftrightarrow \mathrm{Force}[F_1\ p_0\ p_4\ p_2\ p_5] \wedge \mathrm{Force}[F_2\ p_0\ p_6\ p_5\ p_3]) \\ \wedge\ & (\mathrm{Force}[F_3\ p_0\ p_7\ p_2\ p_8] \longleftrightarrow \mathrm{Force}[F_1\ p_0\ p_4\ p_2\ p_5] \wedge \mathrm{Force}[F_4\ p_0\ p_9\ p_5\ p_8])\ ) \\ & \qquad \longrightarrow \mathrm{Equipollence}[F_0\ F_3]\end{aligned}$$

Equivalence of Ax.23 and its variants · The equivalence of axiom Ax.23 with its two variants can now be proved by applying Ax.24 once (figure 92) or twice (figure 93) on Ax.23 (figure 87):

$$\begin{aligned}& \mathrm{Force}[F_0\ p_0\ p_1\ p_2\ p_2] \\ \longleftrightarrow\ & \mathrm{Force}[F_1\ p_0\ p_3\ p_2\ p_4] \wedge \mathrm{Force}[F_2\ p_0\ p_5\ p_4\ p_2] \\ \longleftrightarrow\ & \mathrm{Force}[F_1\ p_0\ p_3\ p_2\ p_4] \wedge \mathrm{Force}[F_3\ p_0\ p_6\ p_4\ p_7] \wedge \mathrm{Force}[F_4\ p_0\ p_8\ p_7\ p_2]\end{aligned}$$

figure 92
first application of Ax.24 upon Ax.23.

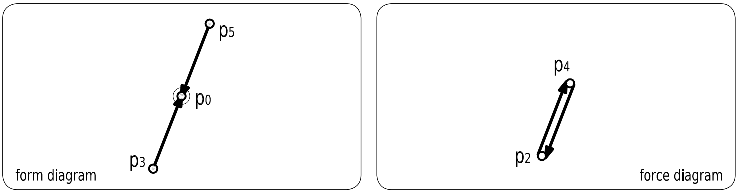
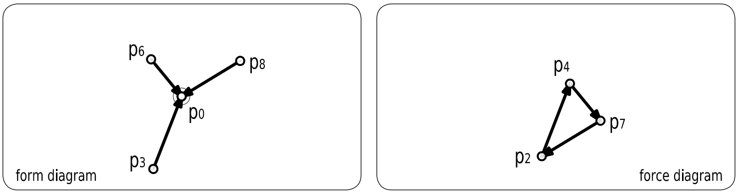


figure 93
second application of Ax.24 upon Ax.23.



To apply axiom Ax.24 again would increase the number of forces acting on p_0 without ever affecting the closed nature of the force polygon.

proof of static equilibrium · If all the n forces in a given network are unique — *i.e.* if for all i and j that belong to [0,n[such that i≠j, $\neg\text{Equipollence}[F_i\ F_j]$ is verified — unless explicitly stated by a $\text{Equipollence}[F_A\ F_B]$ relationship, then any logical sentence satisfying the previous axioms is expected to describe a force network in static equilibrium.

As an illustration, the following lines initially describe a strut-and-tie network and then proves it to be in static equilibrium by using Ax.23 and Ax.24 (figure 94):

describing the strut-and-tie network:

$\text{Force}[F_0\ p_0\ p_1\ p_6\ p_7] \wedge \text{Force}[F_1\ p_2\ p_3\ p_7\ p_8] \wedge \text{Force}[F_2\ p_4\ p_5\ p_8\ p_6]$
$\wedge\, \text{Rod}[F_3\ F_4\ p_0\ p_{10}\ p_2\ p_{11}\ p_7\ p_9]$
$\wedge\, \text{Rod}[F_5\ F_6\ p_2\ p_{12}\ p_4\ p_{13}\ p_8\ p_9]$
$\wedge\, \text{Rod}[F_7\ F_8\ p_4\ p_{14}\ p_0\ p_{15}\ p_6\ p_9]$

using the definition of the Rod *relationship:*

$\longrightarrow \text{Force}[F_0\ p_0\ p_1\ p_6\ p_7] \wedge \text{Force}[F_1\ p_2\ p_3\ p_7\ p_8] \wedge \text{Force}[F_2\ p_4\ p_5\ p_8\ p_6]$
$\wedge\, \text{Force}[F_3\ p_0\ p_{10}\ p_7\ p_9] \wedge \text{Force}[F_4\ p_2\ p_{11}\ p_9\ p_7]$
$\wedge\, \text{Force}[F_5\ p_2\ p_{12}\ p_8\ p_9] \wedge \text{Force}[F_6\ p_4\ p_{13}\ p_9\ p_8]$
$\wedge\, \text{Force}[F_7\ p_4\ p_{14}\ p_6\ p_9] \wedge \text{Force}[F_8\ p_0\ p_{15}\ p_9\ p_6]$

grouping forces by points of applications:

$\longrightarrow \text{Force}[F_0\ p_0\ p_1\ p_6\ p_7] \wedge \text{Force}[F_3\ p_0\ p_{10}\ p_7\ p_9] \wedge \text{Force}[F_8\ p_0\ p_{15}\ p_9\ p_6]$
$\wedge\, \text{Force}[F_1\ p_2\ p_3\ p_7\ p_8] \wedge \text{Force}[F_4\ p_2\ p_{11}\ p_9\ p_7] \wedge \text{Force}[F_5\ p_2\ p_{12}\ p_8\ p_9]$
$\wedge\, \text{Force}[F_2\ p_4\ p_5\ p_8\ p_6] \wedge \text{Force}[F_6\ p_4\ p_{13}\ p_9\ p_8] \wedge \text{Force}[F_7\ p_4\ p_{14}\ p_6\ p_9]$

applying Ax.24*:*

$\longrightarrow \text{Force}[F_A\ p_0\ p_A\ p_6\ p_9] \ \wedge \text{Force}[F_8\ p_0\ p_{15}\ p_9\ p_6]$
$\wedge \text{Force}[F_B\ p_2\ p_B\ p_9\ p_8] \ \wedge \text{Force}[F_5\ p_2\ p_{12}\ p_8\ p_9]$
$\wedge \text{Force}[F_C\ p_4\ p_C\ p_9\ p_6] \ \wedge \text{Force}[F_7\ p_4\ p_{14}\ p_6\ p_9]$

applying Ax.24 *again:*

$\longrightarrow \text{Force}[F_D\ p_0\ p_D\ p_6\ p_6] \wedge \text{Force}[F_E\ p_2\ p_E\ p_8\ p_8] \wedge \text{Force}[F_F\ p_4\ p_F\ p_9\ p_9]$

applying Ax.23*:*

$\longrightarrow$true

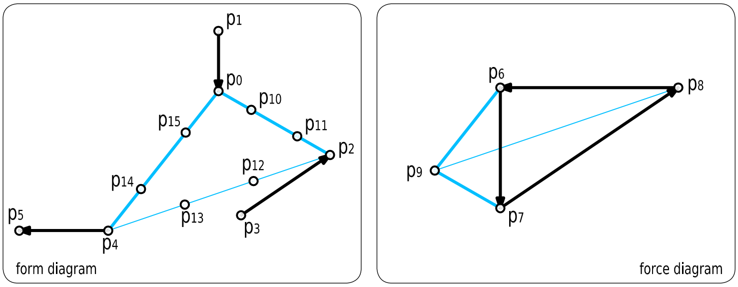


figure 94
a strut-and-tie network in static equilibrium.

The following lines use the same proof technique to identify one of the sufficient conditions for the network of figure 95 to be in static equilibrium:

describing the force network:

$\text{Force}[F_0\ p_0\ p_1\ p_2\ p_3] \wedge \text{Force}[F_1\ p_0\ p_4\ p_3\ p_5]$
$\wedge \text{Force}[F_2\ p_0\ p_6\ p_3\ p_2] \wedge \text{Force}[F_3\ p_0\ p_7\ p_5\ p_2]$

applying Ax.24 *such that* $F_1+F_3=F_4$*:*

$\text{Force}[F_0\ p_0\ p_1\ p_2\ p_3] \wedge \text{Force}[F_2\ p_0\ p_6\ p_3\ p_2] \wedge \text{Force}[F_4\ p_0\ p_6\ p_3\ p_2]$

It is worth noting that F_2 and F_4 cannot be amalgamated because no relationship explicitly states it, in other words: $F_2 \neq F_4$ or $\neg\text{Equipollence}[F_2\ F_4]$. Also, F_0 cannot be duplicated — *i.e.* it cannot be added to F_2 and added again to F_4 — because (1) Ax.26 would imply that $F_0+F_2=F_A$ and that $F_0+F_4=F_A$ and (2) it would force Ax.25 to be false since $F_2 \neq F_4$. Consequently, one way to continue the demonstration is to apply Ax.24 again. For instance:

applying Ax.24 *such that* $F_0+F_2=F_5$*:*

$\text{Force}[F_5\ p_0\ p_5\ p_2\ p_2] \wedge \text{Force}[F_4\ p_0\ p_6\ p_3\ p_2]$

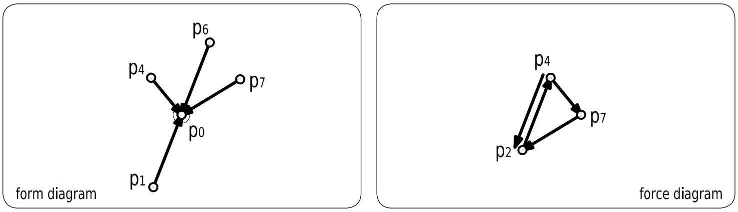
applying Ax.19 *(page 83):*

$\text{Force}[F_5\ p_0\ p_0\ p_2\ p_2] \wedge \text{Force}[F_4\ p_0\ p_6\ p_3\ p_2]$

applying Ax.23 *(page 100)::*

$\text{Force}[F_4\ p_0\ p_6\ p_3\ p_2]$

figure 95
a force network that is not in static equilibrium.



This last relationship is only satisfied if Coincidence[p_2 p_3] is true, meaning that axioms Ax.19 and Ax.23 can then be used again and prove that the original sentence is true. In other words, points p_2 and p_3 must be coincident if the force network has to be in static equilibrium.

As a final note about static equilibrium, it is acknowledged that Ax.23 to Ax.26 are independent of whether forces exert a pull or a push on the application point.

13 uniform reading cycle

This sub-section introduces the last axiom, firstly providing a definition of it and then presenting its direct implications. The need for this axiom is subsequently explained. This explanation leads to the identification of an equivalent axiom. Lastly, further consequences of this axiom are developed.

clockwise reading cycle of minimum force networks · This new axiom takes three successive forces in the force diagram and describes the geometric condition that p_2, *i.e.* the point defining the type of application of the second force, must satisfy in the form diagram so that the three forces are read clockwise in the form diagram, in the same order as described in the force diagram.

First, the axiom defines three forces that are applied on the same point p_0 in the form diagram and that are consecutive in the force diagram, without thereby forming a closed polygon, *i.e.* other forces might be applied on p_0 as well:

$$\mathrm{Force}[F_0\ p_0\ p_1\ p_4\ p_5] \wedge \mathrm{Force}[F_1\ p_0\ p_2\ p_5\ p_6] \wedge \mathrm{Force}[F_2\ p_0\ p_3\ p_6\ p_7]$$

The axiom then says that p_2 must be within the graphical region bordered by the previous and the next forces (figure 96 and figure 97). Given that the angle between these two forces is acute or obtuse, the corresponding region is either the intersection or the union of two half-planes. These two cases are taken into account in a generic manner by defining points p_8 and p_9 such that they are coincident to p_1 and p_3 when the angle is acute (figure 97) and such that they divide the angle into three sections when it is obtuse (figure 96). Point p_8 is consequently defined by the following sentence:

$$\begin{aligned}&\mathrm{Laterality}[p_8\ p_0\ p_0\ p_1] \wedge \mathrm{Laterality}[p_8\ p_0\ p_1\ p_0]\\ &\quad \wedge \mathrm{Proximity}[p_8\ p_0\ p_0\ p_1] \wedge \mathrm{Proximity}[p_0\ p_1\ p_8\ p_0]\\ &\quad \wedge(\neg\mathrm{Proximity}[p_3\ p_0\ p_0\ p_0] \vee \neg\mathrm{Proximity}[p_8\ p_1\ p_1\ p_1] \vee \mathrm{Proximity}[p_8\ p_0\ p_0\ p_0])\\ &\quad \wedge(\mathrm{Proximity}[p_3\ p_0\ p_0\ p_0] \vee \neg\mathrm{Laterality}[p_8\ p_0\ p_3\ p_0] \vee \mathrm{Proximity}[p_8\ p_0\ p_0\ p_0])\end{aligned}$$

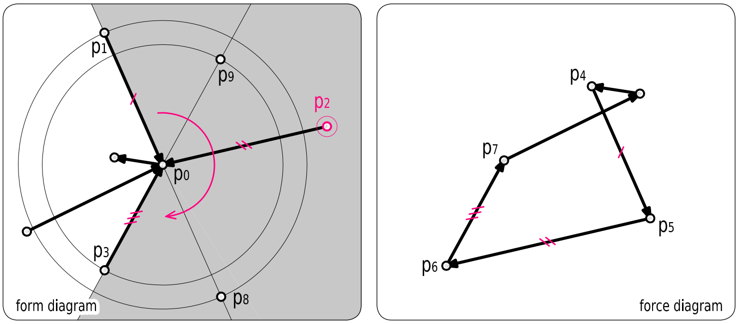


figure 96
Ax.27: the grey area is the obtuse section in which point p_2 must stay in order to ensure a clockwise reading of the forces in the form diagram.

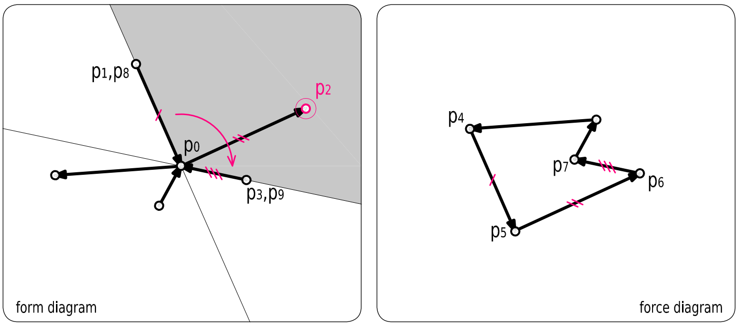


figure 97
Ax.27: the grey area is the acute section in which point p_2 must stay in order to ensure a clockwise reading of the forces in the form diagram.

and point p_9 by:

Laterality[p_9 p_0 p_0 p_3] ∧ Laterality[p_9 p_0 p_3 p_0]
∧ Proximity[p_9 p_0 p_0 p_3] ∧ Proximity[p_0 p_3 p_9 p_0]
∧(¬Proximity[p_1 p_0 p_0 p_0] ∨ ¬Proximity[p_9 p_3 p_3 p_3] ∨ Proximity[p_9 p_0 p_0 p_0])
∧(Proximity[p_1 p_0 p_0 p_0] ∨ ¬Laterality[p_9 p_0 p_0 p_1] ∨ Proximity[p_9 p_0 p_0 p_0])

The need for the last two lines of both definitions will be explained in the paragraph entitled “particular cases” (page 110).

Given p_8 and p_9, p_2 must hold the following disjunction of conjunctions in order for the three forces to be read clockwise:

(Laterality[p_2 p_0 p_1 p_0] ∧ Laterality[p_2 p_0 p_0 p_9])
∨ (Laterality[p_2 p_0 p_1 p_0] ∧ Laterality[p_2 p_0 p_0 p_3])
∨ (Laterality[p_2 p_0 p_8 p_0] ∧ Laterality[p_2 p_0 p_0 p_3])

As a result, the entire axiom is written as follows:

Ax.27 — *local reading cycle of forces:*

$$(\mathrm{Force}[F_0\ p_0\ p_1\ p_4\ p_5] \land \mathrm{Force}[F_1\ p_0\ p_2\ p_5\ p_6] \land \mathrm{Force}[F_2\ p_0\ p_3\ p_6\ p_7])$$

$$\longrightarrow \exists p_A p_B:$$

$$(\quad (\mathrm{Laterality}[p_2\ p_0\ p_1\ p_0] \land \mathrm{Laterality}[p_2\ p_0\ p_0\ p_B])$$
$$\lor (\mathrm{Laterality}[p_2\ p_0\ p_1\ p_0] \land \mathrm{Laterality}[p_2\ p_0\ p_0\ p_3])$$
$$\lor (\mathrm{Laterality}[p_2\ p_0\ p_A\ p_0] \land \mathrm{Laterality}[p_2\ p_0\ p_0\ p_3]) \ )$$

$$\land \mathrm{Laterality}[p_A\ p_0\ p_0\ p_1] \land \mathrm{Laterality}[p_8\ p_0\ p_1\ p_0]$$
$$\land \mathrm{Proximity}[p_A\ p_0\ p_0\ p_1] \land \mathrm{Proximity}[p_0\ p_1\ p_A\ p_0]$$
$$\land(\neg\mathrm{Proximity}[p_3\ p_0\ p_0\ p_0] \lor \neg\mathrm{Proximity}[p_A\ p_1\ p_1\ p_1] \lor \mathrm{Proximity}[p_A\ p_0\ p_0\ p_0])$$
$$\land(\mathrm{Proximity}[p_3\ p_0\ p_0\ p_0] \lor \neg\mathrm{Laterality}[p_A\ p_0\ p_3\ p_0] \lor \mathrm{Proximity}[p_A\ p_0\ p_0\ p_0])$$

$$\land \mathrm{Laterality}[p_B\ p_0\ p_0\ p_3] \land \mathrm{Laterality}[p_B\ p_0\ p_3\ p_0]$$
$$\land \mathrm{Proximity}[p_B\ p_0\ p_0\ p_3] \land \mathrm{Proximity}[p_0\ p_3\ p_B\ p_0]$$
$$\land(\neg\mathrm{Proximity}[p_1\ p_0\ p_0\ p_0] \lor \neg\mathrm{Proximity}[p_B\ p_3\ p_3\ p_3] \lor \mathrm{Proximity}[p_B\ p_0\ p_0\ p_0])$$
$$\land(\mathrm{Proximity}[p_1\ p_0\ p_0\ p_0] \lor \neg\mathrm{Laterality}[p_B\ p_0\ p_0\ p_1] \lor \mathrm{Proximity}[p_B\ p_0\ p_0\ p_0])$$

This axiom does not explicitly prevent a fourth force in the form diagram from being between the first and the third forces as well. However, this case is impossible since the axiom rules every triplet of consecutive forces in the form diagram. For example, figure 98 shows five forces applied on a point p_0.

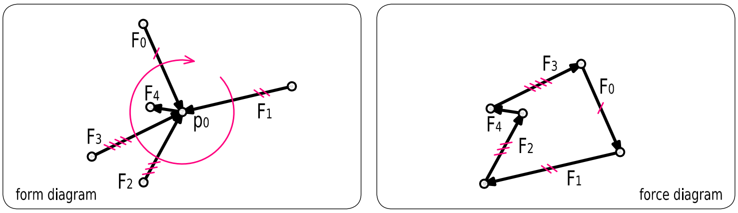


figure 98
five consecutive forces read clockwise.

According to the force polygon in the force diagram, the reading cycle in the form diagram is $\{F_0\ F_1\ F_2\ F_3\ F_4\}$.

The axiom therefore holds five times:

(a) F_1 must be clockwise after F_0 and before F_2
(b) F_2 must be clockwise after F_1 and before F_3
(c) F_3 must be clockwise after F_2 and before F_4
(d) F_4 must be clockwise after F_3 and before F_0
(e) F_0 must be clockwise after F_4 and before F_1

Consequently, force F_3, for instance, will never be between F_0 and F_1 or between F_1 and F_2 since condition (b) means F_3 has to be clockwise after F_2 and condition (c) means F_3 has to be clockwise before F_4. This comment is applicable to all the forces applied on the same point, regardless of their quantity.

particular cases · This paragraph examines five particular behaviours of axiom Ax.27: (1) when it does not constrain the type of application of the force, (2) when the previous force is zero, (3) when the previous and the next force are zero, (4) when all the forces applied are zero forces, and (5) when there are only two forces applied.

(1) In some cases, the second of three consecutive forces remains after the first and before the third regardless of whether it exerts a pull (figure 99) or a push (figure 100). As a consequence, the point defining the type of application of the second force, *i.e.* p_2, has two possible positions. The definition of Ax.27 intrinsically takes this possibility into account.

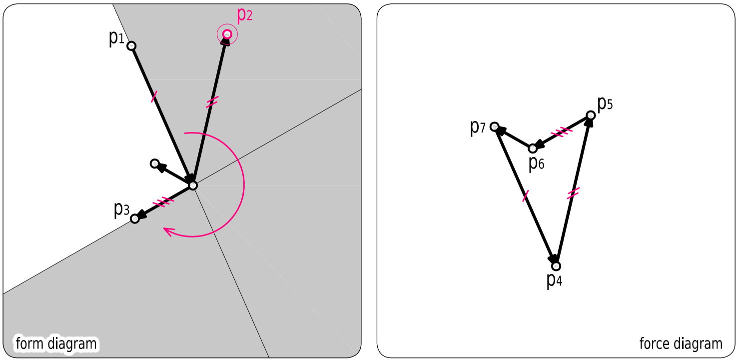


figure 99
in grey, the region in which p_2 must stay in order to ensure a clockwise reading cycle; the force defined by p_2 exerts a pull.

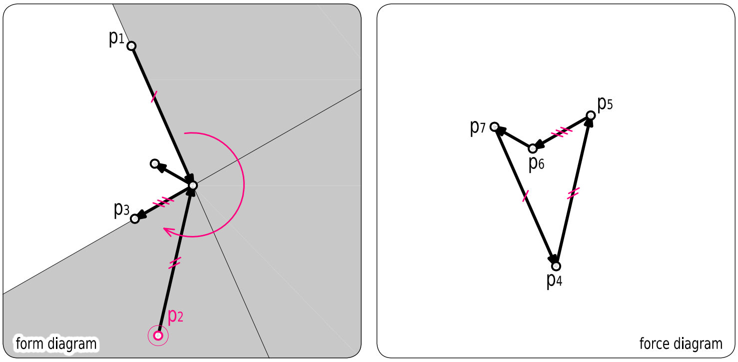


figure 100
in grey, the region in which p_2 must stay in order to ensure a clockwise reading cycle; the force defined by p_2 exerts a push.

(2) Axiom Ax.27 remains sufficient to guarantee a clockwise reading cycle when the preceding force (or the following one) is zero. The following illustration is based on the cycle $\{F_0\ F_1\ F_2\ F_3\ F_4\}$ consisting of the following forces:

$\text{Force}[F_0\ p_0\ p_1\ p_6\ p_7]$
$\text{Force}[F_1\ p_0\ p_2\ p_7\ p_8]$
$\text{Force}[F_2\ p_0\ p_3\ p_8\ p_9]$
$\text{Force}[F_3\ p_0\ p_4\ p_9\ p_{10}]$
$\text{Force}[F_4\ p_0\ p_5\ p_{10}\ p_6]$

When F_0 is a zero force, points p_6 and p_7 are coincident and, according to Ax.19 (page 83), p_0 and p_1 are also coincident. It follows that the application of Ax.27 for the series $\{F_0\ F_1\ F_2\}$ does not condition p_2 at all and is consequently superfluous (figure 101):

Ax.27

$$\begin{aligned}
\longrightarrow (\ & \text{Force}[F_0\ p_0\ p_0\ p_6\ p_6] \wedge \text{Force}[F_1\ p_0\ p_2\ p_6\ p_8] \wedge \text{Force}[F_2\ p_0\ p_3\ p_8\ p_9] \\
& \longrightarrow \exists p_{11} p_{12}: (\ (\text{Laterality}[p_2\ p_0\ p_0\ p_0] \wedge \text{Laterality}[p_2\ p_0\ p_0\ p_{12}]) \\
& \qquad \vee (\text{Laterality}[p_2\ p_0\ p_0\ p_0] \wedge \text{Laterality}[p_2\ p_0\ p_0\ p_3]) \\
& \qquad \vee (\text{Laterality}[p_2\ p_0\ p_{11}\ p_0] \wedge \text{Laterality}[p_2\ p_0\ p_0\ p_3])\ ) \\
& \wedge \text{Laterality}[p_{11}\ p_0\ p_0\ p_0] \wedge \text{Laterality}[p_8\ p_0\ p_0\ p_0] \\
& \wedge \text{Proximity}[p_{11}\ p_0\ p_0\ p_0] \wedge \text{Proximity}[p_0\ p_0\ p_{11}\ p_0] \\
& \wedge(\neg\text{Proximity}[p_3\ p_0\ p_0\ p_0] \vee \neg\text{Proximity}[p_{11}\ p_0\ p_0\ p_0] \vee \text{Proximity}[p_{11}\ p_0\ p_0\ p_0]) \\
& \wedge(\text{Proximity}[p_3\ p_0\ p_0\ p_0] \vee \neg\text{Laterality}[p_{11}\ p_0\ p_3\ p_0] \vee \text{Proximity}[p_{11}\ p_0\ p_0\ p_0]) \\
& \wedge \text{Laterality}[p_{12}\ p_0\ p_0\ p_3] \wedge \text{Laterality}[p_{12}\ p_0\ p_3\ p_0] \\
& \wedge \text{Proximity}[p_{12}\ p_0\ p_0\ p_3] \wedge \text{Proximity}[p_0\ p_3\ p_{12}\ p_0] \\
& \wedge(\neg\text{Proximity}[p_0\ p_0\ p_0\ p_0] \vee \neg\text{Proximity}[p_{12}\ p_3\ p_3\ p_3] \vee \text{Proximity}[p_{12}\ p_0\ p_0\ p_0]) \\
& \wedge(\text{Proximity}[p_0\ p_0\ p_0\ p_0] \vee \neg\text{Laterality}[p_{12}\ p_0\ p_0\ p_0] \vee \text{Proximity}[p_{12}\ p_0\ p_0\ p_0])
\end{aligned}$$

$$\begin{aligned}
\longleftrightarrow (\ & \text{Force}[F_1\ p_0\ p_2\ p_6\ p_8] \wedge \text{Force}[F_2\ p_0\ p_3\ p_8\ p_9] \\
& \longrightarrow \exists p_{11}\ p_{12}: (\ \text{Laterality}[p_2\ p_0\ p_0\ p_{12}] \vee \text{Laterality}[p_2\ p_0\ p_0\ p_3]\ ) \\
& \qquad \wedge \text{Proximity}[p_{11}\ p_0\ p_0\ p_0] \\
& \qquad \wedge \text{Laterality}[p_{12}\ p_0\ p_0\ p_3] \wedge \text{Laterality}[p_{12}\ p_0\ p_3\ p_0] \\
& \qquad \wedge \text{Proximity}[p_{12}\ p_0\ p_0\ p_3] \wedge \text{Proximity}[p_0\ p_3\ p_{12}\ p_0] \\
& \qquad \wedge \neg\text{Proximity}[p_{12}\ p_3\ p_3\ p_3]\ )
\end{aligned}$$

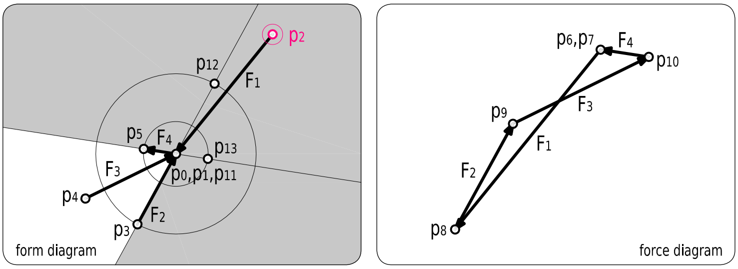


figure 101 behaviour of Ax.27 when the first of the three consecutive forces is a zero force.

$\longleftrightarrow$ (Force[F_1 p_0 p_2 p_6 p_8] $\wedge$ Force[F_2 p_0 p_3 p_8 p_9]
$\longrightarrow \exists p_{11}\, p_{12}$: Proximity[$p_{11}$ p_0 p_0 p_0]
$\wedge$ Laterality[p_{12} p_0 p_0 p_3] $\wedge$ Laterality[p_{12} p_0 p_3 p_0]
$\wedge$ Proximity[p_{12} p_0 p_0 p_3] $\wedge$ Proximity[p_0 p_3 p_{12} p_0]
$\wedge$ $\neg$Proximity[p_{12} p_3 p_3 p_3])

$\longleftrightarrow$ (Force[F_1 p_0 p_2 p_6 p_8] $\wedge$ Force[F_2 p_0 p_3 p_8 p_9] $\longrightarrow$ true)

$\longleftrightarrow$ true

However, since p_6 and p_7 are coincident, axiom Ax.27 also holds for the series {F_4 F_1 F_2} which in turn constrains the position of p_2 (figure 101):

Force[F_4 p_0 p_5 p_{10} p_6] $\wedge$ Force[F_1 p_0 p_2 p_6 p_8] $\wedge$ Force[F_2 p_0 p_3 p_8 p_9]
$\longrightarrow \exists p_{11} p_{12}$: ((Laterality[$p_2$ p_0 p_5 p_0] $\wedge$ Laterality[p_2 p_0 p_0 p_{12}])
$\vee$ (Laterality[p_2 p_0 p_5 p_0] $\wedge$ Laterality[p_2 p_0 p_0 p_3])
$\vee$ (Laterality[p_2 p_0 p_{13} p_0] $\wedge$ Laterality[p_2 p_0 p_0 p_3]))
$\wedge$ ...

(3) Axiom Ax.27 still remains sufficient to guarantee a clockwise reading cycle when the preceding and following forces are zero. If F_0 and F_2 are two zero forces, then points p_6 and p_7 are coincident and points p_8 and p_9 are coincident. According to Ax.19, p_0 and p_1 are consequently coincident, as are p_0 and p_3. This means that the application of Ax.27 for the series {F_0 F_1 F_2} does not condition p_2 at all and is superfluous. But since points p_6 and p_7 are coincident and points p_8 and p_9 are coincident, axiom Ax.27 also holds for the series {F_4 F_1 F_3} which in turn constrains the position of p_2 (figure 102).

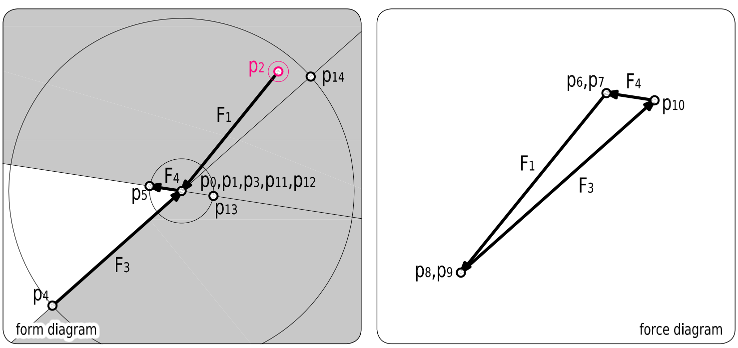


figure 102
behaviour of Ax.27 when the first and the third of the three consecutive forces are a zero force.

(4) Axiom Ax.27 also remains consistent when all the forces applied on a point are zero forces. Indeed, Ax.27 is totally superfluous in that case since it is always satisfied:

Ax.27

$$\begin{aligned}
&\longrightarrow (\mathrm{Force}[F_0\ p_0\ p_0\ p_1\ p_1] \land \mathrm{Force}[F_1\ p_0\ p_0\ p_1\ p_1] \land \mathrm{Force}[F_2\ p_0\ p_0\ p_1\ p_1] \\
&\qquad \longrightarrow \exists p_2 p_3 : (\quad (\mathrm{Laterality}[p_0\ p_0\ p_0\ p_0] \land \mathrm{Laterality}[p_0\ p_0\ p_0\ p_2]) \\
&\qquad\qquad\qquad \lor (\mathrm{Laterality}[p_0\ p_0\ p_0\ p_0] \land \mathrm{Laterality}[p_0\ p_0\ p_0\ p_0]) \\
&\qquad\qquad\qquad \lor (\mathrm{Laterality}[p_0\ p_0\ p_3\ p_0] \land \mathrm{Laterality}[p_0\ p_0\ p_0\ p_0])) \\
&\qquad\qquad \land \mathrm{Laterality}[p_2\ p_0\ p_0\ p_0] \land \mathrm{Laterality}[p_2\ p_0\ p_0\ p_0] \\
&\qquad\qquad \land \mathrm{Proximity}[p_0\ p_2\ p_0\ p_0] \land \mathrm{Proximity}[p_0\ p_0\ p_0\ p_2] \\
&\qquad\qquad \land (\neg\mathrm{Proximity}[p_0\ p_0\ p_0\ p_0] \lor \neg\mathrm{Proximity}[p_2\ p_0\ p_0\ p_0] \\
&\qquad\qquad\qquad\qquad \lor \mathrm{Proximity}[p_2\ p_0\ p_0\ p_0]) \\
&\qquad\qquad \land (\mathrm{Proximity}[p_0\ p_0\ p_0\ p_0] \lor \neg\mathrm{Laterality}[p_2\ p_0\ p_0\ p_0] \\
&\qquad\qquad\qquad\qquad \lor \mathrm{Proximity}[p_2\ p_0\ p_0\ p_0]) \\
&\qquad\qquad \land \mathrm{Laterality}[p_3\ p_0\ p_0\ p_0] \land \mathrm{Laterality}[p3\ p_0\ p_0\ p_0] \\
&\qquad\qquad \land \mathrm{Proximity}[p_0\ p_3\ p_0\ p_0] \land \mathrm{Proximity}[p_0\ p_0\ p_0\ p_3] \\
&\qquad\qquad \land (\neg\mathrm{Proximity}[p_0\ p_0\ p_0\ p_0] \lor \neg\mathrm{Proximity}[p_3\ p_0\ p_0\ p_0] \\
&\qquad\qquad\qquad\qquad \lor \mathrm{Proximity}[p_3\ p_0\ p_0\ p_0]) \\
&\qquad\qquad \land (\mathrm{Proximity}[p_0\ p_0\ p_0\ p_0] \lor \neg\mathrm{Laterality}[p_3\ p_0\ p_0\ p_0] \\
&\qquad\qquad\qquad\qquad \lor \mathrm{Proximity}[p_3\ p_0\ p_0\ p_0])) \\
&\longleftrightarrow (\mathrm{true} \longrightarrow \exists p_2 p_3 : (\mathrm{Laterality}[p_0\ p_0\ p_0\ p_2] \lor \mathrm{Laterality}[p_0\ p_0\ p_3\ p_0]) \\
&\qquad\qquad \land \mathrm{Proximity}[p_0\ p_2\ p_0\ p_0] \land \mathrm{Proximity}[p_0\ p_3\ p_0\ p_0]) \\
&\longleftrightarrow (\mathrm{true} \longrightarrow \mathrm{Laterality}[p_0\ p_0\ p_0\ p_0] \lor \mathrm{Laterality}[p_0\ p_0\ p_0\ p_0]) \\
&\longleftrightarrow \mathrm{true}
\end{aligned}$$

(5) When only two forces are applied on a point, the concept of a reading cycle is meaningless and axiom Ax.27 must always be verified. The following statements prove it (figure 103):

Ax.27

$$\begin{aligned}
&\longrightarrow (\mathrm{Force}[F_0\ p_0\ p_1\ p_3\ p_4] \land \mathrm{Force}[F_1\ p_0\ p_2\ p_4\ p_3] \land \mathrm{Force}[F_0\ p_0\ p_1\ p_3\ p_4] \\
&\qquad \longrightarrow \exists p_5 p_6 : (\quad (\mathrm{Laterality}[p_2\ p_0\ p_1\ p_0] \land \mathrm{Laterality}[p_2\ p_0\ p_0\ p_5]) \\
&\qquad\qquad\qquad \lor (\mathrm{Laterality}[p_2\ p_0\ p_1\ p_0] \land \mathrm{Laterality}[p_2\ p_0\ p_0\ p_1]) \\
&\qquad\qquad\qquad \lor (\mathrm{Laterality}[p_2\ p_0\ p_6\ p_0] \land \mathrm{Laterality}[p_2\ p_0\ p_0\ p_1])\ ) \\
&\qquad\qquad \land \mathrm{Laterality}[p_5\ p_0\ p_0\ p_1] \land \mathrm{Laterality}[p_5\ p_0\ p_1\ p_0] \\
&\qquad\qquad \land \mathrm{Proximity}[p_0\ p_5\ p_0\ p_1] \land \mathrm{Proximity}[p_0\ p_1\ p_0\ p_5] \\
&\qquad\qquad \land (\neg\mathrm{Proximity}[p_1\ p_0\ p_0\ p_0] \lor \neg\mathrm{Proximity}[p_5\ p_1\ p_1\ p_1] \\
&\qquad\qquad\qquad\qquad \lor \mathrm{Proximity}[p_5\ p_0\ p_0\ p_0]) \\
&\qquad\qquad \land (\mathrm{Proximity}[p_1\ p_0\ p_0\ p_0] \lor \neg\mathrm{Laterality}[p_5\ p_0\ p_1\ p_0] \\
&\qquad\qquad\qquad\qquad \lor \mathrm{Proximity}[p_5\ p_0\ p_0\ p_0])
\end{aligned}$$

$$\wedge\ \text{Laterality}[p_6\ p_0\ p_0\ p_1] \wedge \text{Laterality}[p_6\ p_0\ p_1\ p_0]$$
$$\wedge\ \text{Proximity}[p_0\ p_6\ p_0\ p_1] \wedge \text{Proximity}[p_0\ p_1\ p_0\ p_6]$$
$$\wedge\ (\ \neg\text{Proximity}[p_1\ p_0\ p_0\ p_0] \vee \neg\text{Proximity}[p_6\ p_1\ p_1\ p_1]$$
$$\vee\ \text{Proximity}[p_6\ p_0\ p_0\ p_0])$$
$$\wedge\ (\ \text{Proximity}[p_1\ p_0\ p_0\ p_0] \vee \neg\text{Laterality}[p_6\ p_0\ p_0\ p_1]$$
$$\vee\ \text{Proximity}[p_6\ p_0\ p_0\ p_0])\ )$$

$$\longleftrightarrow (\ \text{Force}[F_0\ p_0\ p_1\ p_3\ p_4] \wedge \text{Force}[F_1\ p_0\ p_2\ p_4\ p_3]$$
$$\longrightarrow \exists p_5 p_6 \text{: } (\quad (\ \text{Laterality}[p_2\ p_0\ p_1\ p_0] \wedge \text{Laterality}[p_2\ p_0\ p_0\ p_5]\ )$$
$$\vee\ (\ \text{Laterality}[p_2\ p_0\ p_1\ p_0] \wedge \text{Laterality}[p_2\ p_0\ p_0\ p_1]\ )$$
$$\vee\ (\ \text{Laterality}[p_2\ p_0\ p_6\ p_0] \wedge \text{Laterality}[p_2\ p_0\ p_0\ p_1]\ )\ )$$
$$\wedge\ \text{Proximity}[p_5\ p_2\ p_2\ p_2] \wedge \text{Proximity}[p_6\ p_2\ p_2\ p_2]\ )$$

$$\longleftrightarrow (\ \text{Force}[F_0\ p_0\ p_1\ p_3\ p_4] \wedge \text{Force}[F_1\ p_0\ p_2\ p_4\ p_3]$$
$$\longrightarrow (\ \text{Laterality}[p_2\ p_0\ p_1\ p_0] \wedge \text{Laterality}[p_2\ p_0\ p_0\ p_2]\ )$$
$$\vee\ (\ \text{Laterality}[p_2\ p_0\ p_1\ p_0] \wedge \text{Laterality}[p_2\ p_0\ p_0\ p_1]\ )$$
$$\vee\ (\ \text{Laterality}[p_2\ p_0\ p_2\ p_0] \wedge \text{Laterality}[p_2\ p_0\ p_0\ p_1]\ )\ )$$

$$\longleftrightarrow (\text{Force}[F_0\ p_0\ p_1\ p_3\ p_4] \wedge \text{Force}[F_1\ p_0\ p_2\ p_4\ p_3])$$
$$\longrightarrow (\ \text{Laterality}[p_2\ p_0\ p_1\ p_0] \wedge \text{Laterality}[p_2\ p_0\ p_0\ p_1]\ )$$

$$\longrightarrow \text{true}$$

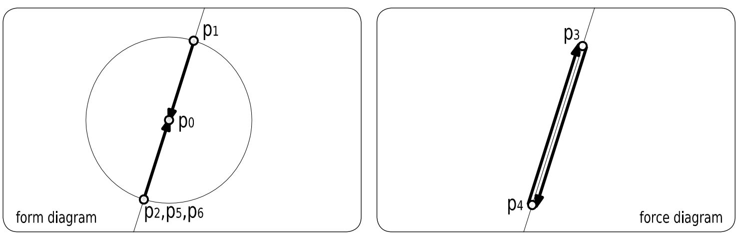


figure 103 behaviour of Ax.27 when there are only two forces applied on a point.

In short, these five particular cases shed light on the internal geometric reasons of axiom Ax.27. The following paragraph shows that the local reading cycles expressed in Ax.27 involve global reading cycles and the paragraph after that identifies the extent to which uniform global reading cycles are mandatory.

why imposing a uniform reading cycle locally involves a uniform reading cycle globally · This paragraph explains how Ax.27 compels the forces applied on any strut-and-tie sub-network to be read clockwise in the form diagram, according to the sequence in which they follow one another in the force diagram. Hence, it shows how a reading cycle condition applied on every minimum force sub-network influences any non-minimum strut-and-tie sub-network containing them.

Given two points p_A and p_B and given two forces F_i and F_j respectively applied on p_A and p_B such that they are sufficiently compatible to form a rod, two cases might arise. Either (1) points p_A and p_B belong to two different strut-and-tie networks (figure 104) or (2) they belong to a unique strut-and-tie network (figure 106, page 117). The following considerations show that, in both cases, if the reading cycle of the forces applied on the sub-network(s) is clockwise when F_i and F_j are considered as two independent forces, then the reading cycle of the forces applied on the new network, obtained by assimilating F_i and F_j as a rod, is also clockwise.

(1) First case (figure 104). If the forces applied on the force sub-network containing F_i are read clockwise in the form diagram, the previous force F_{i-1} and the next force F_{i+1} can be identified in both diagrams. Likewise, if the forces applied on the force sub-network containing F_j are read clockwise in the form

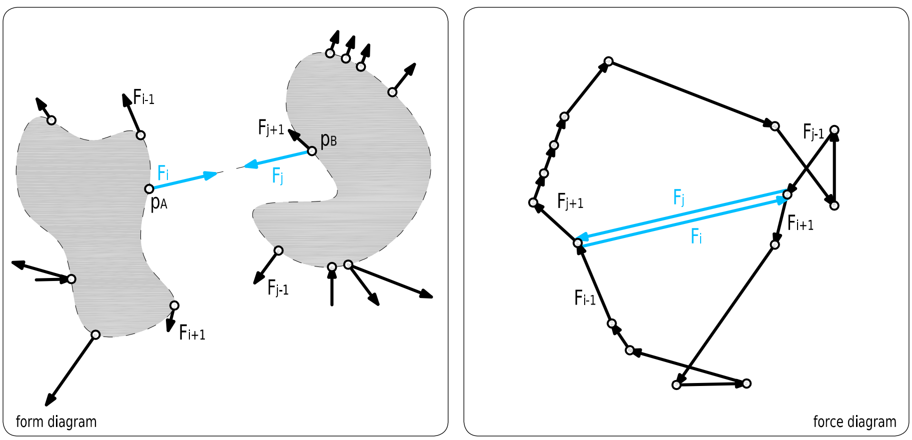


figure 104
first case before treating forces F_i and F_j as a rod.

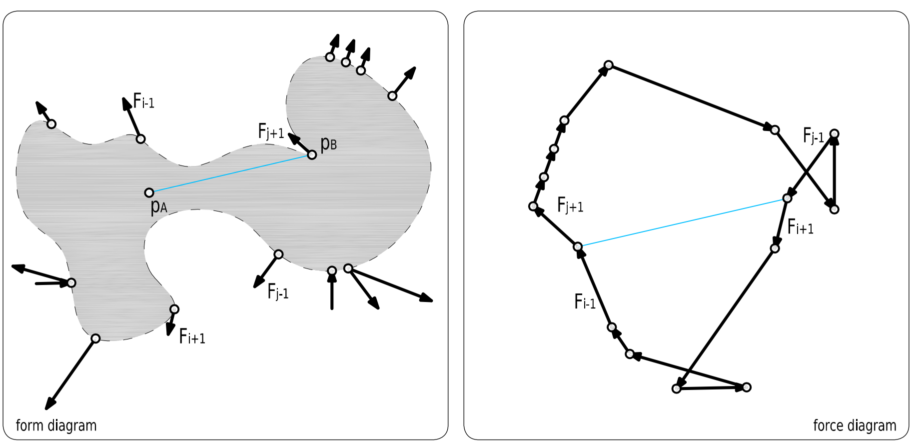


figure 105
first case after transforming forces F_i and F_j into a rod.

diagram, the previous force F_{j-1} and the next force F_{j+1} can be identified in both diagrams. The sequence of forces from F_{i+1} to F_{i-1} is named S_1 and the sequence of forces from F_{j+1} to F_{j-1} is named S_2.

As forces F_i and F_j are aligned in the form diagram and are of equal orientation and magnitude, they can be identified as a rod (figure 105). Whatever the sequences S_1 and S_2, the following properties still hold in the force diagram once the forces F_i and F_j are identified as a rod:

(a) sequence S_1 spans from F_{i+1} to F_{i-1}
(b) sequence S_2 spans from F_{j+1} to F_{j-1}
(c) force F_{j-1} is read just before F_{i+1}
(d) force F_{i-1} is read just before F_{j+1}

Since the same four properties can also be identified in the form diagram, the reading cycle of the forces acting on the global force network is indeed clockwise.

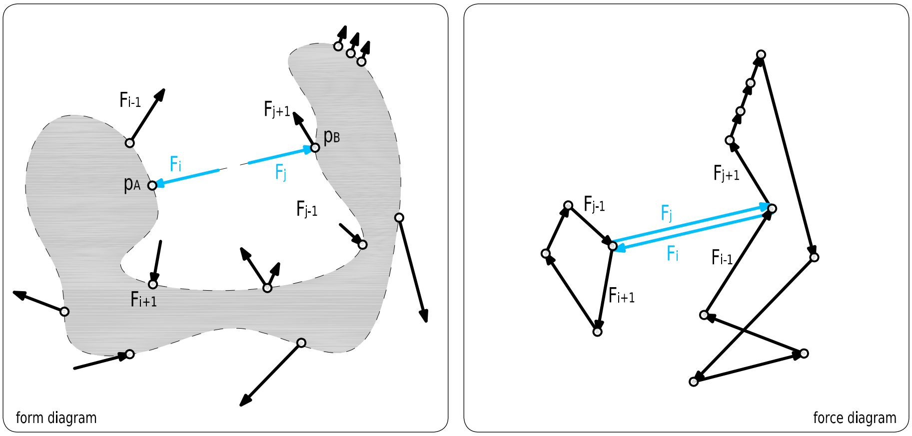
(2) Second case (figure 106). If the forces applied on the force sub-network containing F_i and F_j are read clockwise in the form diagram, forces F_{i-1}, F_{i+1}, F_{j-1} and F_{j+1} can be identified such that F_i succeeds F_{i-1} and precedes F_{i+1} and such that F_j succeeds F_{j-1} and precedes F_{j+1}. The sequence of forces from F_{i+1} to F_{j-1} is named S_3 and the sequence of forces from F_{j+1} to F_{i-1} is named S_4.

As forces F_i and F_j are aligned in the form diagram and are of equal orientation and magnitude, they can be identified as a rod (figure 107). Whatever the sequences S_3 and S_4, the following properties still hold in the force diagram once the forces F_i and F_j are identified as a rod:

(a) sequence S_3 spans from F_{i+1} to F_{j-1}
(b) sequence S_4 spans from F_{j+1} to F_{i-1}
(c) force F_{j-1} is read just before F_{i+1}
(d) force F_{i-1} is read just before F_{j+1}

Since the same four properties can also be identified in the form diagram, the reading cycle of the forces acting on the global force network is indeed clockwise.

In conclusion, the following two facts jointly prove that any global strut-and-tie network always presents a clockwise reading cycle when each minimum-force network presents a clockwise reading cycle:



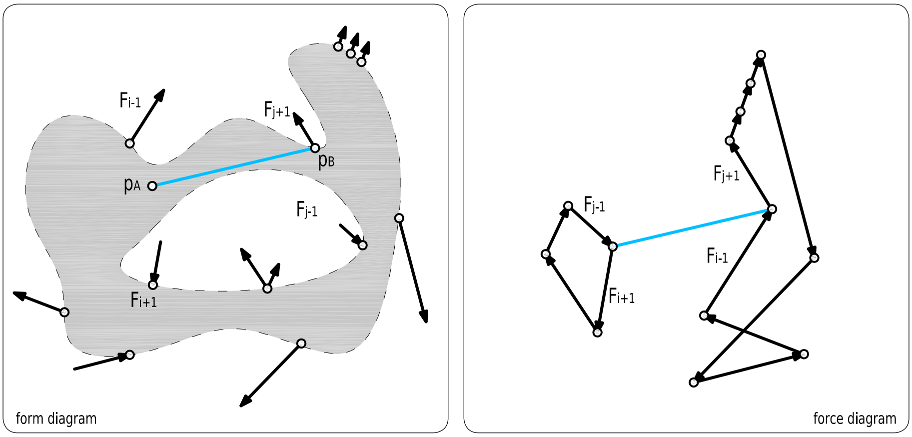


figure 106 second case before considering forces F_i and F_j as a rod.

figure 107 second case after transforming forces F_i and F_j into a rod.

(a) any strut-and-tie network can be seen as a combination of all the minimum force networks it contains (thanks to the definition of the Rod relationship)

(b) when compatible forces are identified as rods, each of these minimum force networks reacts exactly as the initial sub-networks containing p_A and p_B shown in the two cases above.

Consequently, as soon as Ax.27 is verified for each minimum force network, a clockwise global reading cycle is ensured for any strut-and-tie network.

A practical example is shown in figure 109, figure 110 and figure 111. They are three possible strut-and-tie sub-networks of the truss depicted in figure 108. Each reading cycle is clockwise, regardless of the sub-network considered.

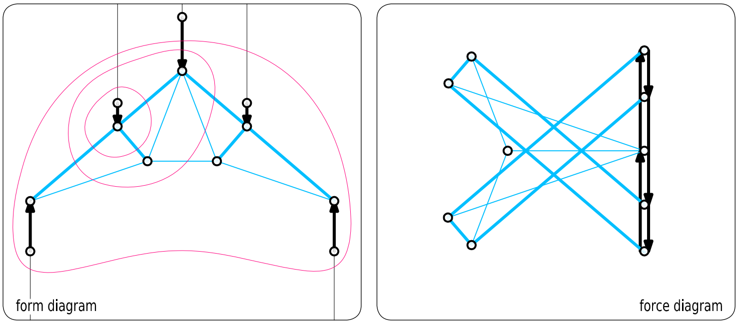


figure 108
form and force diagrams of a truss; hulls correspond to the strut-and-tie sub-networks detailed in the three following figures.

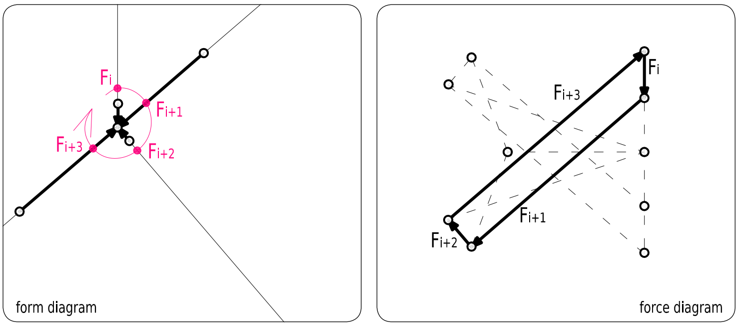


figure 109
clockwise reading cycle of the first strut-and-tie sub-network.

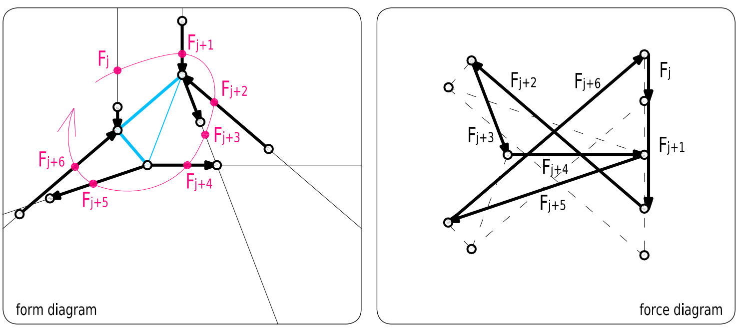


figure 110
clockwise reading cycle of the second strut-and-tie sub-network.

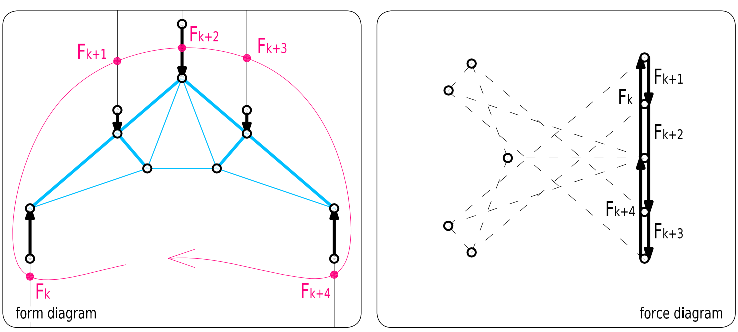


figure 111
clockwise reading cycle of the third strut-and-tie sub-network.

historical review of reciprocity in graphic statics · Prior to expressing the need for Ax.27, a historical review of what reciprocity means in graphic statics is given here. The reciprocal nature of form and force diagrams has undergone subtle developments over the years.

The first identifications of reciprocity were the ones stated by William John Macquorn Rankine in 1858 and James Clerk Maxwell in 1864 (figure 32, page 49 and figure 112):

> *"Two plane figures are reciprocal when they consist of an equal number of lines, so that corresponding lines in the two figures are parallel, and corresponding lines which converge to a point in one figure form a closed polygon in the other."* (Maxwell·1864, page 251)

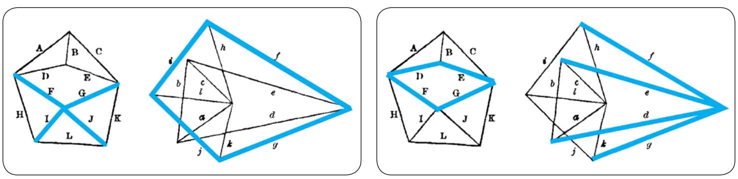


figure 112 illustration of the reciprocal rule for two different polygons (reworked figure from Maxwell·1864, page 253, figure 2).

Studying a strut-and-tie network and its inner forces with two reciprocal closed diagrams, such as those defined by Maxwell soon becomes tedious since external forces applied to the strut-and-tie are not drawn. This shortcoming was corrected immediately by Karl Culmann (Culmann·1866), Luigi Cremona (Cremona·1868) and James Clerk Maxwell himself (Maxwell·1867) by adding the representation of forces within both diagrams (figure 113).

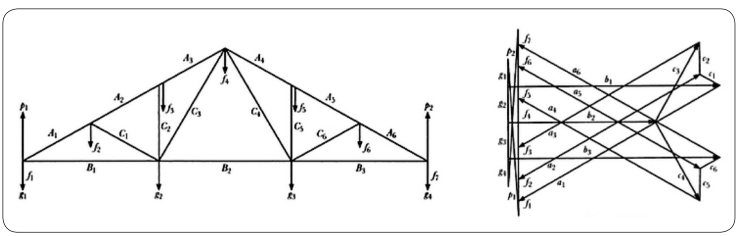


figure 113 two reciprocal diagrams according to Maxwell·1867 (figure copied from Maxwell/...·1995 pages 315 and 316).

This inclusion has the consequence of opening the form diagram. Indeed, each force in the form diagram is borne by an infinite half-line. If each polygon in the force diagram is to be linked to a continuous set of rods and forces in the form diagram, then this set is sometimes a closed polygon, sometimes a region open to infinity. This is the reason why the definition of reciprocity given by Rankine and Maxwell was soon replaced by new conditions. For instance, the following set of conditions for reciprocity is due to Robert Henry Bow (Bow·1873, page 52):

(1) *"Corresponding lines, whether representing constituent parts of the frame, or external forces, which meet at a point in the frame, form a closed polygon in the* [force] *diagram.*

(2) *Corresponding lines which represent constituent parts of the frame, and form a closed polygon in it, meet in a point in the* [force] *diagram.*

(3) *The lines representing all the external forces acting on the frame should form a closed polygon in the* [force] *diagram.*

(4) *Lines — some of which represent external forces — which meet in a point in the* [force] *diagram, have the corresponding lines contiguous in the frame; but these may form a partial boundary to an infinite area* [...]."

In other words, these four conditions define reciprocity as a bijection between polygons or open areas and intersections of forces and rods (figure 114 and figure 115). However, they do not prevent a rod from being represented by more than one line in the force diagram or, in other words, they do not prevent opposite forces forming a rod from being nonadjacent in the force diagram.

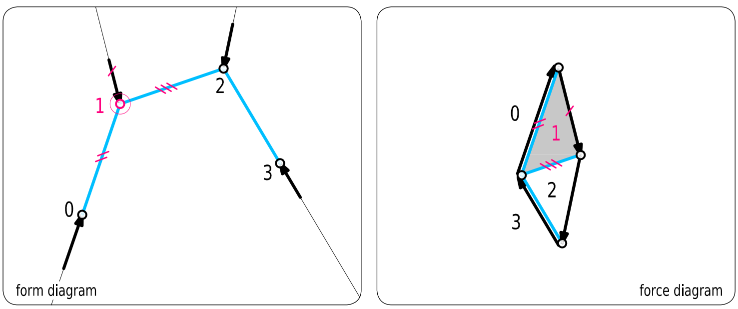


figure 114 classical nature of reciprocity; intersection in the form diagram and corresponding closed polygon in the force diagram.

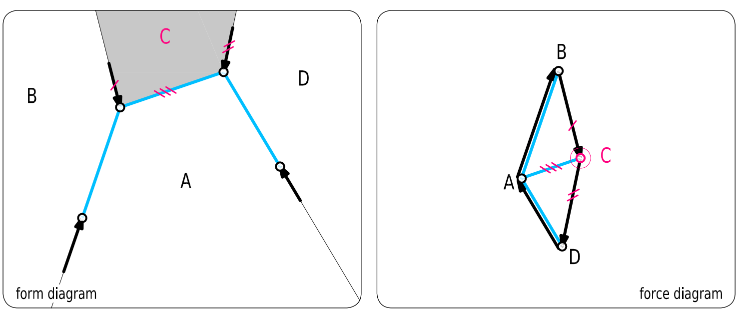


figure 115 classical nature of reciprocity; intersection in the force diagram and corresponding open area in the form diagram.

Bow consequently inserted a fifth condition to this list:

(5) *"There should be only one line in the diagram of forces to represent any one force acting on the frame or in a part of it."*

Unlike the first four conditions, the fifth condition is not required for static equilibrium. It is just added for convenience: (a) corresponding force polygons are identified more easily, (b) diagrams are more compact and confusion between lines is avoided, and (c) the identification of rods in the force polygon is univocal since it is always equivalent to two opposite forces.

As clarified by Albert Pirard in 1950 (Pirard·1950, page 91), preventing forces from being drawn more than once in the force diagram, *i.e.* to guarantee Bow's fifth condition, can only be achieved if the following rule is observed: every polygon constituted of forces and rods in the force diagram must be read in an order that is identical to the order obtained when those forces and rods are read in the form diagram, either always clockwise or always anti-clockwise. The choice of clockwise or anti-clockwise order is a convention. What matters here is that it is always the same for every minimum force network.

The need for this additional rule can be illustrated with the figure 116 from Pirard·1950. Diagrams (II), (III) and (IV) are the force polygons corresponding to points A, D and E, respectively in the form diagram (I). Since these three sets of forces are read clockwise in the form diagram, their combination pro-

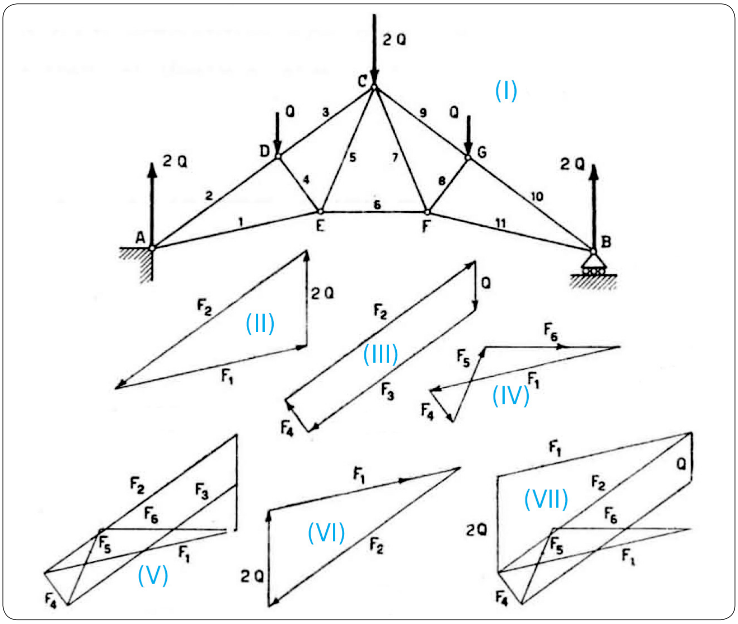


figure 116
obligation to have a uniform reading cycle for each force polygon in order to prevent duplication of force in the force diagram; figure from Pirard·1950, page 91.

duces a force diagram (V) where each pair of forces corresponding to one rod is superimposed. If, however, one set of forces (VI) is read anti-clockwise in the form diagram, the combination of multiple force polygons (VII) duplicates some lines representing one rod, *e.g.* F_1, the force acting on rod 1, is represented twice in (VII).

the need for uniform reading cycles towards reciprocal diagrams · As far as the axiomatisation undergone in this section is concerned, the choice is made to make form and force diagrams fulfil the classical conditions for reciprocity, as for instance those stated by Bow·1873 (page 52). Axioms Ax.23 and Ax.24 actually already ensure static equilibrium and bidirectional relationships between the intersection of forces and corresponding open areas or closed polygons, *i.e.* they are equivalent to the first four conditions of Bow·1873, page 52. Axiom Ax.27 is therefore only meant to guarantee the uniqueness of forces in the force diagram, *i.e.* the fifth condition of Bow·1873.

In other words, Ax.27 is meant to ensure that if two opposite forces are of equal magnitudes and are aligned in the form diagram, they are then superimposed in the force diagram. So, if two forces are compatible enough to be assimilated into a rod in the form diagram, they are also compatible for being assimilated into a rod in the force diagram. Therefore Ax.27 guarantees that the definition of the non-fundamental Rod relationship (page 91) contributes to the reciprocity of diagrams.

Without Ax.22, the visual expressiveness of pairs of forces capable of forming a rod would be less direct since it would imply the variation of the order in which forces succeed one another in each force polygon until one order means that all the compatible forces are superimposed.

Moreover, Ax.27 also makes the uniform reading cycle clockwise rather than anti-clockwise.

As previously noted, the existence of Ax.27 expresses the need to make use of the points that define the application type of forces — the second parameter of each Force[p_0 p_1 p_2 p_3] relationship. See the paragraph entitled “an axiom to constrain the point that defines the application type of force” (page 82).

why the definition of rod does not invalidate reciprocal rules · When the Rod relationship is defined in the paragraph entitled "geometric definition of a rod" (page 87), a choice has been made to compel the two opposite forces in the form diagram to be placed between the two points of applications (figure 72, page 90) rather than outside them (figure 73, page 90) — see condition (5) page 89 . The reason for this choice is due to the need for a global uniform reading cycle as imposed by Ax.27 — see the paragraph entitled "why imposing a uniform reading cycle locally involves a uniform reading cycle globally" (page 114).

This can be shown by comparing two general networks involving a rod. F_i and F_j are the two forces that form that rod. In figure 117 the points defining their type of application, *i.e.* p_1 and p_5, are placed outside the extreme points of the rods, *i.e.* p_0 and p_4. In figure 118 they are placed between them. Owing to Ax.27, local uniform reading cycles must be clockwise and forces F_{i-1}, F_{i+1}, F_{j-1} and F_{j+1} can be localised in both diagrams so that they are respectively before and after F_i and F_j. However Ax.27 also leads to global uniform reading cycles. As far as the global strut-and-tie network (including p_0 and p_4) is concerned, the force diagram shows that in both cases, F_{j-1} comes before F_{i+1} and F_{i-1} comes before F_{j+1}. However, only the case in figure 118 verifies this order when reading the forces clockwise in the form diagram. This proves that

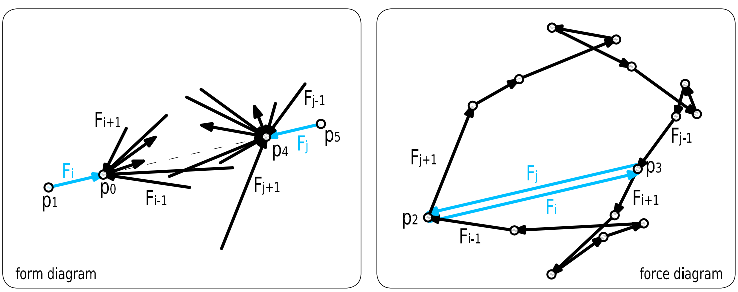


figure 117
two forces F_i and F_j unable to form a rod without breaking the reading cycle.

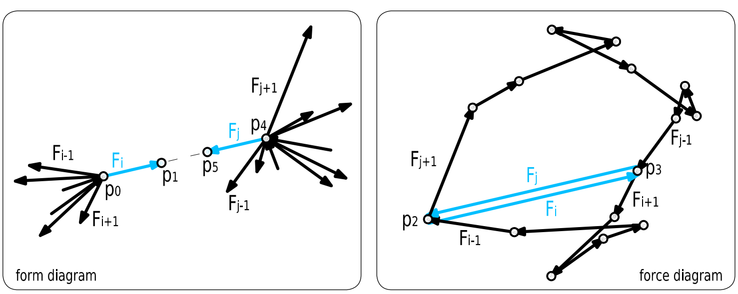


figure 118
two forces F_i and F_j able to form a rod without breaking the reading cycle.

the points defining the type of application of two forces forming a rod must always be placed between their application points, regardless of whether the forces exert a pull or a push.

why a uniform reading cycle imposes the absence of almost any intersection of rods in the space diagram · As a result of axioms Ax.19, Ax.24 and Ax.27, it is impossible to have a strut-and-tie network that presents (1) the crossing of two rods in the form diagram, or (2) the crossing of a rod with a half-line of force in the form diagram, unless it is possible to move the points in the form diagram so that (a) the altered strut-and-tie network no longer contains an intersection of this kind, and (b) the reading cycles of the altered strut-and-tie network remain unchanged. However, the crossing of two half-lines of force inside a unique strut-and-tie network does not endanger the reciprocity of both diagrams.

The only way to build these crossings is to divide the rod concerned in two — the parallelism of the two parts of the rod being ensured by appropriate geometric relationships — or to duplicate the forces concerned.

The six following examples outline these various scenarios and define means of rectification:

(1) an impossible strut-and-tie network due to the crossing of two rods in the form diagram
(2) an impossible strut-and-tie network due to the crossing of a rod with a half-line of force in the form diagram
(3) a possible strut-and-tie network where two rods cross without being divided
(4) a possible strut-and-tie network where one rod crosses one half-line of force without being duplicated
(5) a possible strut-and-tie network where half-lines of force cross one another
(6) two possible strut-and-tie network where their rods cross without being divided.

(1) The strut-and-tie network in figure 119 does not satisfy the axiomatic set. Evidence of this is that it is impossible to move the force polygons in figure 120 in order to superimpose pairs of forces aimed at being replaced by rods. The solution is to divide the crossing rods into two. This has the effect of adding a new point in the form diagram — p_A — and a new force polygon in the force diagram (figure 121).

figure 119
a strut-and-tie network that do not satisfy the axiomatic set.

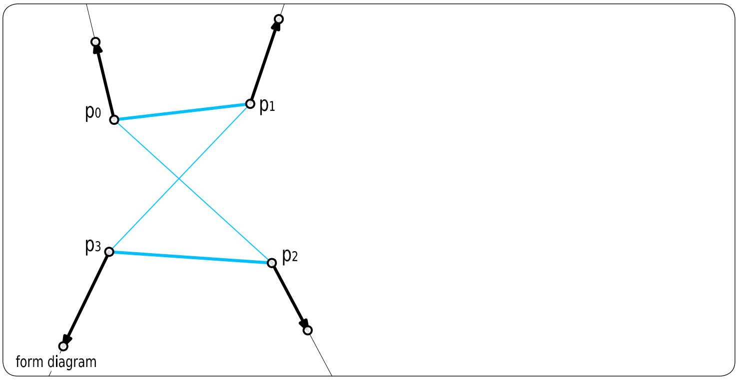


figure 120
these force polygons cannot be joined to form rods.

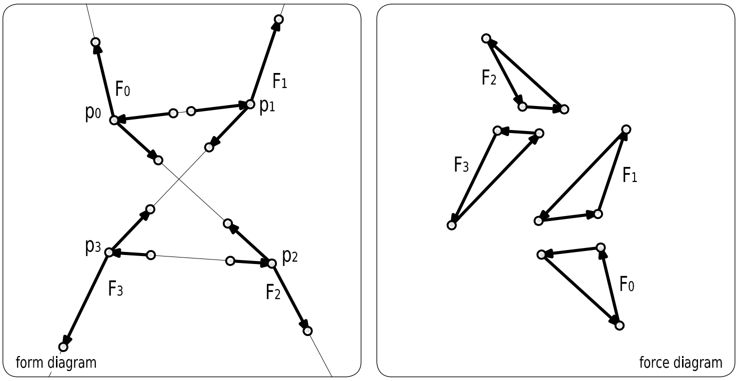
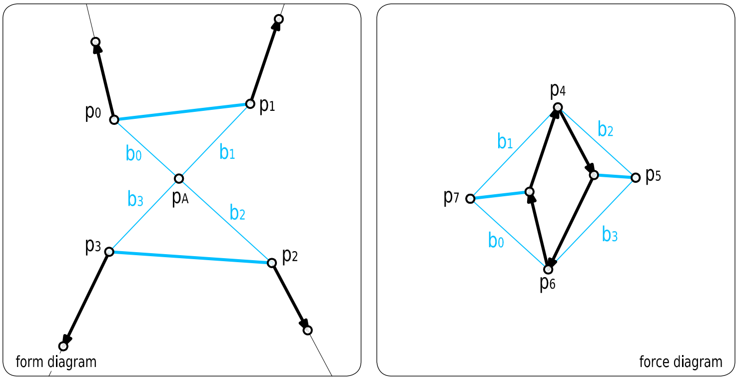


figure 121
adding a point at the intersection of crossing rods makes the network compatible with the axiomatic set.



In order to ensure that the four rods b_0, b_1, b_2 and b_3 act as two crossing rods, the following geometric statement must hold:

Collinearity[p_1 p_3 p_A] ∧ Collinearity[p_0 p_2 p_A]

which means that:

Parallelism[p_4 p_5 p_6 p_7] ∧ Parallelism[p_4 p_7 p_5 p_6]

and that:

Equidistance[p_4 p_5 p_6 p_7] ∧ Equidistance[p_4 p_7 p_5 p_6]

(2) The strut-and-tie network in figure 122 does not satisfy the axiomatic set either. The minimum force networks in figure 123 cannot be repositioned in order to allow the superposition of the pairs of forces aimed to act as rods. The solution shown in figure 124 consists of adding a force polygon by duplicating force F_0 and by dividing the rod crossed by its half-line.

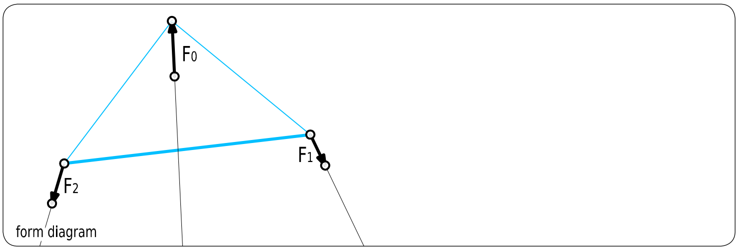


figure 122
a strut-and-tie network that do not satisfy the axiomatic set.

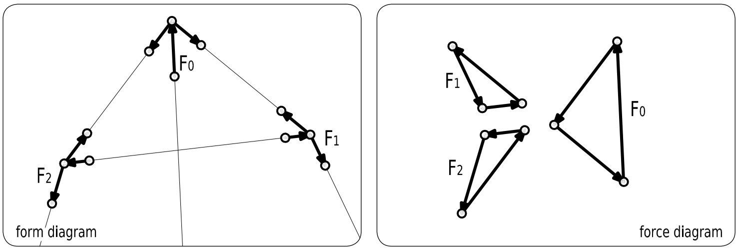


figure 123
these force polygons cannot be joined to form rods.

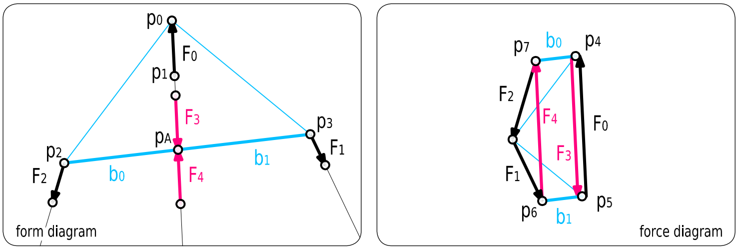


figure 124
adding a point at the intersection of the rod with the half-line of force makes the network compatible with the axiomatic set.

Again, the new point p_A should satisfy the following geometric statement:

Collinearity[p_0 p_1 p_A] $\wedge$ Collinearity[p_2 p_3 p_A]

which means that:

Parallelism[p_4 p_5 p_6 p_7] $\wedge$ Parallelism[p_4 p_7 p_5 p_6]

and that:

Equidistance[p_4 p_5 p_6 p_7] $\wedge$ Equidistance[p_4 p_7 p_5 p_6]

(3) The strut-and-tie network in figure 125 satisfies the axiomatic set although two rods cross. The reason for this is explained in figure 126 and is due to the fact that points p_A and p_B can be moved in the form diagram such that (a) the rods no longer cross and (b) the force diagram remains unchanged, *i.e.* the magnitudes and orientations of forces remain equal.

(4) The example in figure 127 is similar to the previous one. Two pairs of rods cross and a half-line of force crosses a rod. However, points p_A and p_B can be moved in the form diagram such that these crossings vanish without altering the force diagram. The strut-and-tie network in figure 128 and its force diagram consequently satisfy the axiomatic set.

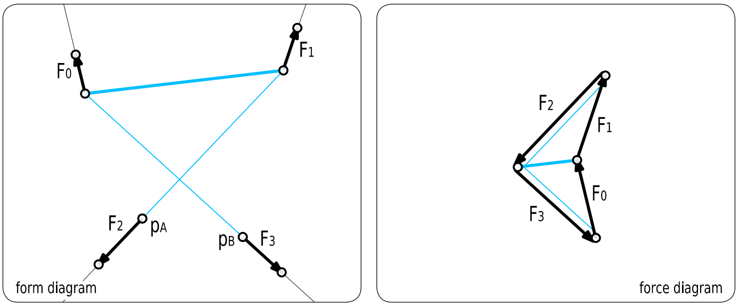


figure 125
a strut-and-tie network with two intersecting rods that does not have to be divided.

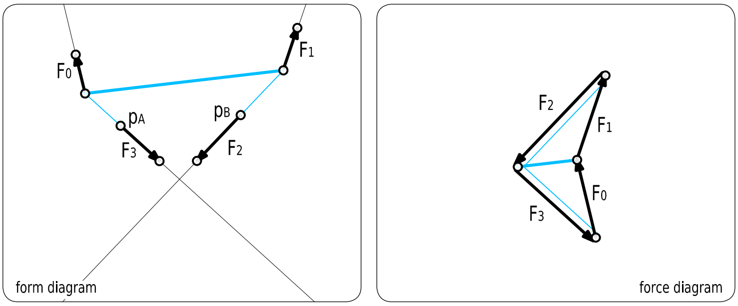


figure 126
other positions of p_A and p_B: rods do not cross and the force diagram remains identical.

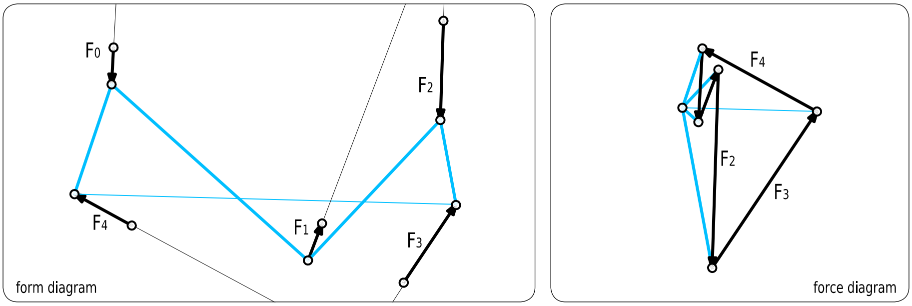


figure 127

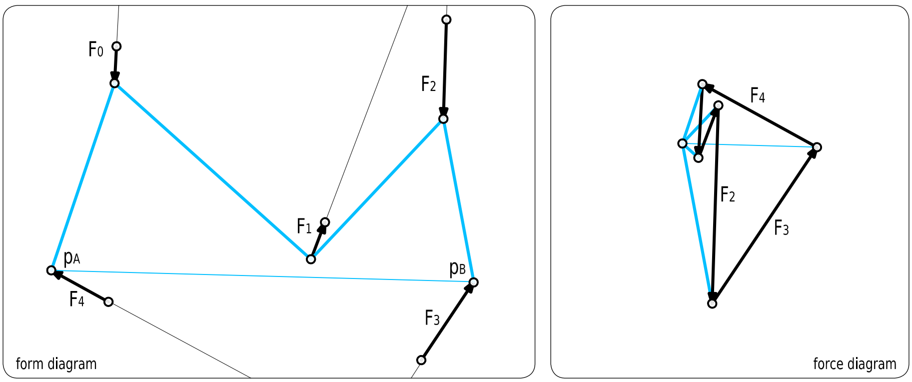


figure 128

(5) The example in figure 129 shows that the intersection of half-lines of forces with half-lines of forces does not prevent either diagram being in compliance with the axiomatic set.

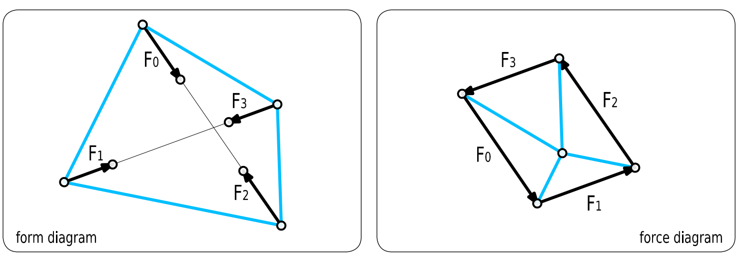


figure 129
a strut-and-tie network with intersections of half-lines of forces.

(6) Two rods intersect in the form diagram in figure 130. However, this is not an issue since they belong to two separate strut-and-tie networks, each with its own force diagram.

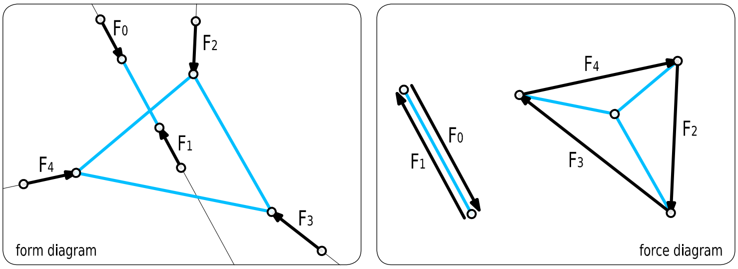


figure 130
two strut-and-tie networks with an intersection of rods.

how a uniform reading cycle allows multiple form and force diagrams to have the same structural configuration · Cases (3) and (4) have shown that some intersections of rods and forces did not prevent the satisfaction of the axiomatic set, and did not require rods to be divided and forces to be duplicated (figure 125 and figure 127). However, it is still possible to divide these rods and duplicate these forces. This means that the same structural configuration sometimes allows different form and force diagrams (figure 131 and figure 132).

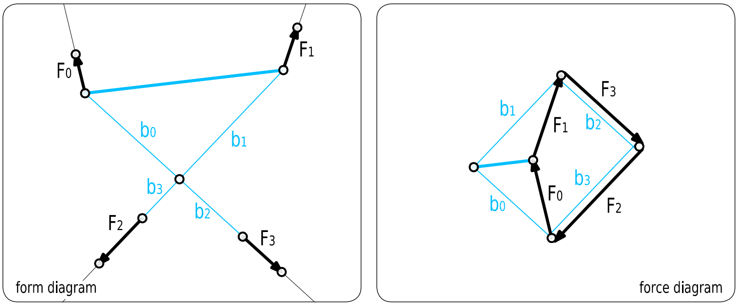


figure 131
copy of the strut-and-tie network from figure 125 after dividing the crossing rods.

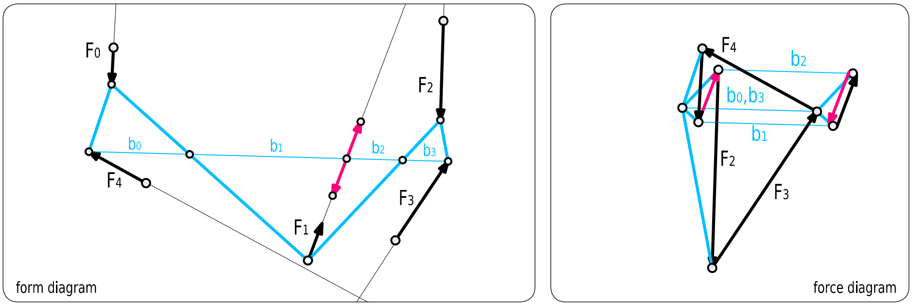


figure 132
copy of the strut-and-tie network from figure 127 after duplicating crossing rods and forces.

how a uniform reading cycle implies the distinction between pulls and pushes · A direct corollary of not being able to cross rods and half-line of forces is that a strut-and-tie network might present a different force diagram depending on a force is pulling or pushing. The distinction between pulls and pushes is therefore fundamental to the proposed axiomatic set. An example is shown in figure 133 and figure 134.

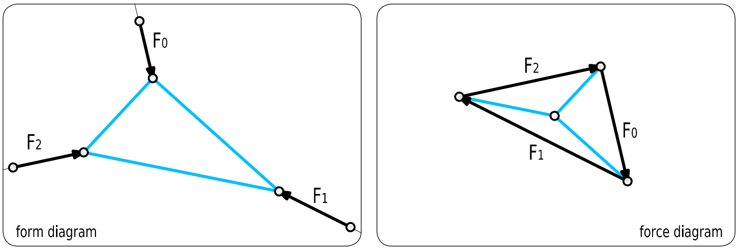


figure 133
a strut-and-tie network on which force F_1 exerts a pull.

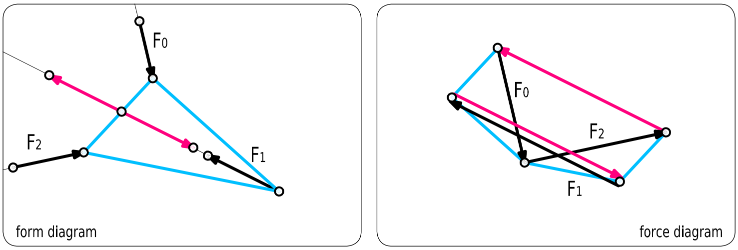


figure 134
a strut-and-tie network on which the same force F_1 exerts a push.

how the studied hull influences the uniform reading cycle · Reading cycles concern the forces that are applied on a given sub-network. It has been seen that these sub-networks can be identified using hulls whose definition is given by the set of points contained in the sub-network — see the “sub-network hulls” (page 95). Hulls can actually be distinguished further by the hidden intersections of forces they encompass or not.

For example, figure 136, figure 137 and figure 138 show various hulls that all concern the entire global network in figure 135. However, as a consequence of the fact that the external forces intersecting these hulls must be read clockwise in the form diagram, they lead to different force diagrams.

figure 135
a strut-and-tie network.

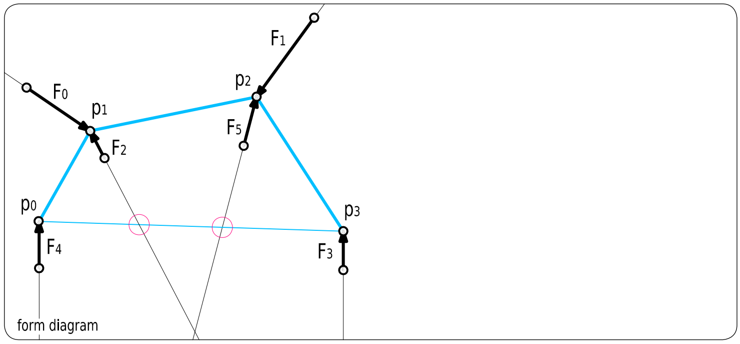


figure 136
minimum hull containing the entire strut-and-tie network from figure 135.

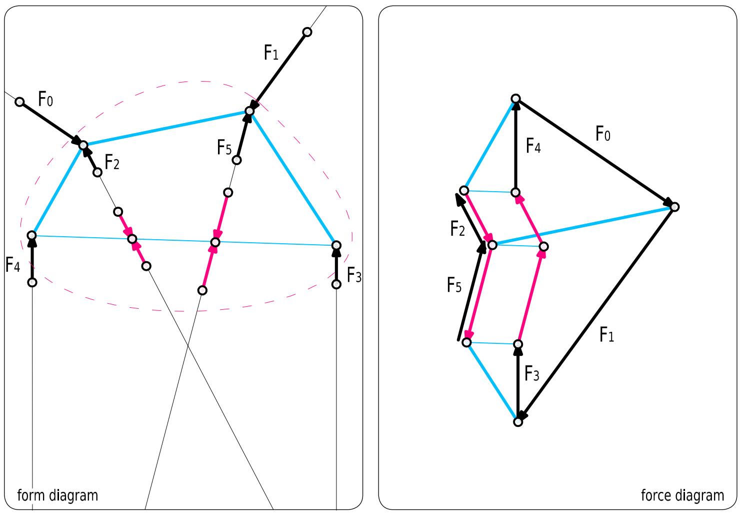
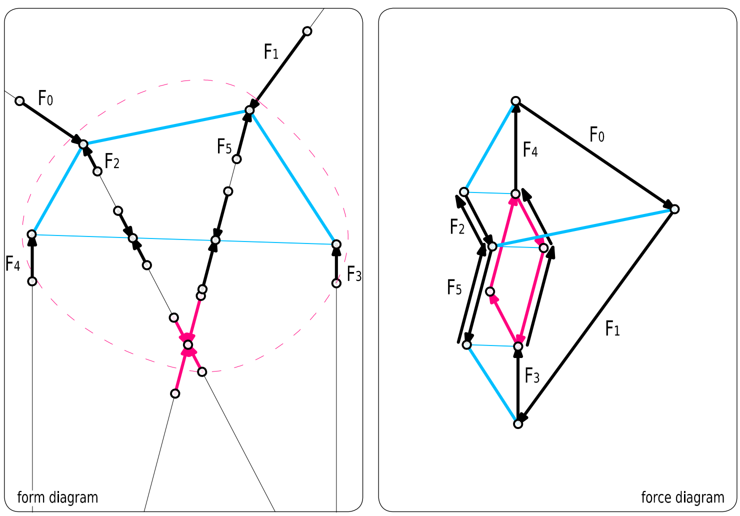


figure 137
intermediate hull containing the entire strut-and-tie network from figure 135.



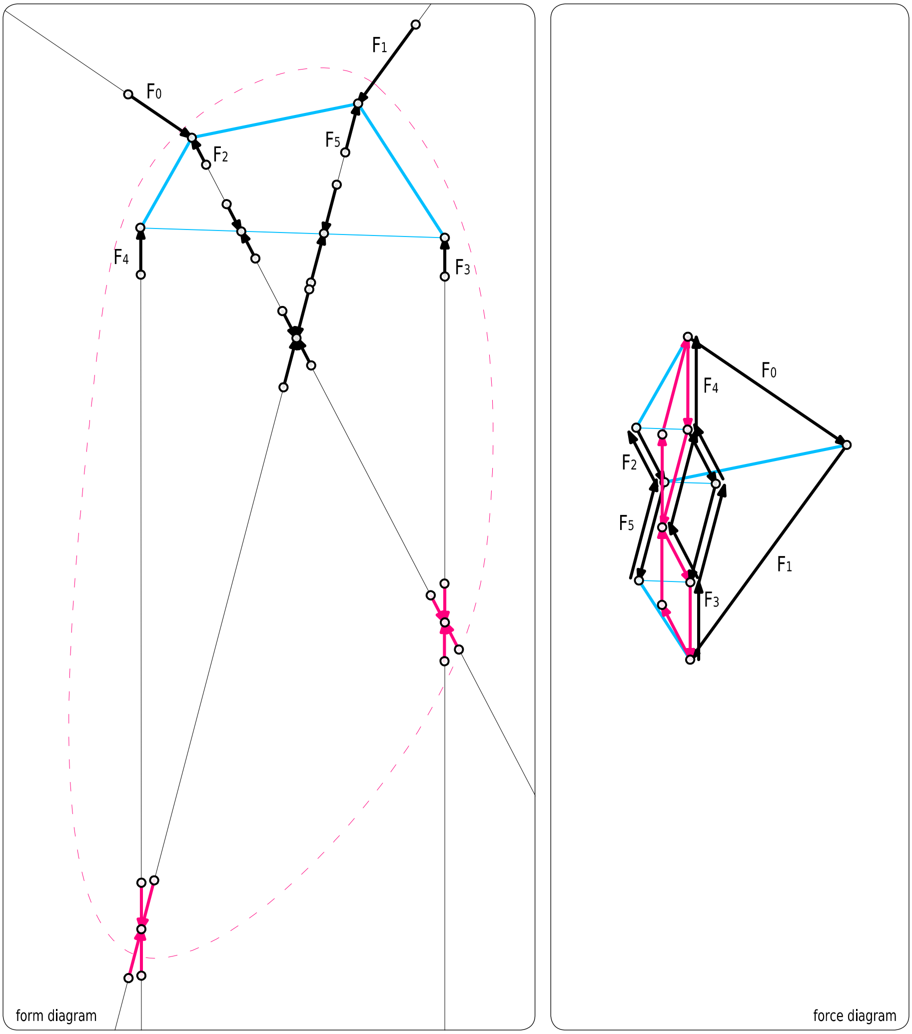


figure 138
maximal hull containing the entire strut-and-tie network from figure 135.

14 recapitulation

This sub-section briefly summarises the series of axioms that have been introduced throughout this section.

geometric support · The axiomatisation exclusively concerns fundamental relationships between positions of points and forces in the plane. The geometric support on which it is based uses Laterality and Proximity relationships as (incompletely) defined by axioms Ax.1 to Ax.11 (page 62 to page 66). These relationships natively enable the definition of inequalities of distances and relative directions.

Points may belong to the form diagram or the force diagram depending on whether they verify the FormDiagramMembership relationship or the ForceDiagramMembership relationship. Distances in these diagrams may be compared with a unit distance equal to the one that satisfies the UnitDistance relationship. These three fundamental relationships are defined by axioms Ax.12 to Ax.17 (page 76 to page 79).

forces, equilibrium and reciprocity · Forces are represented by Force relationships, defined by axioms Ax.18 to Ax.20 (page 82 to page 83). Two forces are said equivalent if they satisfy the fundamental relationship Equipollence, defined by axioms Ax.21 and Ax.22 (page 84). Unless explicitly tated by an Equipollence relationship, all the forces are different.

Any logical sentence that satisfies the five axioms Ax.23 to Ax.26 (page 100 to page 102) describes, by means of two diagrams, a strut-and-tie network that is in static equilibrium. The last axiom Ax.27 (page 109) guarantees that the form and the force diagrams are reciprocal.

non-fundamental relationships · Using the seven fundamental relationships Laterality, Proximity, UnitDistance, FormDiagramMembership, ForceDiagramMembership, Force and Equipollence, more complex non-fundamental relationships can be defined. They might be used to determine new geometric concepts, but also new structural concepts such as, for example, the rod (figure 75, page 91):

$$
\begin{aligned}
&\mathrm{Rod}[F_0\ F_1\ p_0\ p_1\ p_2\ p_3\ p_4\ p_5] :\longleftrightarrow \exists p_A p_B:\\
&\quad \mathrm{Force}[F_0\ p_0\ p_1\ p_4\ p_5] \wedge \mathrm{Force}[F_1\ p_2\ p_3\ p_5\ p_4]\\
&\quad \wedge \mathrm{Laterality}[p_4\ p_5\ p_0\ p_2] \wedge \mathrm{Laterality}[p_4\ p_5\ p_2\ p_0]\\
&\quad \wedge \mathrm{Laterality}[p_0\ p_1\ p_B\ p_A] \wedge \mathrm{Laterality}[p_2\ p_3\ p_A\ p_B]\\
&\quad \wedge \mathrm{Equidistance}[p_0\ p_A\ p_0\ p_2] \wedge \mathrm{Equidistance}[p_2\ p_A\ p_0\ p_2] \wedge \mathrm{Laterality}[p_A\ p_0\ p_0\ p_2]\\
&\quad \wedge \mathrm{Equidistance}[p_0\ p_B\ p_0\ p_2] \wedge \mathrm{Equidistance}[p_2\ p_B\ p_0\ p_2] \wedge \mathrm{Laterality}[p_B\ p_2\ p_2\ p_0]
\end{aligned}
$$

DYNAMIC HANDLING OF GEOMETRIC CONSTRAINTS

This section establishes how the graphic static rules developed in the previous section can be observed when points in their diagram move.

Sub-section 15 ("graphical regions and dynamic compliance with geometric relationships", page 137) defines three fundamental graphical constraints, relates their application onto points with the verification of geometric relationships, and introduces a mechanism that allows the displacement of constrained positions.

Sub-section 16 ("constraint (inter)dependencies", page 155) clarifies how this mechanism also handles interdependent constraints. Constraints intended for graphical computation are exemplified in sub-section 17 ("examples of graphical computations", page 165). Sub-section 18 ("switching constraint dependencies", page 195) exhibits a systematic method to switch dependencies between constrained points symbolically, *i.e.* without the need to reconsider the entire geometric construction. Sub-section 19 ("constraint propagations", page 201) develops methods that construct new constraints symbolically in order to ensure that a solution always exists. The advantages and limitations of these methods are discussed in the same sub-section.

Sub-section 21 ("constraints for a uniform reading cycle of forces", page 243) defines the constraints that are required in order for diagrams to display a uniform reading cycle of forces and sub-section 22 ("facilitating the crossing of rods", page 259) discusses automated mechanisms that allow crossing rods in the form diagram.

Sub-section 20 ("dynamic conditional geometric statements", page 233) finally describes constraints that allow the graphical execution of dynamic logic with the only aforementioned devices.

15 graphical regions and dynamic compliance with geometric relationships

The role of this section is to characterise the set of values that a certain parameter can hold in order to satisfy a given geometric statement with other given parameters. Since all these parameters are chosen to be only positions of points, the set of values can be described entirely with graphical regions in the plane.

fundamental graphical regions · It has been decided that these graphical regions will be described using three fundamental shapes: the half-plane, the inside of a disc and the outside of a disc. They are defined as follows.

(1) a half-plane, denoted HalfPlane[p_A p_B p_C], is defined as the closed region, *i.e.* with the boundary included, to the left of p_A according to the direction going from p_B to p_C (figure 139, left). The inversion of a half-plane, denoted \HalfPlane[p_A p_B p_C], is logically the open region, *i.e.* with the boundary excluded, to the right of p_A according to the direction going from p_B to p_C (figure 139, right);

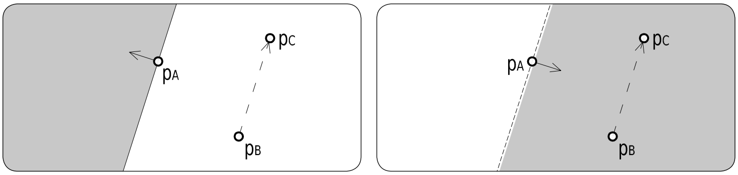


figure 139 a HalfPlane [p_A p_B p_C] region (left) and its inverse (right).

(2) the inside of a disc, denoted DiscInside[p_A p_B p_C], is defined as the closed region, *i.e.* with the boundary included, inside the circle of centre p_A and radius p_Bp_C (figure 140, left). The inversion of the inside of a disc, denoted \DiscInside[p_A p_B p_C], is logically the open region, *i.e.* with the boundary excluded, outside the circle of centre p_A and radius p_Bp_C (figure 140, right).

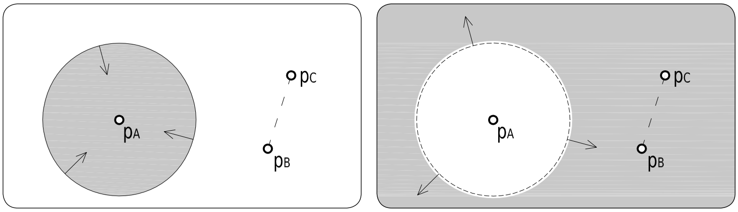


figure 140
a DiscInside [p_A p_B p_C] region (left) and its inverse (right).

(3) the outside of a disc, denoted DiscOutside[p_A p_B p_C], is defined as the closed region, *i.e.* with the boundary included, outside the circle of centre p_A and radius p_Bp_C (figure 141, left). The inversion of the outside of a disc, denoted \DiscOutside[p_A p_B p_C], is logically the open region, *i.e.* with the boundary excluded, inside the circle of centre p_A and radius p_Bp_C (figure 141, right).

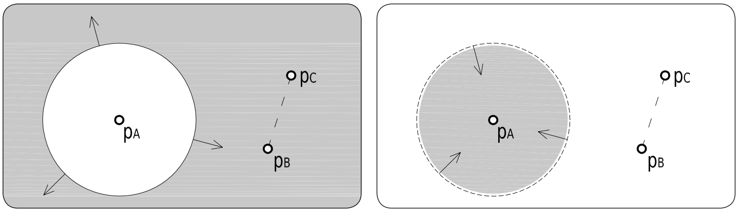


figure 141
a DiscOutside [p_A p_B p_C] region (left) and its inverse (right).

Particular cases of DiscInside and DiscOutside regions occur when points p_B and p_C are coincident. The DiscInside[p_A p_B p_B] region is the single position p_A (figure 142, left) and its inverse is the entire plane minus the position of p_A (figure 142, right). The DiscOutside[p_A p_B p_B] region is the entire plane and its inverse does not exist.

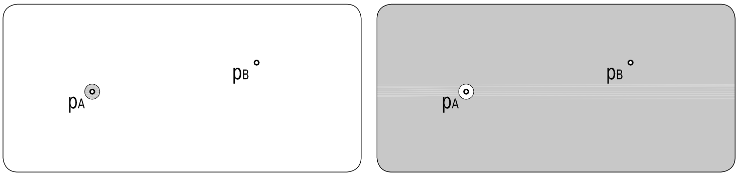


figure 142
a DiscOutside [p_A p_B p_B] region (left) and its inverse (right).

Boolean combinations of graphical regions · More complex graphical regions can be obtained by Boolean combinations, *e.g.* unions (∪), intersections (∩), inversions (\), differences (− or \∩), of these three fundamental regions. For example, figure 143 shows the following region:

(\(HalfPlane[p_0 p_0 p_1] ∪ HalfPlane[p_0 p_2 p_0]) ∪ HalfPlane[p_0 p_3 p_4])
∩ DiscOutside[p_0 p_0 p_1] ∩ DiscInside[p_0 p_0 p_2]

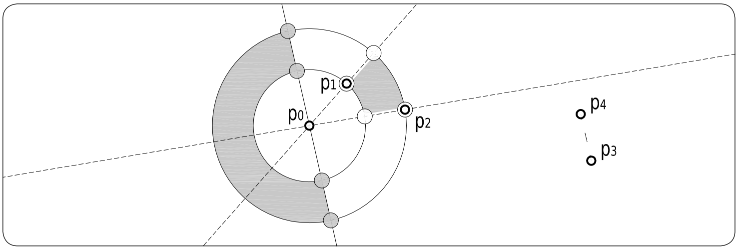


figure 143
example of Boolean combination.

Combined regions can be defined as new non-fundamental regions in order to make them easier to use. The following definitions illustrate two non-fundamental regions corresponding to a line and a circle respectively:

Straightedge[p_0 p_1 p_2] := HalfPlane[p_0 p_1 p_2] ∩ HalfPlane[p_0 p_2 p_1]
Compass[p_0 p_1 p_2] := DiscInside[p_0 p_1 p_2] ∩ DiscOutside[p_0 p_1 p_2]

The line given by two points of passage and the circle given by its centre and a point of passage can be defined has follows:

VeeringStraightedge[p_0 p_1] := HalfPlane[p_0 p_0 p_1] ∩ HalfPlane[p_0 p_1 p_0]
CollapsibleCompass[p_0 p_1] := DiscInside[p_0 p_0 p_1] ∩ DiscOutside[p_0 p_0 p_1]

The region corresponding to a single position p_0 and the entire plane on which p_0 is positioned can also be defined by new non-fundamental constraints:

Position[p_0] := DiscInside[p_0 p_0 p_0]
Ω := Plane[p_0] := DiscOutside[p_0 p_0 p_0]

pure equivalent unions and intersections · A Boolean combination can be written in many equivalent ways. Two types of equivalent writings are of particular interest: pure equivalent intersections and pure equivalent unions.

- a pure equivalent union of a given Boolean combination is an equivalent Boolean combination in which (1) intersections only group fundamental regions or inverses of fundamental regions and (2) unions do not group other unions
- a pure equivalent intersection of a given Boolean combination is an equivalent Boolean combination in which (1) unions only group fundamental regions or inverses of fundamental regions and (2) intersections do not group other intersections

The transformation of a given Boolean combination into one of its pure equivalent can easily be made by using distributivity and De Morgan's laws. If R_0, R_1 and R_2 are three graphical regions, these laws are as follows:

$R_0 \cup (R_1 \cap R_2) = (R_0 \cup R_1) \cap (R_0 \cup R_2)$
$R_0 \cap (R_1 \cup R_2) = (R_0 \cap R_1) \cup (R_0 \cap R_2)$

$\backslash(R_0 \cap R_1) = \backslash R_0 \cup \backslash R_1$
$\backslash(R_0 \cup R_1) = \backslash R_0 \cap \backslash R_1$

As a practical example, the pure equivalent Boolean combinations of the region illustrated in figure 143, page 139 is as follows:

pure equivalent union:
(\HalfPlane[p_0 p_0 p_1] ∩ \HalfPlane[p_0 p_2 p_0]
∩ DiscOutside[p_0 p_0 p_1] ∩ DiscInside[p_0 p_0 p_2])
∪ (HalfPlane[p_0 p_3 p_4] ∩ DiscOutside[p_0 p_0 p_1] ∩ DiscInside[p_0 p_0 p_2])

pure equivalent intersection:
(\HalfPlane[p_0 p_0 p_1] ∪ HalfPlane[p_0 p_3 p_4])
∩ (\HalfPlane[p_0 p_2 p_0] ∪ HalfPlane[p_0 p_3 p_4])
∩ DiscOutside[p_0 p_0 p_1]
∩ DiscInside[p_0 p_0 p_2]

constraining points in graphical regions · The three fundamental shapes HalfPlane, DiscInside and DiscOutside are actually closely linked to the two fundamental relationships Laterality and Proximity, defined previously in sub-section 08 (“relationships of proximity and laterality”, page 61). HalfPlane regions share the ability to describe relative direction with Laterality relationships and DiscInside and DiscOutside regions share the ability to describe inequalities of distances with Proximity relationships — see the paragraph entitled “beyond classical compass-and-straightedge constructions” (page 75). Moreover, the following statements hold

- if four points hold a Laterality[p_0 p_1 p_2 p_3] relationship, they each have to remain within a half-plane, whose position and orientation are defined by the three other points (figure 144)
- if four points hold a Proximity[p_0 p_1 p_2 p_3] relationship, they each have to remain either inside or outside a disc, whose position and radius are defined by the three other points (figure 145).

More precisely, a Laterality[p_0 p_1 p_2 p_3] relationship is always true when, for any positions p_0, p_1, p_2 and p_3, the following statements are verified:

- p_0 is in the HalfPlane[p_1 p_2 p_3] region (figure 144, top left)
- p_1 is in the HalfPlane[p_0 p_3 p_2] region (figure 144, top right)
- p_2 is in the HalfPlane[p_3 p_1 p_0] region (figure 144, bottom left)
- p_3 is in the HalfPlane[p_2 p_0 p_1] region (figure 144, bottom right).

figure 144
four points holding a Laterality [p_0 p_1 p_2 p_3] relationship, each of them belongs to a HalfPlane region.

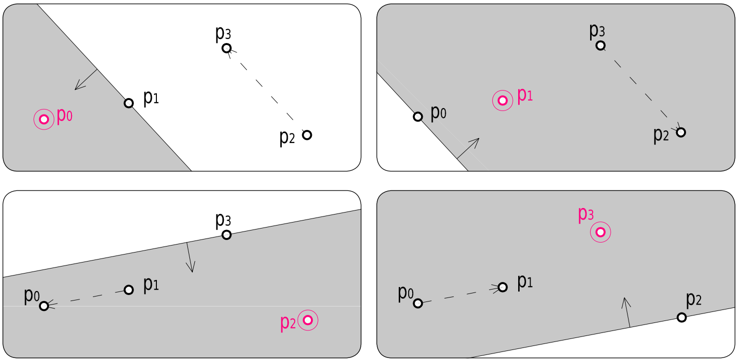
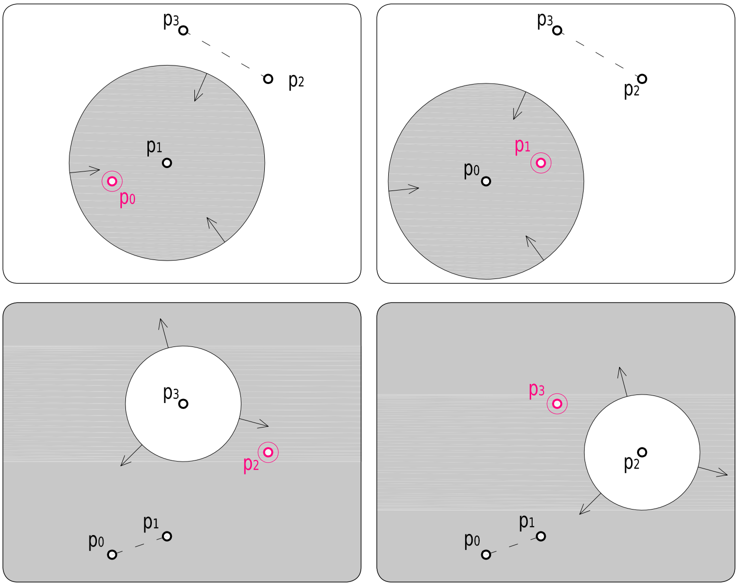


figure 145
four points holding a Proximity [p_0 p_1 p_2 p_3] relationship, each of them belongs to a DiscInside region or a DiscOutside region.



Likewise, a Proximity[p_0 p_1 p_2 p_3] relationship is always true when, for any positions p_0, p_1, p_2 and p_3, the following statements are verified:

- p_0 is in the DiscInside[p_1 p_2 p_3] region (figure 145, top left)
- p_1 is in the DiscInside[p_0 p_2 p_3] region (figure 145, top right)
- p_2 is in the DiscOutside[p_3 p_0 p_1] region (figure 145, bottom left)
- p_3 is in the DiscOutside[p_2 p_0 p_1] region (figure 145, bottom right).

Hence, compelling a point to remain on the Boolean combination of HalfPlane, DiscInside and DiscOutside regions is equivalent to ensuring that the corresponding logical combination of Laterality and Proximity is verified.

For instance, if a point p_x remains in the region of figure 143, page 139 (figure 146), the following statement is always true:

$$(\neg(\text{Laterality}[p_x\ p_0\ p_0\ p_1] \vee \text{Laterality}[p_x\ p_0\ p_2\ p_0]) \vee \text{Laterality}[p_x\ p_0\ p_3\ p_4])$$
$$\wedge\ \text{Proximity}[p_0\ p_1\ p_x\ p_0] \cap \text{Proximity}[p_0\ p_x\ p_0\ p_2]$$

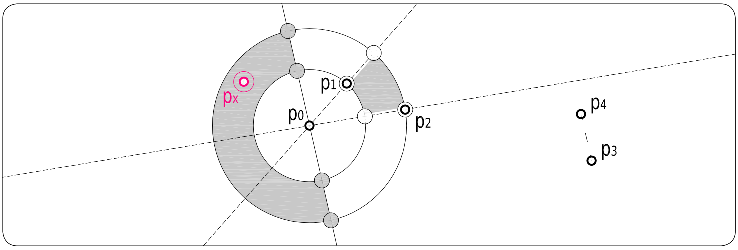


figure 146
point p_x must stay within the grey region.

It is subsequently decided that geometric statements are, from now on, described as logical conjunctions of a point's membership of a region. This is the reason why each region will be called a "*constraint*". The symbol ∈ will stand for the point's membership of constraint.

For example, the following memberships describe three point's memberships of constraints. Points p_3 and p_4 bisect the line p_0p_1 and p_5 is constrained on the orthogonal projection of p_2 onto the line p_0p_1 (figure 147):

$$p_3 \in \text{Compass}[p_0\ p_0\ p_1] \cap \text{Compass}[p_1\ p_0\ p_1] \cap \text{HalfPlane}[p_0\ p_0\ p_1]$$
$$p_4 \in \text{Compass}[p_0\ p_0\ p_1] \cap \text{Compass}[p_1\ p_0\ p_1] \cap \text{HalfPlane}[p_0\ p_1\ p_0]$$
$$p_5 \in \text{Straightedge}[p_0\ p_0\ p_1] \cap \text{Straightedge}[p_2\ p_3\ p_4]$$

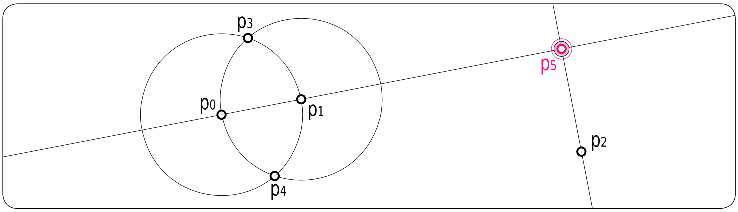


figure 147
point p_5 is constrained on the orthogonal projection of p_2 onto the line p_0p_1.

This series of memberships is equivalent to the satisfaction of the following geometric relationships:

$$(\text{Equidistance}[p_3\ p_0\ p_0\ p_1] \wedge \text{Equidistance}[p_3\ p_1\ p_0\ p_1] \wedge \text{Laterality}\ [p_3\ p_0\ p_0\ p_1])$$
$$\wedge\ (\text{Equidistance}[p_4\ p_0\ p_0\ p_1] \wedge \text{Equidistance}[p_4\ p_1\ p_0\ p_1] \wedge \text{Laterality}\ [p_4\ p_0\ p_1\ p_0])$$
$$\wedge\ (\text{Parallelism}[p_5\ p_0\ p_0\ p_1] \wedge \text{Parallelism}[p_5\ p_2\ p_3\ p_4])$$

It may be noted that the order in which constraints are applied has no influence on the geometric result.

restriction on the scope of available geometric statements · The choice of describing geometric statements by means of logical conjunctions of memberships reduces the scope of available geometric statements (compared to those studied in the previous section). Although graphical constraints can be made of unions, intersections and inverses of fundamental regions, conditions of membership must be joined by logical conjunctions only, *i.e.* no logical implication, bijection, disjunction, negation and quantifier is allowed.

In terms of logical relationships, this means that the only available geometric relationships allowed are those meeting the following requirement: given the geometric relationship written such that (1) conjunctions only join disjunctions, fundamental relationships, *i.e.* Laterality and Proximity relationships, or their inverses and (2) disjunctions only join fundamental relationships or their inverses; if a point p_0 is used in one of the fundamental relationships that is in disjunction with other fundamental relationships φ_j, then this point must also be used in all fundamental relationships φ_j.

For example, the following relationship is allowed, where φ_0 to φ_6 are fundamental relationships:

$$\varphi_0[\ldots p_0\ldots] \wedge (\varphi_1[\ldots p_0\ldots] \vee \varphi_2[\ldots p_0\ldots] \vee \varphi_3[\ldots p_0\ldots]) \wedge \varphi_4[\ldots] \wedge (\varphi_5[\ldots] \vee \varphi_6[\ldots])$$

But this one is prohibited:

$$\varphi_0[\ldots p_0\ldots] \vee \varphi_2[\ldots] \textit{ where } p_0 \textit{ is not a parameter of } \varphi_2$$

This restriction of scope has no technical or practical rationale except that it makes the current research easier. Hopefully, it has a limited impact on the purpose of common graphic statics constructions. This is mostly due to the fact that geometric statements can generally be expressed in many different ways. A brief overview of capabilities will be given in sub-section 17 ("examples of graphical computations", page 165). Moreover, sub-section 20 ("dynamic conditional geometric statements", page 233) will later show that conditional constructions are still available despite this restriction.

allowing dynamic displacements of points · Only static geometric statements have been considered so far, *i.e.* each point occupies a fixed given position that satisfies the relationships to which it is subjected. The ability to move points in their plane is now introduced and studied. It follows that each logical geometric relationship (obtained by conjunction of memberships) must be satisfied for any new change of position. This paragraph compares two options for guaranteeing the permanent verification of a relationship when points are moving and selects the latter.

As recalled in the paragraph entitled "constraining points in graphical regions" (page 140), Laterality and Proximity relationships are symmetrical — see Ax.1 (page 62) and Ax.6 (page 64). As a consequence, guaranteeing the application of a HalfPlane[p_1 p_2 p_3] constraint onto a point p_0 would ensure that the Laterality[p_0 p_1 p_2 p_3] relationship is satisfied, and would therefore be equivalent to compelling p_1 within the HalfPlane[p_0 p_3 p_2] region, p_2 within the HalfPlane[p_3 p_1 p_0] region and p_3 within the HalfPlane[p_2 p_0 p_1] region. The same is true for DiscInside and DiscOutside constraints. Combinations of fundamental constraints present similar properties as well, *i.e.* the application of a non-fundamental constraint onto a point remains true as long as all the parameter points of that constraint stay within certain graphical regions.

A first option would be to constrain each point within the corresponding graphical region, *i.e.* they cannot be dragged outside that region. This option is not adopted for practical reasons. The four following examples explain why:

(1) Given four points p_0, p_1, p_2, p_3 holding a Laterality[p_0 p_1 p_2 p_3] relationship, p_1 is asked to be moved onto a fifth point p_4. Since they hold a Laterality relationship, points p_0, p_1, p_2 and p_3 would each be constrained in a HalfPlane region. If C_0, C_1, C_2 and C_3 are those constraints, this means that $p_0 \in C_0$, $p_1 \in C_1$, $p_2 \in C_2$ and $p_3 \in C_3$. As a consequence, point p_1 would not be able to move onto p_4 (figure 148), it would be stopped on the boundary of C_1 (figure 149). In order for p_1 to reach p_4, p_0, p_2 or p_3 must first be moved further than p_4 (figure 150). This means that two movements are required for one desired move (figure 151).

The movement of p_0, p_2 or p_3 can be automated but this would require choosing what point to move first. When a point is constrained by more complex Boolean combinations, it could become necessary to move multiple points after having selected them from a larger set. This selection should not be left to the computer since it may modify input data without the user being aware of it, which might be confusing — *e.g.* controlling the position of a line is different from controlling is orientation. On the other hand, this selection should not be left to the user either since it may represent a long and laborious process.

figure 148
initial situation, point p_1 has to be moved upon point p_4.

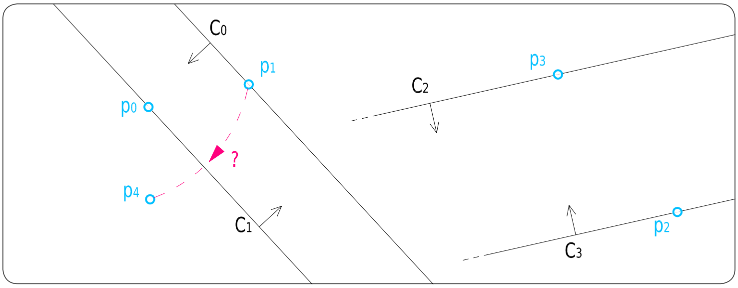


figure 149
first attempt to move of p_1.

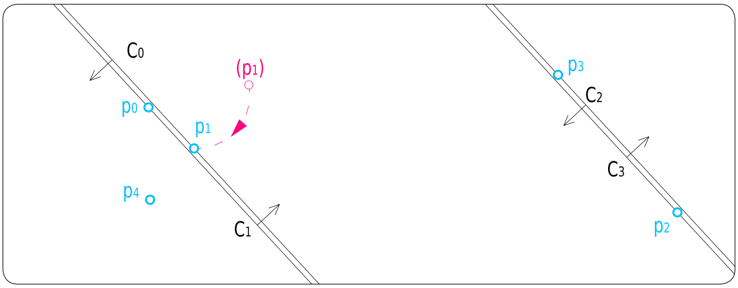


figure 150
required move of p_0.

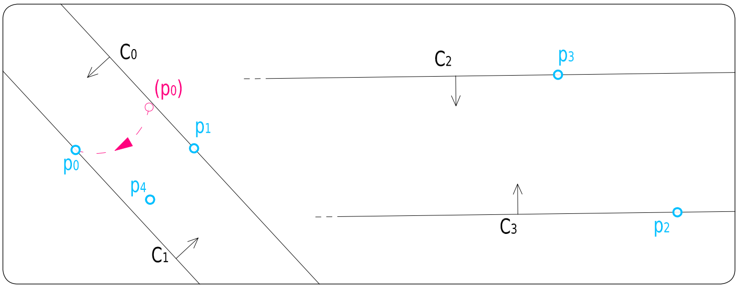
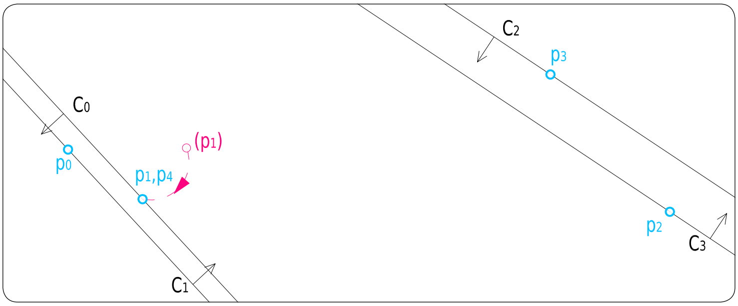


figure 151
final move of p_1.



(2) A similar drawback is obtained with four points p_0, p_1, p_2, p_3 holding a Proximity[p_0 p_1 p_2 p_3] relationship, p_1 is asked to be moved onto a fifth point p_4. Since they hold a Proximity relationship, points p_0 and p_1 would each be constrained in a DiscInside region and points p_2 and p_3 would each be constrained in a DiscOutside region. If C_0, C_1, C_2 and C_3 are those constraints, this means that $p_0 \in C_0$, $p_1 \in C_1$, $p_2 \in C_2$ and $p_3 \in C_3$. As a consequence, point p_1 would not be able to move onto p_4 (figure 152), it would be stopped on the boundary of C_1 (figure 153). In order for p_1 to reach p_4, p_0 must first be moved closer to p_4 (figure 154 and figure 155) or p_3 and p_4 must be moved far apart from each other. In the event that p_4 is far further away from p_1 (figure 156) and where the distance p_2p_3 cannot be changed, p_0 and p_1 would need to be dragged a number of times by small increments. This means that multiple movements are required for one desired move.

(3) The third example is even more worrying. Given four points p_0, p_1, p_2, p_3 holding a Parallelism[p_0 p_1 p_2 p_3] relationship, *i.e.* the conjunction of two Laterality relationships, they would each be constrained on a Straightedge, *i.e.* the intersection of two HalfPlane regions, the position and orientation of which are given by the three other points (figure 157). This means that it is impossible to move these four points anywhere except on their Straightedge constraint. In other words, the orientation of and the space between the two parallels are unchangeable.

(4) The final example is similar. Given four points p_0, p_1, p_2, p_3 holding an Equidistance[p_0 p_1 p_2 p_3] relationship, *i.e.* the conjunction of two Proximity relationships, they would each be constrained on a Compass, *i.e.* the intersection of a DiscInside region with a DiscOutside region, the position and radius of which are given by the three other points (figure 158). This means that it is impossible to move these four points anywhere except on their Compass constraint. In other words, it is impossible to modify the distances p_0p_1 and p_2p_3.

For all these reasons, the second option is adopted. It constrains just one of the concerned points in its graphical region. Since this allows the other points to be moved outside their own graphical region, they might take the constrained point outside its own graphical region as well. To avoid this, the constrained point is automatically moved to the closest suitable position as soon as it is taken outside its region.

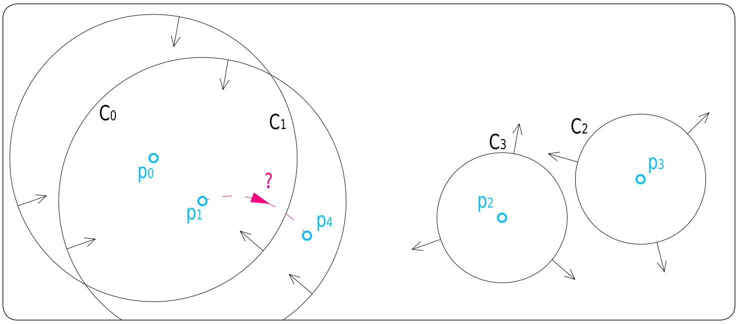


figure 152
initial situation, point p_1 has to be moved upon point p_4.

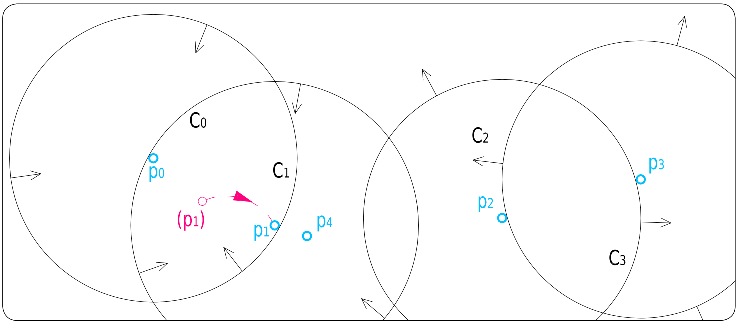


figure 153
first attempt to move p_1.

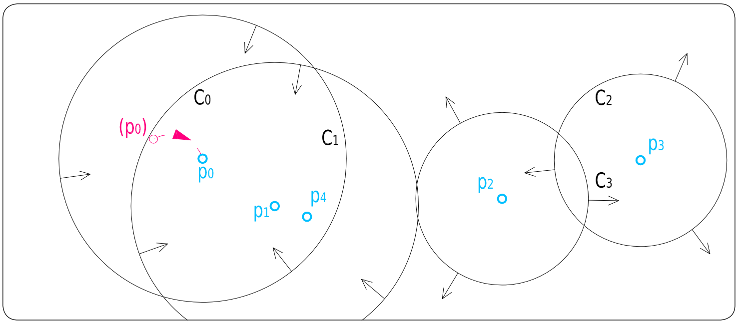


figure 154
required move of p_0.

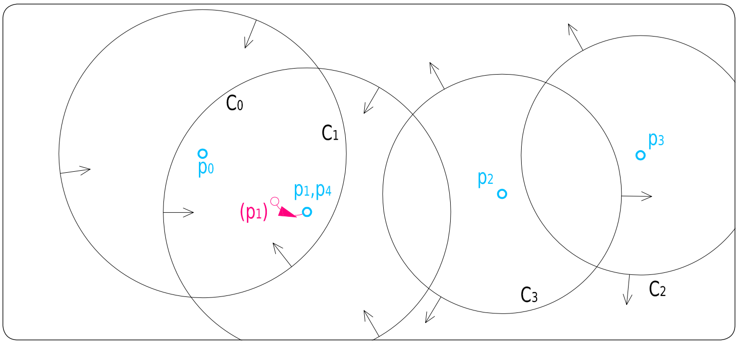


figure 155
final move of p_1.

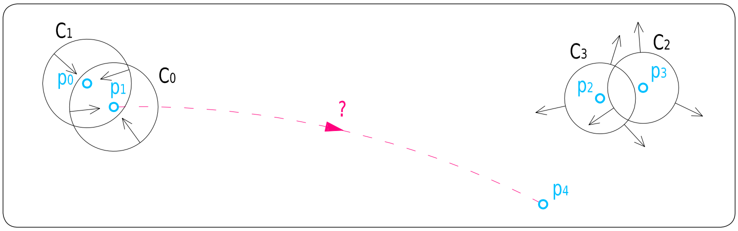


figure 156
same situation as in figure 152 except that the distance $p_1 p_4$ is greater.

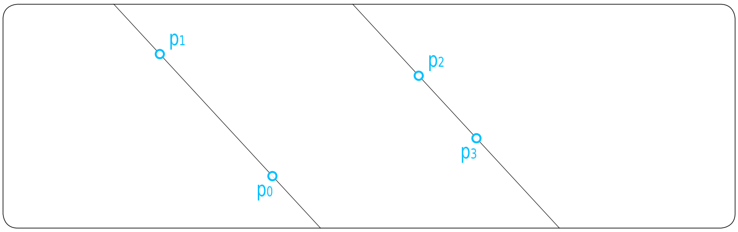


figure 157
four points constrained on parallels.

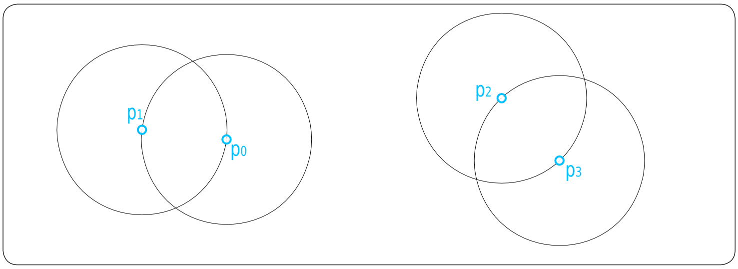


figure 158
four points constrained on similar circles.

For example, given four points and the membership $p_0 \in$ HalfPlane[p_1 p_2 p_3] (figure 159), moving p_2 clockwise around p_3 would rotate the half-plane around p_1 and would change no other position until $p_2 p_3$ becomes parallel to $p_0 p_1$ (figure 160). However, if p_2 keeps moving further, it would cause the position of p_0 to be updated on its closest position within the HalfPlane[p_1 p_2 p_3] constraint (figure 161).

As a result of this new rule, the nature of the point on which the constraint is applied differs from the other points. This point must be chosen and known. That is the reason why the logical grammar using geometric relationships for describing the geometric state in full is no longer used and the description of point memberships in graphical regions is used instead.

The closest position of a point is actually either its orthogonal projection onto the given border or an inflexion point of that boundary. Thanks to the nature of the three fundamental regions, this new position is fairly quick to compute because it is either an orthogonal projection on a line or a circle, or the in-

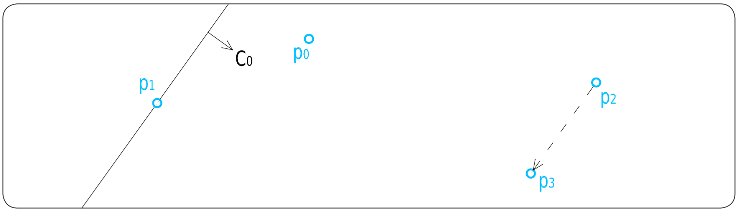


figure 159
a point p_0 constrained on a half-plane.

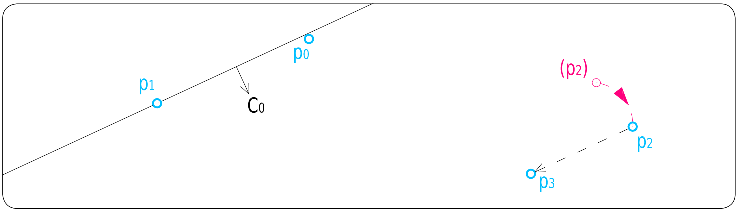


figure 160
move of p_2, it rotates the half-plane.

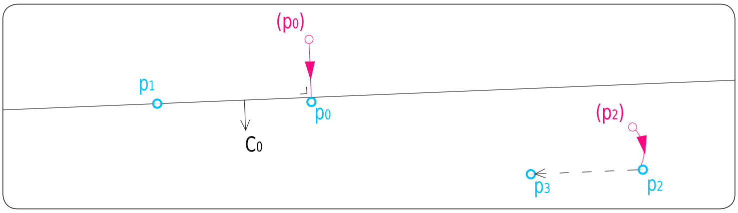


figure 161
move of p_2, it forces the update of p_0 on its closest position within the half-plane.

tersection between two lines, two circles or a line with a circle. One simple method to find the closest point is to find all the candidates, *i.e.* orthogonal projections on fundamental regions and intersections between fundamental regions, and to compare their distance from the point that has to be moved (figure 162).

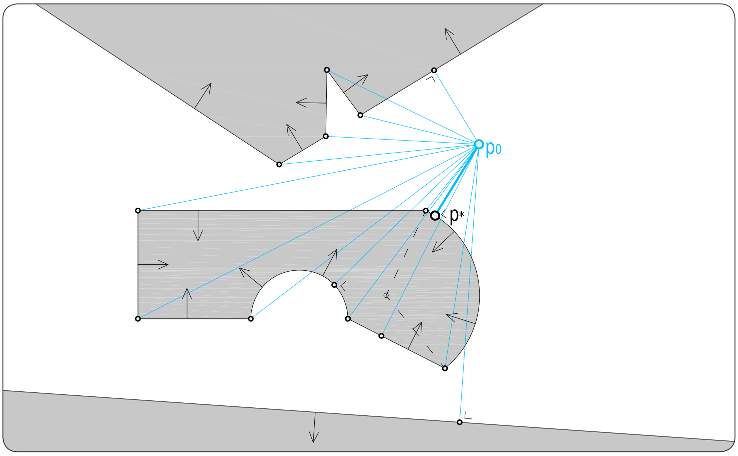


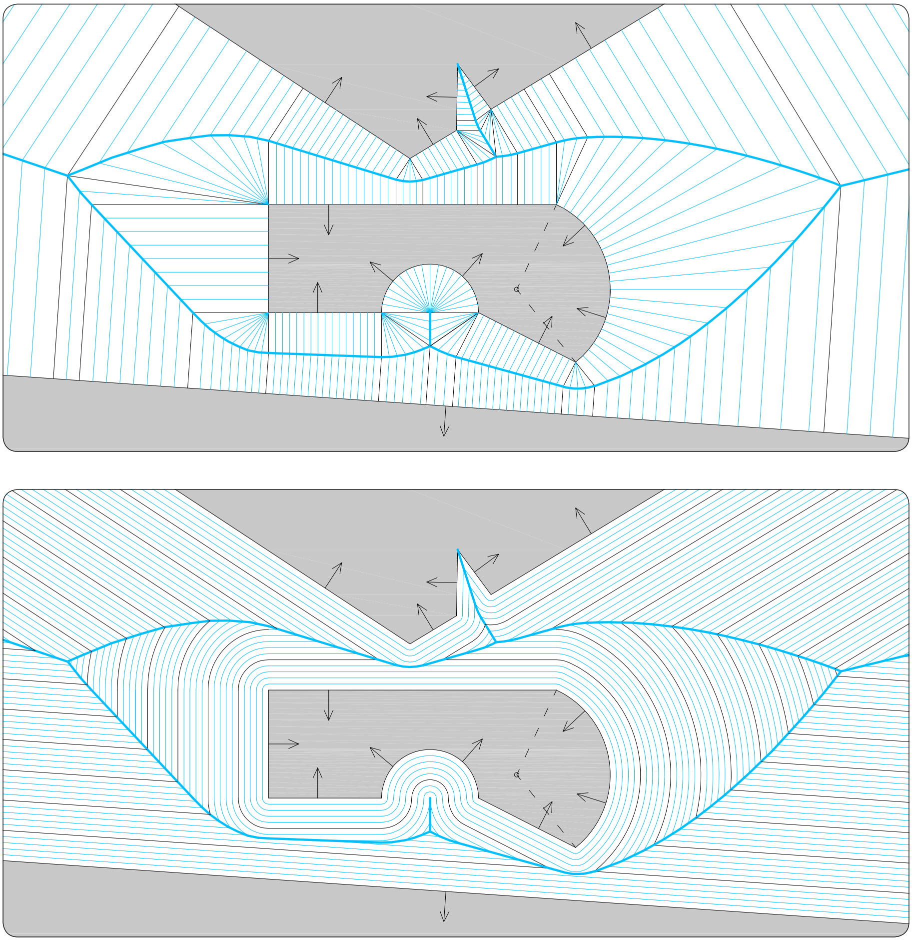
figure 162
search for the closest candidate points; point p_0 is not within its applied region and must therefore be moved to its closest point p_*.

If the region is not empty, every point always has a closest point belonging to that region (figure 163). For some positions, a point might have multiple closest positions inside its region. The set of all these positions (figure 161) is known has the *topological skeleton* and shares similarities with *Voronoi skeletons* and *medial axis* (Blum·1967, Ogniewicz/Ilg·1992 and Siddiqi/…·2008).

If the candidate closest position is on a border that is not included in the region, *i.e.* because of an inverse constraint, the closest position is assumed to be infinitesimally just beyond that border.

When a constraint consists of multiple non-continuous convex regions, it can be noticed that in most cases this action minimises the disturbance of the model when a point jumps from one region to another.

figure 163
if the region (grey area) is not empty, every position has a closest point.



domains of solutions · Since a point might belong to many graphical constraints, it is useful to describe the global intersection of these constraints. This global intersection is called the *domain*. By default, the domain of a point is full, *i.e.* the point is not constrained and can be moved anywhere in the plane.

Constraints applied to a point may have different purposes and roles. As a result, it is possible to distinguish between different kinds of domains:

(1) The "*input domain*" of a point is the intersection of all the constraints that are specifically applied by the user to the point. This is the initial set of positions outside of which the point can not go.

(2) The "*strict domain*" of a point is the intersection of the initial domain with all the constraints obtained by the symmetry of constraints applied on other points. In other words, the strict domain is the set of positions in which a point can move without changing the position of any other point, *i.e.* no update to closest positions would be required but available geometric modifications would be seriously limited, as discussed in the first option of the previous paragraph entitled "allowing dynamic displacements of points" (page 143)

(3) The "*propagation domain*" of a point is the set of positions that the point can have so that every point related to it is guaranteed to have at least one position in the plane. This domain is generally more restrictive than the input domain and requires the computation of specific solver algorithms. Sub-section 19 ("constraint propagations", page 201) discusses some of them.

(4) The "*domain of solutions*" of a point is the set of positions that solves the initial geometric statement. It is the domain that satisfies the constraints applied by the user and for which every point as a non-empty domain. It is consequently equivalent to the intersection of the input domain with the propagation domain.

figure 164 sometimes, it might be multiple closest points; the bold line groups all the positions that have multiple solutions. The bold line is known as the topological skeleton of the inverse of the region.

These four domains can be illustrated using the following construction (figure 165):

p_0 ∈ VeeringHalfplane[p_1 p_2] ∩ CollapsibleDiscOutside[p_1 p_2]
p_6 ∈ VeeringHalfplane[p_3 p_0] ∩ VeeringHalfplane[p_4 p_5] ∩ VeeringHalfplane[p_1 p_2]

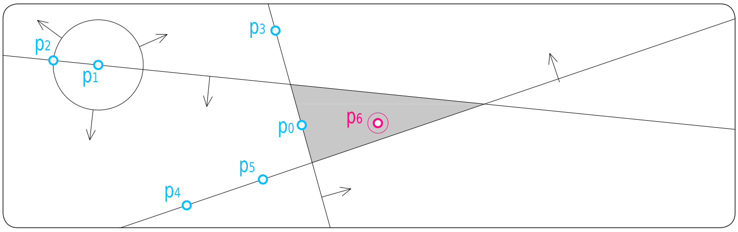


figure 165
the input domain of p_6 is the shaded area;.

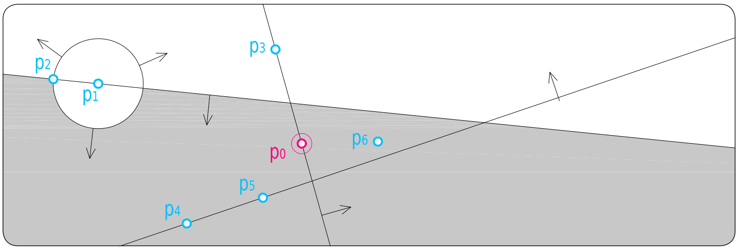


figure 166
the input domain of p_0.

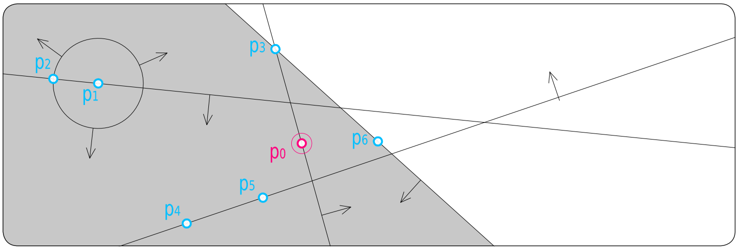


figure 167
the strict domain of p_0.

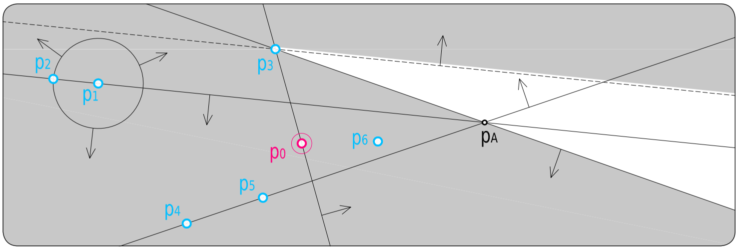


figure 168
the propagation domain of p_0.

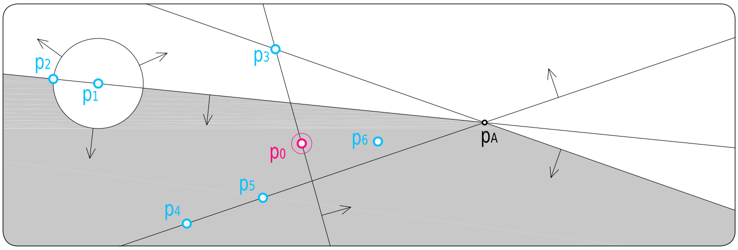


figure 169
the domain of solutions of p_0.

The various domains of p_0 are (figure 166, figure 167, figure 168 and figure 169):

inputDom[p_0] = VeeringHalfplane[p_1 p_2] ∩ CollapsibleCompass[p_1 p_2]

strictDom[p_0] = VeeringHalfplane[p_6 p_3] — p_6 *does not move*

propagationDom[p_0] = VeeringHalfplane[p_A p_3] ∪ \HalfPlane[p_3 p_1 p_2]
where p_A ∈ Straightedge[p_1 p_2] ∩ Straightedge[p_4 p_5]
— p_6 *has a non-empty domain*

solutionDom[p_0] = VeeringHalfplane[p_1 p_2] ∩ CollapsibleCompass[p_1 p_2]
∩ (VeeringHalfplane[p_A p_3] ∪ \HalfPlane[p_3 p_1 p_2])

constraining forces · The Force[F_0 p_0 p_1 p_2 p_3] relationship is no exception to the new rule of point displacement. To verify a Force[F_0 p_0 p_1 p_2 p_3] relationship means that one of the four points must be constrained by the others in order to satisfy Ax.19 (page 83). The second point, *i.e.* p_1, the point that defines whether the force exerts a pull or a push, is chosen to be constrained rather than the others because this favours a direct control of the force diagram and choosing p_0 instead does not make any sense. Sub-section 18 ("switching constraint dependencies", page 195) will show that this choice may actually be changed at any time.

As a consequence of this choice, p_1 must belong to the following domain in order to verify Ax.19 when the relationship Force[F_0 p_0 p_1 p_2 p_3] exists (page 83):

forceDom[p_1] ∈ Straightedge[p_0 p_2 p_3] ∩ CoincidenceCondition[p_0 p_2 p_3]

The constraint CoincidenceCondition[p_2 p_3 p_0] returns the position of p_0 if p_2 and p_3 are coincident and returns the inverse of the position of p_0 — *i.e.* the entire plane minus the position of p_0 — if not. The definition of this non-fundamental constraint will be held in sub-section 20 ("dynamic conditional geometric statements", page 233).

The satisfaction of Ax.27 (page 83) will also be guaranteed by a specific domain called ReadingCycleDom. Its construction will be developed in sub-section 21 ("constraints for a uniform reading cycle of forces", page 243).

other fundamental constraints · Corresponding to each fundamental relationship introduced in the previous section is a new fundamental constraint. Thereby, the points can be constrained as follows:

$p_0 \in$ ForceDiagram	*meaning that* ForceDiagramMembership[p_0] ⟷ true
$p_0 \in$ FormDiagram	*meaning that* FormDiagramMembership[p_0] ⟷ true
$p_0 \in$ UnitCompass[p_1]	*meaning that* Equidistance[p_0 p_1] ⟷ true
$p_0 \in$ PiCompass[p_1]	*meaning that* PiDistance[p_0 p_1] ⟷ true

And similarly for each transcendental number — see paragraph entitled "the fundamental relationship of unit distance" (page 76).

16 constraint (inter) dependencies

directed graphs of dependencies · When a constraint is applied to a point, the point is dependent on the points that define the constraint. These points can in turn be dependent on other points, and so on. Being able to know this genealogy is of great importance since it allows the identification of a chronology for updating positions when a point is dragged.

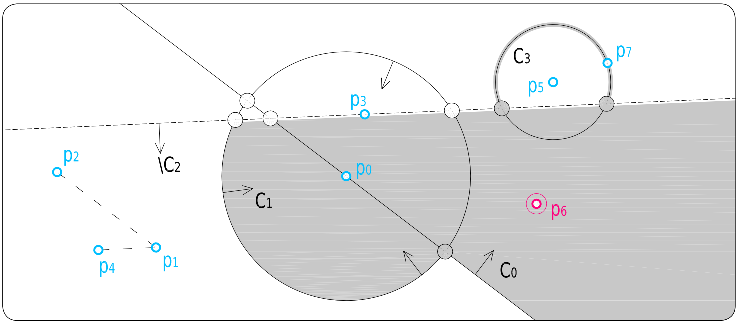
The analysis of constraint dependencies is usually performed using a directed graph. Nodes correspond to points and each arrow going from a point p_A to a point p_B represents a constraint dependent on p_A and applied to p_B.

For example, figure 171 shows the graph of dependencies of the following memberships (figure 170):

$p_3 \in C_1$
$p_5 \in C_2$
$p_6 \in C_4$
$p_7 \in C_3 \cap C_2$

where :
C_0 = HalfPlane[p_0 p_2 p_1]
C_1 = DiscInside[p_0 p_1 p_2]
C_2 = HalfPlane[p_3 p_4 p_1]
C_3 = DiscInside[p_5 p_1 p_4] ∩ DiscOutside[p_5 p_1 p_4]
$C_4 = ((C_0 \cup C_1) - C_2) \cup C_3$

figure 170
the input domain of p_6.



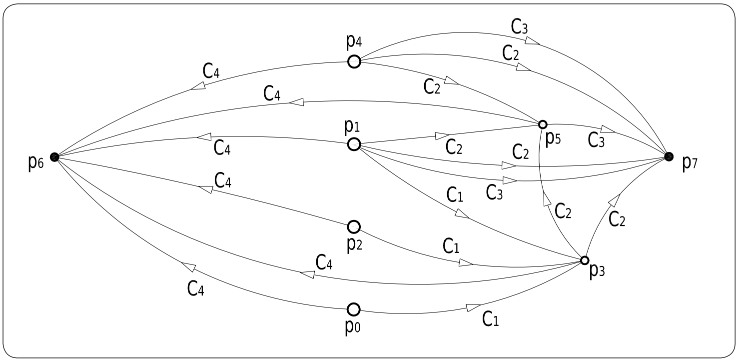


figure 171
the directed graph of dependencies associated with the construction of figure 170.

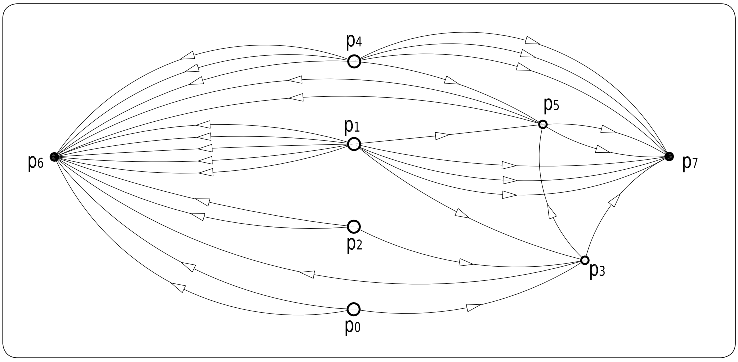


figure 172
the directed graph of fundamental dependencies associated with the construction of figure 170.

The graph in figure 171 shows only the combined constraints. The graph in figure 172 details the dependencies for all the fundamental constraints.

This directed graph allows the identification of particular points:

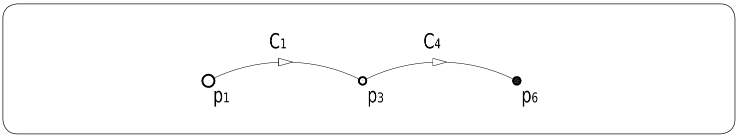
- "*father points*" of a given point p_A are all the points that define the constraints directly applied on p_A
- "*grandfather points*" of a given point p_A include all the father points of p_A as well as all the father points of its grandfather points, *i.e.* all its ancestors.
- "*child points*" of a given point p_A are all the points that directly depend on p_A
- "*grandchild points*" of a given point p_A include all the child points of p_A as well as all the child points of its grandchild points, *i.e.* all its progeny.

In the example in figure 172, the following sets can be distinguished:

children[p_5] = {p_6 p_7}
grandchildren[p_5] = {p_6 p_7}
fathers[p_5] = {p_1 p_3 p_4}
grandfathers[p_5] = {p_0 p_1 p_2 p_3 p_4}

Other particular points are orphans and childless points. The former have no father point, *i.e.* they depend on no other point — and the latter have no child point. For instance, figure 173 is a detail of figure 170 and shows that p_1 is an orphan and p_6 a childless point.

figure 173
detail of
figure 170.



movement updating · As previously mentioned, the graph of fundamental dependencies provides a straightforward method for updating the positions of points when one of them — p_A — is moved. Firstly, the subset of grandchild points of p_A provides all the points whose position has to be checked — other points do not need any update since they are not dependent on p_A. Secondly, these grandchildren can be checked from father to child paying attention so that, for each generation, a child point is only checked if all its grandfathers are either already checked or are grandfathers of p_A as well. Thirdly, checking a point will only involve an update of position if it no longer satisfies the constraints that are applied on it.

For example, if point p_3 is moved, it means that points p_0, p_1, p_2 and p_4 do not have to be checked since they are not the grandchildren of p_3 (figure 172). Out all p_3's grandchildren {p_5 p_6 p_7}, it is deduced that p_6 and p_7 can not be checked before p_5 since p_5 is the father of p_6 and p_7. As a consequence, after having updated the position of p_3, the position of p_5 must be checked. If the position of p_5 is moved, both positions of p_6 and p_7 must be checked — whether p_6 is checked before or after p_7 is a matter of choice as they do not depend on one other.

interdependency · Interdependency occurs when the directed graph presents a cycle, meaning that each point in that cycle is constrained by itself, directly or through in-between parameters. For example, the previous geometric construction becomes interdependent when the following constraint is applied on it (figure 174 and figure 175):

C_5 = DiscInside[p_5 p_1 p_4]
$p_0 \in C_5$

This cycle concerns the three points p_0, p_3 and p_5 all linked together by the following constraints:

$p_3 \in$ DiscInside[p_0 p_1 p_2]
$p_5 \in$ HalfPlane[p_3 p_1 p_4]
$p_0 \in$ DiscInside[p_5 p_1 p_4]

When one of them is moved, the update of its children might cause its own update and hence might result in a loop. After a sufficiently high number of iterations, it is envisaged that this loop will stop in one of three ways:

(1) the loop stops by itself because one of the self-constrained points is within its applied region, meaning that its position does not have to be updated anymore and consequently, neither do the positions of its children

(2) the loop produces smaller and smaller displacements — each position of self-constrained points converges to a single position without ever reaching it exactly — meaning that an additional mechanism has to stop the loop as soon as a displacement caused by the "ClosestPoint" update becomes smaller than a given value

(3) the loop does not converge, meaning that the geometric construction which caused it should be prevented.

These are the only mechanisms that have to be implemented in order to deal with interdependency. The following examples illustrate these three cases.

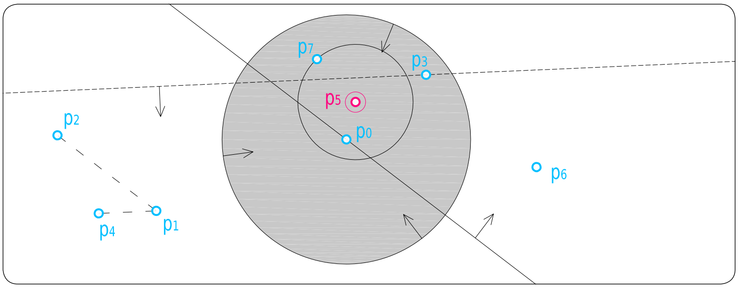


figure 174
the input domain of p_5 after the application of C_5.

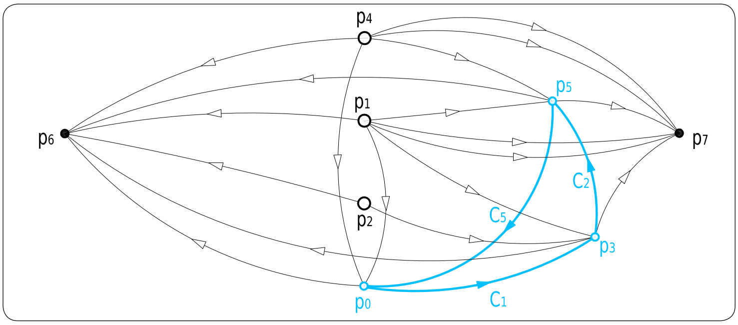


figure 175
the directed graph of fundamental dependencies associated with the construction of figure 174.

(1) Given the following construction, p_2 is moved onto a position p_2' (figure 176):

$p_2 \in \text{HalfPlane}[p_0\ p_1\ p_0]$
$p_1 \in \text{HalfPlane}[p_0\ p_2\ p_0]$

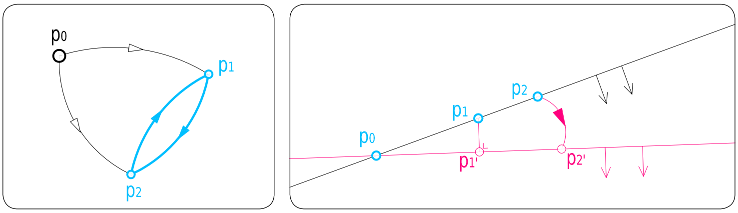


figure 176
graph of dependencies and example of movement of interdependent points, whose positions converge directly.

Although p_1 and p_2 are two interdependent points, moving p_2 will update the position of p_1 onto p_1' (its closest position within the region C_1) and this will not cause a loop since p_2' already belongs to the constraint applied on it. The same behaviour for two other new positions of p_2 is shown in figure 177.

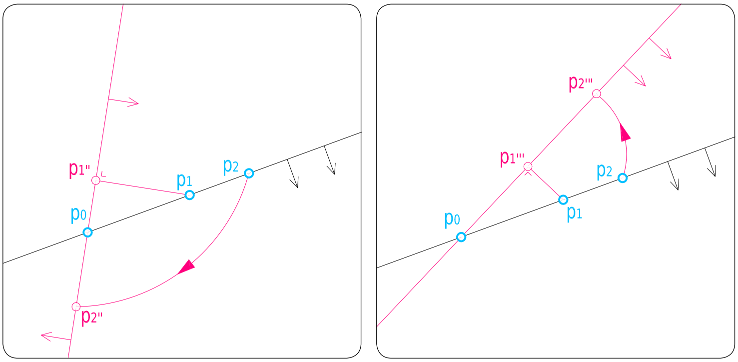


figure 177
two other displacements.

The following example shows the same behaviour for a construction involving four interdependent points, constrained on four perpendicular lines (figure 178):

$p_4 \in \text{HalfPlane}[p_3\ p_0\ p_1] \cap \text{HalfPlane}[p_3\ p_1\ p_0]$
$p_5 \in \text{HalfPlane}[p_4\ p_0\ p_2] \cap \text{HalfPlane}[p_4\ p_2\ p_0]$
$p_6 \in \text{HalfPlane}[p_5\ p_0\ p_1] \cap \text{HalfPlane}[p_5\ p_1\ p_0]$
$p_7 \in \text{HalfPlane}[p_6\ p_0\ p_1] \cap \text{HalfPlane}[p_6\ p_1\ p_0]$

Moving p_3 onto the position p_3' will update the position of p_4 onto its closest position p_4', but no update of p_5 will be required. Once again, the loop will stop by itself.

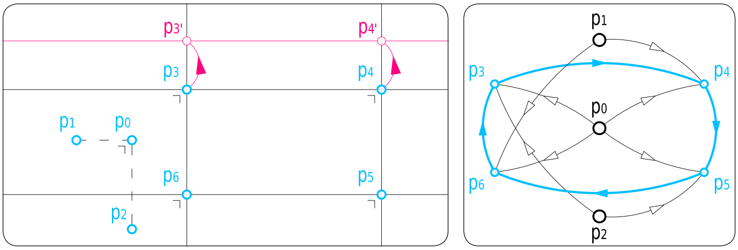


figure 178
example of movement and graph of dependencies of interdependent points, whose positions converge directly.

(2) The second kind of stop occurs when one of the three following interdependent points p_3, p_4 or p_5 is moved (figure 179):

$p_4 \in \text{HalfPlane}[p_3\ p_0\ p_1] \cap \text{HalfPlane}[p_3\ p_1\ p_0]$
$p_5 \in \text{HalfPlane}[p_4\ p_0\ p_2] \cap \text{HalfPlane}[p_4\ p_2\ p_0]$
$p_3 \in \text{HalfPlane}[p_5\ p_1\ p_2] \cap \text{HalfPlane}[p_5\ p_2\ p_1]$

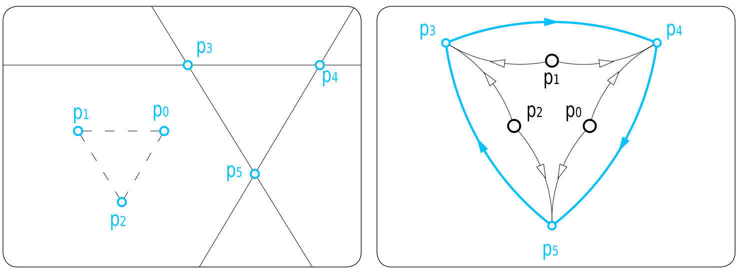


figure 179
initial situation and graph of dependencies of interdependent points.

The successive updates of these three points subsequent to the move of p_3 is shown in figure 180: moving p_3 onto p_3' implies that p_4 is updated onto p_4', implying that p_5 is updated onto p_5', p_3' onto p_3'', p_4' onto p_4'', *etc*. This will continue and may be stopped when the distance between a point and its previously updated position is smaller than a predetermined value.

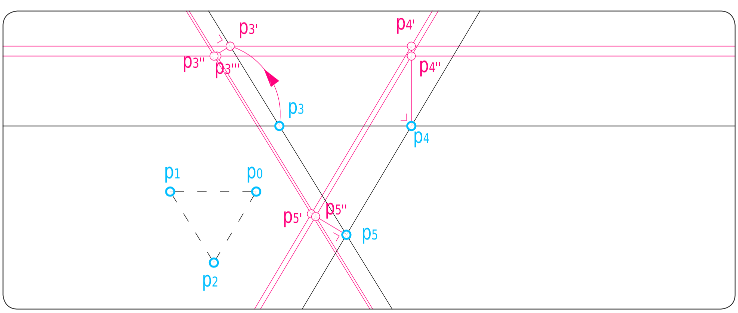


figure 180
example of movements of interdependent points, whose positions converge.

Practical uses of convergent interdependencies are the definition of a constraint with the shape of any algebraic curve. Some of them are detailed in sub-section 17 ("examples of graphical computations", page 165).

(3) The non-convergent case can be exemplified using the following applications of constraints (figure 181):

$p_2 \in$ \DiscInside[p_0 p_0 p_1]
$p_1 \in$ \DiscInside[p_0 p_0 p_2]

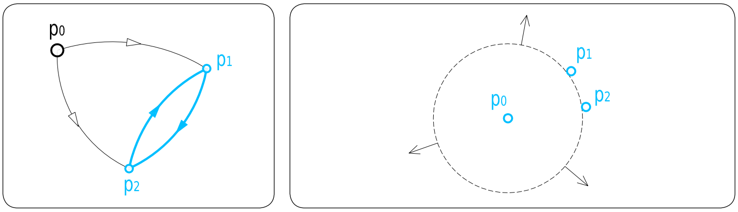


figure 181
graph of dependencies and example of movement of interdependent points, whose positions does not converge.

Points p_1 and p_2 will never reach a stable position; they will take each other away from p_0 ad infinitum. This construction must therefore be avoided. However, with additional proper constraints, these ongoing displacements might be controlled and used for specific purposes involving dynamic loops.

static anchorage due to interdependency · Some interdependencies of constraints have the effect of anchoring points on a domain that cannot be changed by any movement of other points. Constructions of this kind should consequently be avoided. Following are three examples.

(1) This construction constrains two points on a line whose orientation can no longer be altered (figure 182):

$p_3 \in$ HalfPlane[p_2 p_0 p_1] ∩ HalfPlane[p_2 p_1 p_0]
$p_2 \in$ HalfPlane[p_3 p_0 p_1] ∩ HalfPlane[p_3 p_1 p_0]

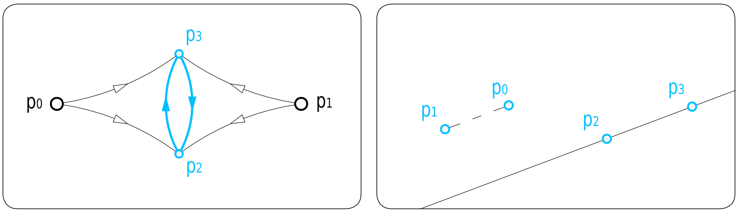


figure 182
graph of dependencies and situation of the first example of static anchorage.

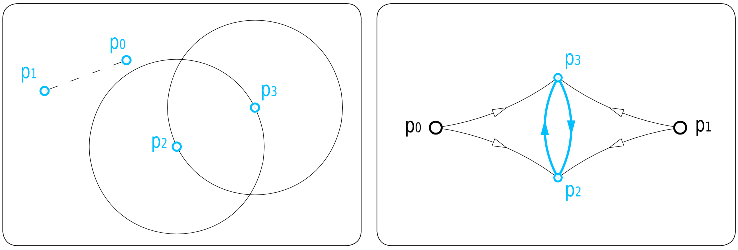


figure 183
situation and graph of dependencies of the second example of static anchorage.

(2) This construction constrains two points on two circles whose radius can no longer be altered (figure 183):

$p_3 \in \mathrm{DiscInside}[p_2\ p_0\ p_1] \cap \mathrm{DiscOutside}[p_2\ p_0\ p_1]$
$p_2 \in \mathrm{DiscInside}[p_3\ p_0\ _{p1}] \cap \mathrm{DiscOutside}[p_3\ p_0\ p_1]$

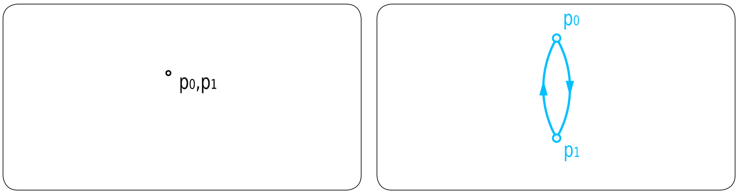


figure 184
situation and graph of dependencies of the third example of static anchorage.

(3) This construction constrains two points on positions that can no longer be altered (figure 184):

$p_0 \in \mathrm{DiscInside}[p_1\ p_1\ p_1]$
$p_1 \in \mathrm{DiscInside}[p_0\ p_0\ p_0]$

inner self-constraining · Interdependencies might also occur when a point is constrained by a fundamental constraint that is directly dependent of it. Interdependent fundamental constraints might have different behaviours:

- they may be useless — *i.e.* they constrain nothing at all, they are always satisfied naturally —, this is the case for $p_0 \in \mathrm{HalfPlane}[p_1\ p_0\ p_0]$, $p_0 \in \mathrm{DiscInside}[p_0\ p_1\ p_2]$, $p_0 \in \mathrm{DiscOutside}[p_1\ p_0\ p_0]$ and $p_0 \in \mathrm{DiscOutside}[p_1\ p_0\ p_0]$ (figure 185)
- they may be never satisfied, *e.g.* $p_0 \in \mathrm{DiscOutside}[p_0\ p_1\ p_2]$
- they may produce static anchorage, *e.g.* $p_0 \in \mathrm{DiscInside}[p_0\ p_0\ p_0]$
- they may produce particular conditions, for instance: $p_0 \in \mathrm{HalfPlane}[p_1\ p_0\ p_2]$ is strictly equivalent to $p_0 \in \mathrm{HalfPlane}[p_1\ p_1\ p_2]$ (figure 186)

figure 185
graph of dependencies and situation of $p_0 \in$ HalfPlane[p_0 p_1 p_2].

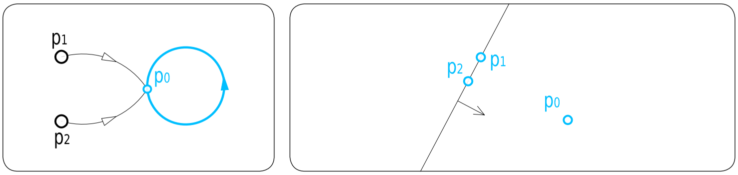


figure 186
graph of dependencies and situation of $p_0 \in$ HalfPlane[p_1 p_0 p_2].

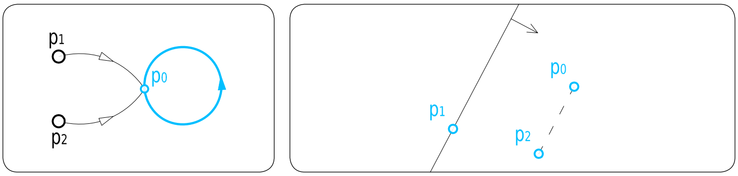


figure 187
graph of dependencies and situation of $p_0 \in$ DiscInside[p_1 p_0 p_2].

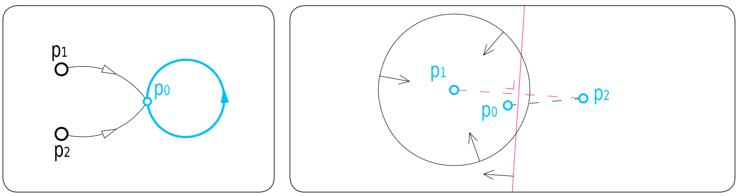
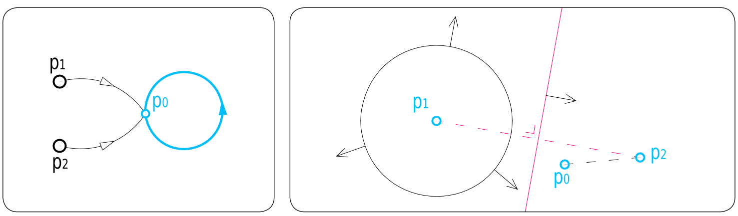


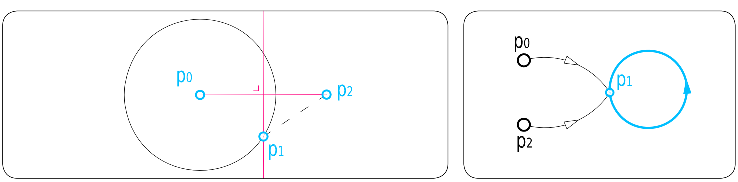
figure 188
graph of dependencies and situation of $p_0 \in$ DiscOutside [p_1 p_0 p_2].



$p_0 \in$ DiscInside[p_1 p_0 p_2] compels p_0 to stay on the side of p_1 given by the line segment bisector of p_1p_2 (figure 187)

$p_0 \in$ DiscOutside[p_1 p_0 p_2] compels p_0 to stay on the side of p_2 given by the line segment bisector of p_1p_2 (figure 188)

figure 189
situation and graph of dependencies of $p_0 \in$ DiscInside [p_0 p_1 p_2] ∩ DiscOutside [p_0 p_1 p_2].



$p_0 \in$ DiscInside[p_1 p_0 p_0] compels p_0 to stay upon p_1 and is therefore equivalent to $p_0 \in$ DiscInside[p_1 p_1 p_1].

Combinations of constraints of this kind may also produce specific results. For instance, $p_0 \in$ DiscInside[p_0 p_1 p_2] ∩ DiscOutside[p_0 p_1 p_2] constrains p_1 on the bisector of the segment p_0p_2 (figure 189).

locus of positions · The previous example highlights one difficulty that may arise with interdependencies: the displayed domain of a point may be different from the entire set of positions it can hold. This difference only concerns what is displayed and not what is allowed to move.

For example, figure 189 shows that p_1 is constrained on a circle although it can move all along the bisector. This does not mean that p_1 cannot be dragged all along the bisector: dragging p_1 outside is current position would first fix the radius of the circle and than update the position of p_1 onto its closest position belonging to the circle, which in turn would alter the radius of circle again and would update p_1 onto its newest closed position, resulting in a loop that will make p_1 converging to the bisector. Other examples using this device are presented in the following sub-section.

Since it might be preferable for the user to visualise the entire available locus of positions properly, a "*locus domain*" may be introduced. The locus domain of a point p_A given by a point p_B, where p_B is a father point of p_A, is the set of all the positions that p_A can hold when p_B travels around its own domain of solution. The locus domain of an interdependent point is therefore defined by p_A=p_B.

Thanks to the property explained in the paragraph entitled "similarity between the locus and the propagation domain" (page 230), some locus domains may be produced using accurate, symbolic algorithms similar to those developed in sub-section 19 ("constraint propagations", page 201). However, other locus domains, *e.g.* those produced by interdependency, such as the curves put forward in the next sub-section, may only be approached by numerical analytical or trial-and-error techniques performed on a discretised sample region.

17 examples of graphical computations

This sub-section exhibits a series of geometric constraints capable of performing advanced computations. Rather than listing them all, the purpose is to illustrate the means by which they are achieved and their practical interests.

The first five paragraphs show how classical compass-and-straightedge operations can be automated with non-fundamental graphical constraints. The next paragraph explains how the classical limitations of compass-and-straightedge constructions can be overcome using interdependencies of constraints. The final two paragraphs point out two approaches to constrain points on curves other than on the circle.

To make it more concise, the following abbreviations of constraints are used:

HP[p_0 p_1 p_2] := HalfPlane[p_0 p_1 p_2]
VHP[p_0 p_1] := VeeringHalfplane[p_0 p_1] = HalfPlane[p_0 p_0 p_1]

SE[p_0 p_1 p_2] := Straightedge[p_0 p_1 p_2] = HalfPlane[p_0 p_1 p_2] ∩ HalfPlane[p_0 p_2 p_1]
VSE[p_0 p_1] := VeeringStraightedge[p_0 p_1] = Straightedge[p_0 p_0 p_1]

DI[p_0 p_1 p_2] := DiscInside[p_0 p_1 p_2]
CDI[p_0 p_1] := CollapsibleDiscInside[p_0 p_1] = DiscInside[p_0 p_0 p_1]

DO[p_0 p_1 p_2] := DiscOutside[p_0 p_1 p_2]
CDO[p_0 p_1] := CollapsibleDiscOutside[p_0 p_1] = DiscOutside[p_0 p_0 p_1]

C[p_0 p_1 p_2] := Compass[p_0 p_1 p_2] = DiscInside[p_0 p_1 p_2] ∩ DiscOutside[p_0 p_1 p_2]
CC[p_0 p_1] := CollapsibleCompass[p_0 p_1] = Compass[p_0 p_0 p_1]

[p_0] := Position[p_0] = DiscInside[p_0 p_0 p_0]

dynamic compass-and-straightedge constructions · Successive applications of fundamental constraints can be stored to handle more abstract geometric concepts, *e.g.* constraining lengths and angles rather than positions. The following list describes some of them:

- The MidPoint[p_1 p_2] constraint is the middle position of the line segment p_1p_2:

  MidPoint[p_1 p_2] := VSE[p_1 p_2] ∩ VSE[p_A p_B] ∩ CC[p_1 p_2]
  where: p_A ∈ CC[p_1 p_2] ∩ CC[p_2 p_1] ∩ VHP[p_1 p_2]
  p_B ∈ CC[p_1 p_2] ∩ CC[p_2 p_1] ∩ VHP[p_2 p_1]

The application p_0 ∈ MidPoint[p_1 p_2] is illustrated in figure 190. This constraint remains valid in the particular case where p_1 and p_2 are coincident.

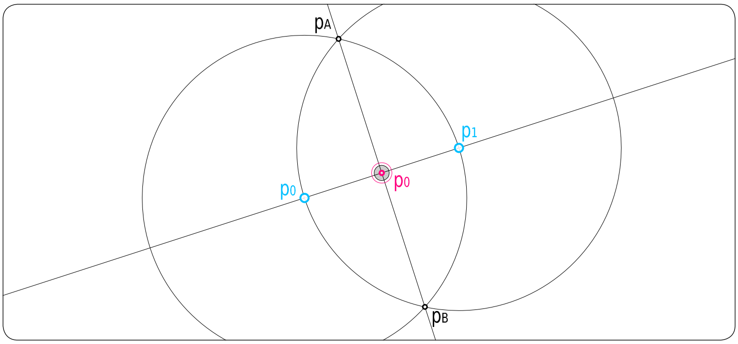


figure 190
construction and domain of p_0 ∈ MidPoint [p_1 p_2].

- A line passing through p_0 and p_1 is always perpendicular to a line passing through p_2 and p_3 if p_0 ∈ LinePerpendicularToLine[p_1 p_2 p_3] (figure 191) or if p_2 ∈ LinePerpendicularToLine[p_3 p_0 p_1]:

  LinePerpendicularLine[p_1 p_2 p_3] := Straightedge[p_1 p_A p_B]
  where: p_A ∈ CC[p_2 p_3] ∩ CC[p_3 p_2] ∩ VHP[p_2 p_3]
  p_B ∈ CC[p_2 p_3] ∩ CC[p_3 p_2] ∩ VHP[p_3 p_2]

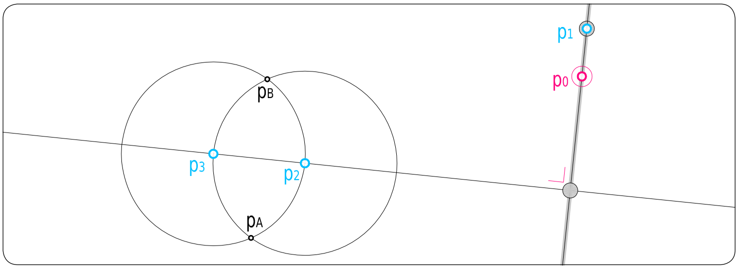


figure 191
construction and domain of p_0 ∈ LinePerpendicular ToLine[p_1 p_2 p_3].

For instance, if p_2 or p_3 moves, the position of p_0 is updated to its closest position and the lines remain perpendicular. If p_2 and p_3 are coincident, p_0 can be anywhere in the plane. The following constraints ensure that p_2p_3 and p_0p_1 are real lines:

$p_2 \in \backslash[p_3]$
$p_0 \in \backslash[p_1]$

• The TranslatedPosition[p_0 p_1 p_2] constraint gives the position that p_2 would have if it was translated according to the orientation going from p_0 to p_1 and to the distance p_0p_1 (figure 192):

TranslatedPosition[p_0 p_1 p_2] := $C_4 \cap C_5 \cap C_6$

where: C_0 = VeeringHalfplane[p_1 p_2]
C_1 = VeeringHalfplane[p_2 p_1]
C_2 = CollapsibleCompass[p_1 p_2]
C_3 = CollapsibleCompass[p_2 p_1]
C_4 = Compass[p_0 p_1 p_2]
C_5 = Straightedge[p_0 p_1 p_2]
C_6 = HalfPlane[p_0 p_A p_B]
$p_A \in C_0 \cap C_2 \cap C_3$
$p_B \in C_1 \cap C_2 \cap C_3$

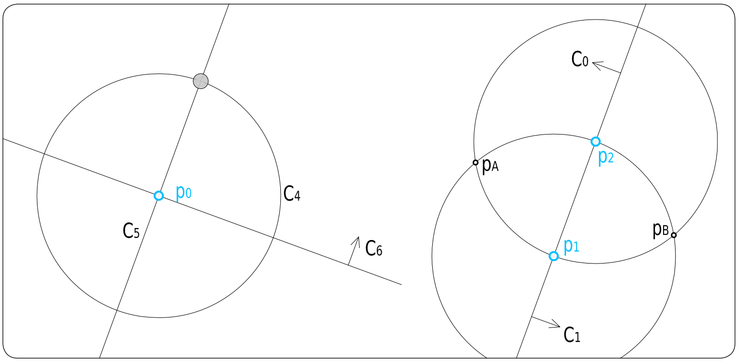


figure 192 construction and domain of TranslatedPosition [p_0 p_1 p_2] is highlighted in grey.

This constraint remains correct when p_1 and p_2 are coincident.

• If an angle α is defined by three points p_0, p_1 and p_2 as being the angle read clockwise between the line p_0p_1 and the line p_0p_2, then a line p_3p_4 always forms an angle α with a line p_5p_6 if $p_4 \in$ OrientedLine[p_3 p_5 p_6 p_0 p_1 p_2] (figure 193) such that:

OrientedLine[p_3 p_5 p_6 p_0 p_1 p_2] := (VSE[p_3 p_B] ∪ VSE[p_3 p_C]) − [p_3]

where: $p_A \in$ SE[p_3 p_5 p_6] ∩ C[p_3 p_0 p_1]
— two allowable positions but one unique final result
$p_B \in$ C[p_3 p_0 p_2] ∩ C[p_7 p_1 p_2] ∩ VHP[p_3 p_A]
$p_C \in$ C[p_3 p_0 p_2] ∩ C[p_A p_1 p_2] ∩ VHP[p_A p_3]

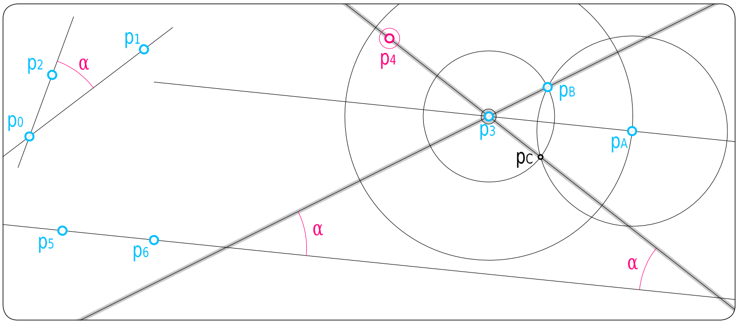


figure 193 construction and domain of $p_4 \in$ OrientedLine $[p_3\ p_5\ p_6\ p_0\ p_1\ p_2]$.

The result remains valid for any value of α, as long as the following conditions hold:

$p_1 \in \backslash[p_0]$
$p_2 \in \backslash[p_0]$
$p_3 \in \backslash[p_4]$
$p_5 \in \backslash[p_6]$

- A line passing through p_2p_3 is always tangential to a circle of centre p_0 and passing through p_1 if $p_3 \in$ LineTangentToCircle$[p_0\ p_1\ p_2]$ (figure 194):

LineTangentToCircle$[p_0\ p_1\ p_2]$:= (SE$[p_D\ p_F\ p_G]$ ∪ SE$[p_E\ p_H\ p_I]$) − $[p_2]$

where:

— *line* p_Ap_B *is perpendicular to line* p_0p_2 :

C_0 = CollapsibleCompass$[p_0\ p_2]$
C_1 = CollapsibleCompass$[p_2\ p_0]$
$p_A \in C_0 \cap C_1$
$p_B \in C_0 \cap C_1 - [p_A]$

— p_C *is midpoint of* p_0p_2 *:*

C_2 = VeeringHalfplane$[p_0\ p_2]$
C_3 = VeeringHalfplane$[p_2\ p_0]$
C_4 = VeeringStraightedge$[p_A\ p_B]$
$p_C \in C_2 \cap C_3 \cap C_4$

— p_D *and* p_E *are two points of tangency (they may be coincident with* p_2*) :*

C_5 = CollapsibleCompass$[p_C\ p_2]$
C_6 = CollapsibleCompass$[p_0\ p_1]$
$p_D \in C_2 \cap C_5 \cap C_6$
$p_E \in C_3 \cap C_5 \cap C_6$

— *line* p_Fp_G *is perpendicular to line* p_0p_D *:*

C_7 = CollapsibleCompass$[p_D\ p_0]$
$p_F \in C_6 \cap C_7$
$p_G \in C_6 \cap C_7 - [p_F]$

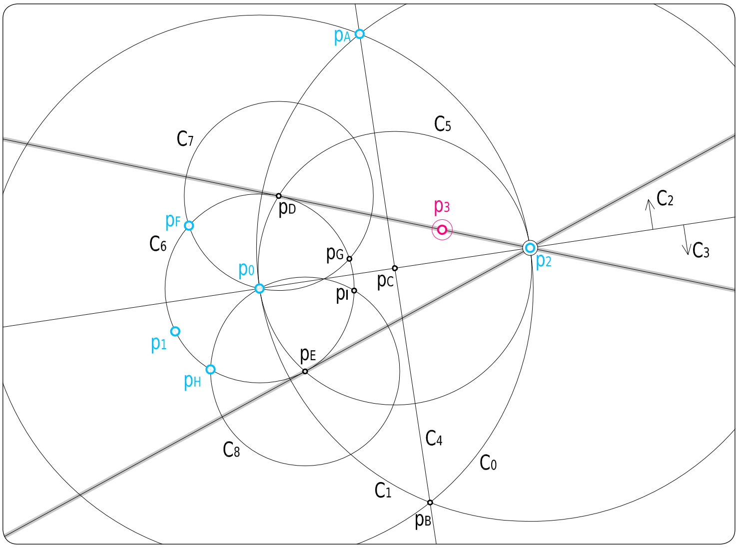
— *line* p_Hp_I *is perpendicular to* p_0p_E *:*

C_8 = CollapsibleCompass[p_E p_0]

$p_H \in C_6 \cap C_8$

$p_I \in C_6 \cap C_8 - [p_H]$

figure 194 construction and domain of $p_3 \in$ LineTangentToLine [p_0 p_1 p_2].



To be defined effectively, the problem should constrain p_2 such that it is not inside the circle: $p_2 \in$ CDO[p_0 p_1].

When p_2 is not superimposed onto the circle, lines p_2p_D and p_2p_E already provide the two tangents and the search for p_F, p_G, p_H and p_{11} is superfluous — *i.e.* SE[$p_Dp_Fp_G$] is equivalent to VSE[p_2p_D] and SE[$p_Dp_Hp_I$] is equivalent to VSE[p_2p_E]. However, points p_F, p_G, p_H and p_I are useful for finding the unique tangent when p_2 is superimposed onto the circle — in that case, SE[p_D p_F p_G] and SE[$p_Ep_Hp_I$] are equivalent.

- The distance between a point p_4 and a line p_0p_1 is equal to the distance between p_2 and p_3 if $p_4 \in$ OrthogonalDistanceToLine[p_0 p_1 p_2 p_3] (figure 195):

OrthogonalDistanceToLine[p_0 p_1 p_2 p_3] := SE[p_D p_0 p_1] ∪ SE[p_E p_0 p_1]

where: C_0 = CC[p_0 p_1]

C_1 = CC[p_1 p_0]

$p_A \in C_0 \cap C_1 \cap$ VHP[p_0 p_1]

$p_B \in C_0 \cap C_1 \cap$ VHP[p_1 p_0]

C_2 = VSE[p_A p_B]
C_3 = VSE[p_0 p_1]
$p_C \in C_2 \cap C_3$

C_4 = C[p_7 p_3 p_4]
C_5 = VHP[p_0 p_1]
C_6 = VHP[p_1 p_0]
$p_D \in C_2 \cap C_4$
$p_E \in C_2 \cap C_4 - [p_D]$

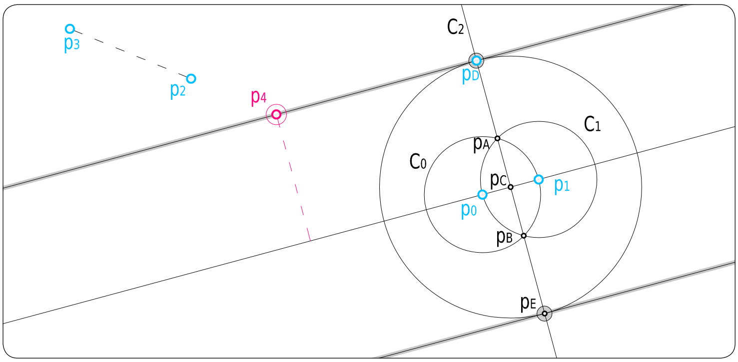


figure 195 construction and domain of $p_4 \in$ OrthogonalDistance ToLine [p_0 p_1 p_2 p_3].

arithmetic operations · This paragraph shows how new distances can be achieved through the addition, subtraction, division and multiplication of distances. Each of these new lengths is stored using a Compass constraint:

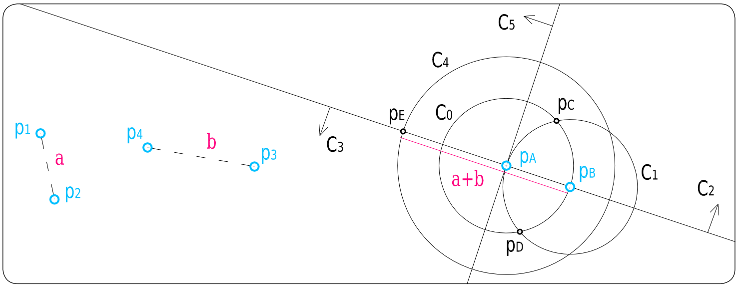
- the SumCompass[p_0 p_1 p_2 p_3 p_4] constraint returns a Compass constraint that is centred on p_0 and whose radius is the sum of the distances p_1p_2 and p_3p_4 (figure 196):

SumCompass[p_0 p_1 p_2 p_3 p_4] := Compass[p_0 p_B p_E]
where: C_0 = Compass[p_A p_1 p_2]
$p_B \in C_0$

C_1 = Compass[p_B p_1 p_2]
C_2 = VeeringHalfplane[p_A p_B]
C_3 = VeeringHalfplane[p_B p_A]
$p_C \in C_0 \cap C_1 \cap C_2$
$p_D \in C_0 \cap C_1 \cap C_3$

C_4 = Compass[p_A p_3 p_4]
C_5 = HalfPlane[p_A p_D p_C]
$p_E \in C_2 \cap C_3 \cap C_4 \cap C_5$

figure 196
construction of a SumCompass [p_0 p_1 p_2 p_3 p_4] constraint.



- the DifferenceCompass[p_0 p_1 p_2 p_3 p_4] constraint returns a Compass constraint that is centred on p_0 and whose radius is the difference between the distances p_1p_2 and p_3p_4 (figure 197):

DifferenceCompass[p_0 p_1 p_2 p_3 p_4] := Compass[p_0 p_B p_E]
where: C_0 = Compass[p_A p_1 p_2]
$p_B \in C_0$

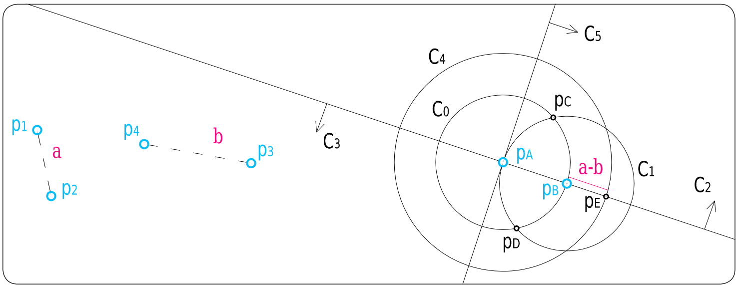
C_1 = Compass[p_B p_1 p_2]
C_2 = VeeringHalfplane[p_A p_B]
C_3 = VeeringHalfplane[p_B p_A]
$p_C \in C_0 \cap C_1 \cap C_2$
$p_D \in C_0 \cap C_1 \cap C_3$

C_4 = Compass[p_A p_3 p_4]
C_5 = HalfPlane[p_A p_C p_D]
$p_E \in C_2 \cap C_3 \cap C_4 \cap C_5$

With distances always positive, the following property always holds:

DifferenceCompass[p_0 p_1 p_2 p_3 p_4] = DifferenceCompass[p_0 p_3 p_4 p_1 p_2]

figure 197
construction of a DifferenceCompass [p_0 p_1 p_2 p_3 p_4] constraint.



- The ProductCompass[p_0 p_1 p_2 p_3 p_4] constraint returns a Compass constraint that is centred on p_0 and whose radius is the product of the distances p_1p_2 and p_3p_4 (figure 198):

ProductCompass[p_0 p_1 p_2 p_3 p_4] := Compass[p_0 p_A p_E]

where: C_0 = UnitCompass[p_A]

$p_B \in C_0$

C_1 = VeeringStraightedge[p_A p_B]

C_2 = Compass[p_A p_1 p_2]

$p_C \in C_2 \cap (\backslash C_1 \cup [p_A])$

C_3 = Compass[p_A p_3 p_4]

$p_D \in C_1 \cap C_3$ — *two available positions*

C_4 = VeeringStraightedge[p_A p_C]

C_5 = Straightedge[p_D p_B p_C]

$p_E \in (C_4 \cap C_5 - C_1) \cup [p_A]$

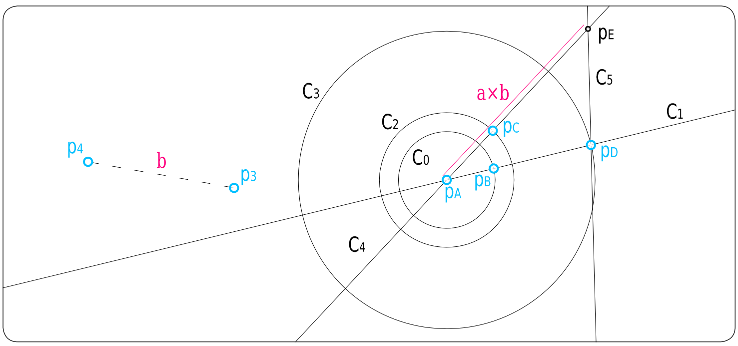


figure 198
construction of a ProductCompass [p_0 p_1 p_2 p_3 p_4] constraint.

This construction is a direct implementation of similar triangles. Two similar triangles have proportional sides. If the first triangle has sides of lengths a,b and e, and if the second triangle have sides of lengths c, d and f (figure 199), then the equality a/b=c/d holds. This can be written as a×d=b×c, which means that if one of these lengths is set to be equal to the unit length — given by the UnitCompass constraint, see the paragraph entitled “other fundamental constraints” (page 154) —, the other lengths consequently describe the multiplication of two lengths and its product.

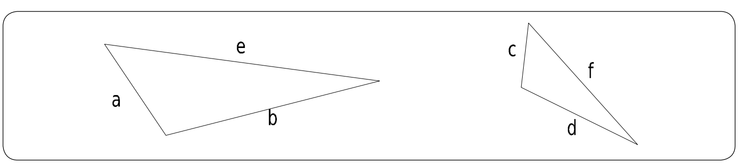


figure 199
two similar triangles.

The ProductCompass constraint remains valid when one length is set equal to zero. For example, if the length p_3p_4 is multiplied by a zero length, the radius of the ProductCompass[p_0 p_1 p_2 p_3 p_4] constraint will be zero as well:

$p_1 \in [p_2]$ $\longrightarrow C_2 = [p_A]$
$\longrightarrow p_C \in [p_A]$
$\longrightarrow C_4 = \Omega$ — *the entire plane*
$\longrightarrow C_5 = C_1$
$\longrightarrow p_E \in [p_A]$

The ProductCompass constraint multiplies lengths together. If, on the other hand, a given length has to be multiplied by a given amount, *i.e.* by an integer, the SumCompass constraint should be used instead and applied as many times as desired.

- The QuotientCompass[p_0 p_1 p_2 p_3 p_4] constraint returns a Compass constraint that is centred on p_0 and whose radius is the division of the distance p_1p_2 by the distance p_3p_4 (figure 200):

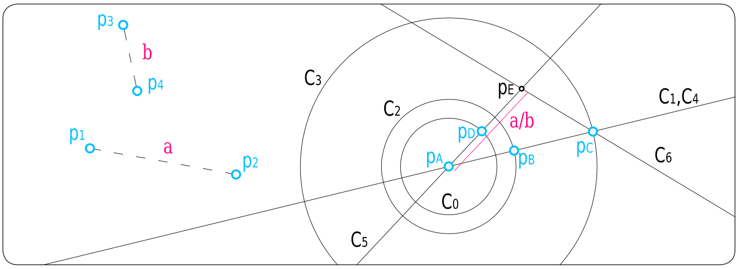
QuotientCompass[p_0 p_1 p_2 p_3 p_4] := Compass[p_0 p_A p_E]
where: C_0 = Compass[p_A p_3 p_4]
$p_B \in C_0$

C_1 = VeeringStraightedge[p_A p_B]
C_2 = Compass[p_A p_1 p_2]
$p_C \in C_1 \cap C_2$

C_3 = UnitCompass[p_A]
C_4 = VeeringStraightedge[p_A p_C]
$p_D \in C_3 \cap (\backslash C_4 \cup [p_A])$

C_5 = VeeringStraightedge[p_A p_D]
C_6 = Straightedge[p_C p_B p_D]
$p_E \in C_5 \cap C_6$

figure 200
construction of a QuotientCompass [p_0 p_1 p_2 p_3 p_4] constraint.



Again, this constraint remains valid for particular divisions: (1) 0/x, (2) x/0, and (3) 0/0:

(1) *when* p_0 *and* p_1 *are coincident:*

$C_2 = [p_A]$
⟶ $p_C \in [p_A]$
⟶ $p_E \in [p_A]$
⟶ *the radius* $p_A p_9$ *is zero.*

(2) *when* p_3 *and* p_4 *are coincident:*

$C_0 = [p_A]$
⟶ $p_6 \in [p_A]$
⟶ $C_1 = \Omega$
⟶ $p_C \in C_2$
⟶ C_6 *and* C_5 *are parallel*
⟶ $p_E \in \emptyset$ *and the construction is not allowed*

(3) *when couples* $p_0 p_1$ *and* $p_3 p_4$ *are simultaneously coincident:*

$C_0 = [p_A]$
⟶ $p_B \in [p_A]$
⟶ $C_1 = \Omega$
⟶ $C_2 = [p_A]$
⟶ $p_C \in [p_A]$
⟶ $C_4 = \Omega$
⟶ $p_D \in [p_A]$
⟶ $C_5 = \Omega$
⟶ $C_6 = \Omega$
⟶ $p_E \in \Omega$
⟶ *the distance* $p_A p_E$ *can be any desired length*

trigonometric operations · Trigonometric operations can be performed using similar non-fundamental Compass constraints.

- The CosineCompass[p_0 p_1 p_2 p_3 p_4] constraint returns a Compass constraint that is centred on p_0 and whose radius is the cosine of the angle that is read clockwise from orientation $p_1 p_2$ to orientation $p_3 p_4$ (figure 201):

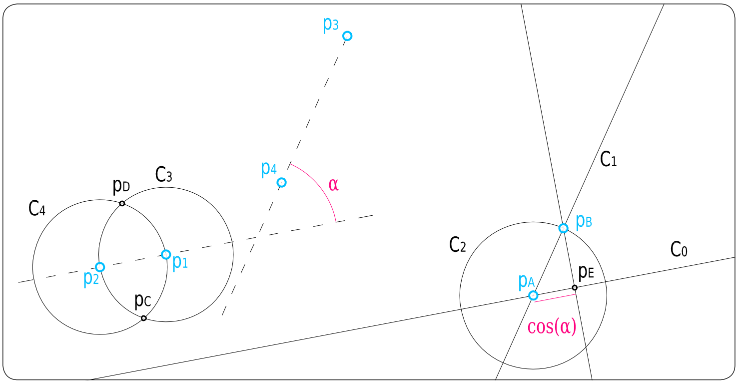
CosineCompass[p_0 p_1 p_2 p_3 p_4] := Compass[p_0 p_A p_E]
where: C_0 = Straightedge[p_A p_1 p_2]
C_1 = Straightedge[p_A p_3 p_4]
C_2 = UnitCompass[p_A]
$p_B \in C_1 \cap C_2$ — *two available positions but one unique final result*

C_3 = CollapsibleCompass[p_1 p_2]
C_4 = CollapsibleCompass[p_2 p_1]
$p_C \in C_3 \cap C_4$
$p_D \in C_3 \cap C_4 - [p_C]$

C_5 = Straightedge[p_B p_C p_D]
$p_E \in C_0 \cap C_5$

The computed cosine remains correct when both orientations are distinct parallels — *i.e.* cos[0]=1.

figure 201
construction of a CosineCompass [p_0 p_1 p_2 p_3 p_4] constraint.



• The SineCompass[p_0 p_1 p_2 p_3 p_4] constraint returns a Compass constraint that is centred on p_0 and whose radius is the sine of the angle that is read clockwise from orientation p_1p_2 to orientation p_3p_4 (figure 202):

SineCompass[p_0 p_1 p_2 p_3 p_4] := Compass[p_0 p_A p_E]
where: C_0 = Straightedge[p_A p_3 p_4]
C_1 = UnitCompass[p_A]
$p_B \in C_1 \cap C_2$ — *two allowable positions but one unique final result*

C_2 = CollapsibleCompass[p_1 p_2]
C_3 = CollapsibleCompass[p_2 p_1]
$p_C \in C_2 \cap C_3$

$p_D \in C_2 \cap C_3 - [p_C]$
C_4 = Straightedge[p_A p_C p_D]
C_5 = Straightedge[p_B p_1 p_2]
$p_E \in C_4 \cap C_5$

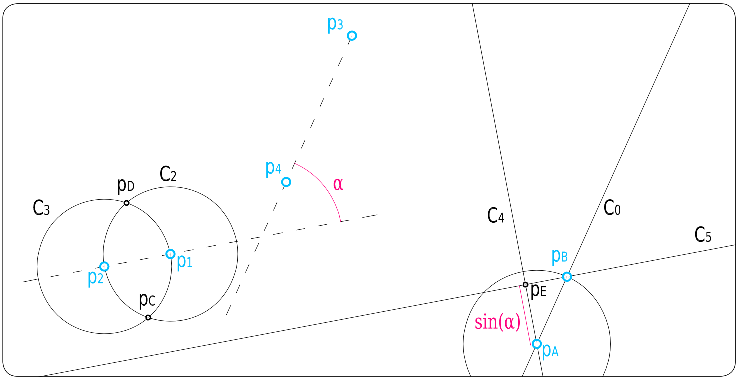


figure 202 construction of a SineCompass [p_0 p_1 p_2 p_3 p_4] constraint.

isometric transformations of constraints using mechanical instrument analogies · Applying the TranslatedPosition constraint (page 167) on each point defining another constraint C* is equivalent to copying and translating that constraint C*. In order to perform other isometric transformations of constraints, non-fundamental constraints can be constructed by analogy with linkages having similar purposes (Reuleaux·1876, Kempe·1877, Barr·1899 and Hinkle·1953). For example, the implementation of a pantograph (figure 203) would build a constraint that copies and scales constraints.

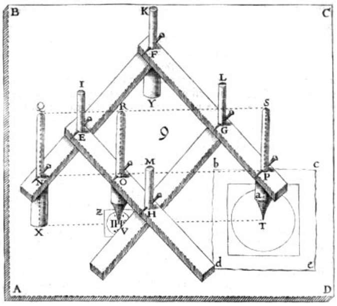


figure 203 Scheiner's pantograph (in Scheiner·1631, page 29).

The following lines implement a plagiograph (figure 204, figure 205). The purpose of this instrument is to reproduce the rotation of a given drawing. The Plagiograph[p_0 p_1 p_2 p_3 p_4 p_5 p_6 p_7] constraint is a position obtained by rotating p_7 (hinge E) around p_6 (hinge I). The angle of rotation α is defined by the angle between p_2p_3 and p_4p_5. The distance p_0p_1 is half the greatest radius p_6p_7 — *e.g.* p_0p_1 defines the length of rods O-A, B-C, O-B, A-D and B-E.

figure 204 Sylvester's plagiograph (in Bartolini/… 2006, page 132).

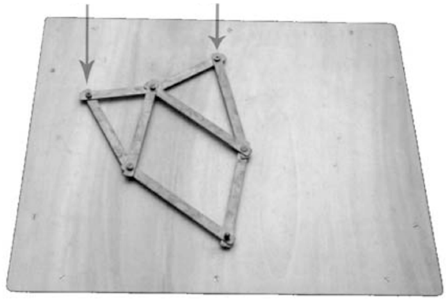
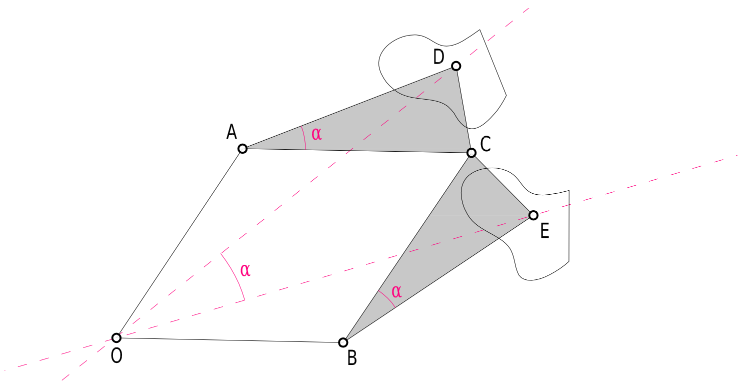


figure 205 mechanism of Sylvester's plagiograph.



That constraint is shown in figure 206:

Plagiograph[p_0 p_1 p_2 p_3 p_4 p_5 p_6 p_7] := $C_{11} \cap C_{12} \cap C_{13}$

where: — *measure of the rotation angle :*

C_0 = VeeringStraightedge[p_2 p_3]

C_1 = VeeringStraightedge[p_4 p_5]

$p_A \in C_0 \cap C_1$

C_2 = Compass[p_A p_0 p_1]

$p_B \in C_0 \cap C_2$ — *two allowable positions but one unique final result*

C_3 = VeeringHalfplane[p_B p_A]

$p_C \in C_1 \cap C_2 \cap C_3$

— *search for* p_D *= hinge B*

C_4 = Compass[p_6 p_0 p_1]

C_5 = Compass[p_7 p_0 p_1]

C_6 = VeeringHalfplane[p_7 p_6]

$p_D \in C_4 \cap C_5 \cap C_6$

— *search for* p_E *= hinge C*

C_7 = Compass[p_D p_0 p_1]

C_8 = Compass[p_7 p_B p_C]

C_9 = VeeringHalfplane[p_6 p_7]

$p_E \in C_7 \cap C_8 \cap C_9$

— *search for* p_F *= hinge A*

C_{10} = Compass[p_E p_0 p_1]

$p_F \in C_4 \cap C_{10} - [p_D]$

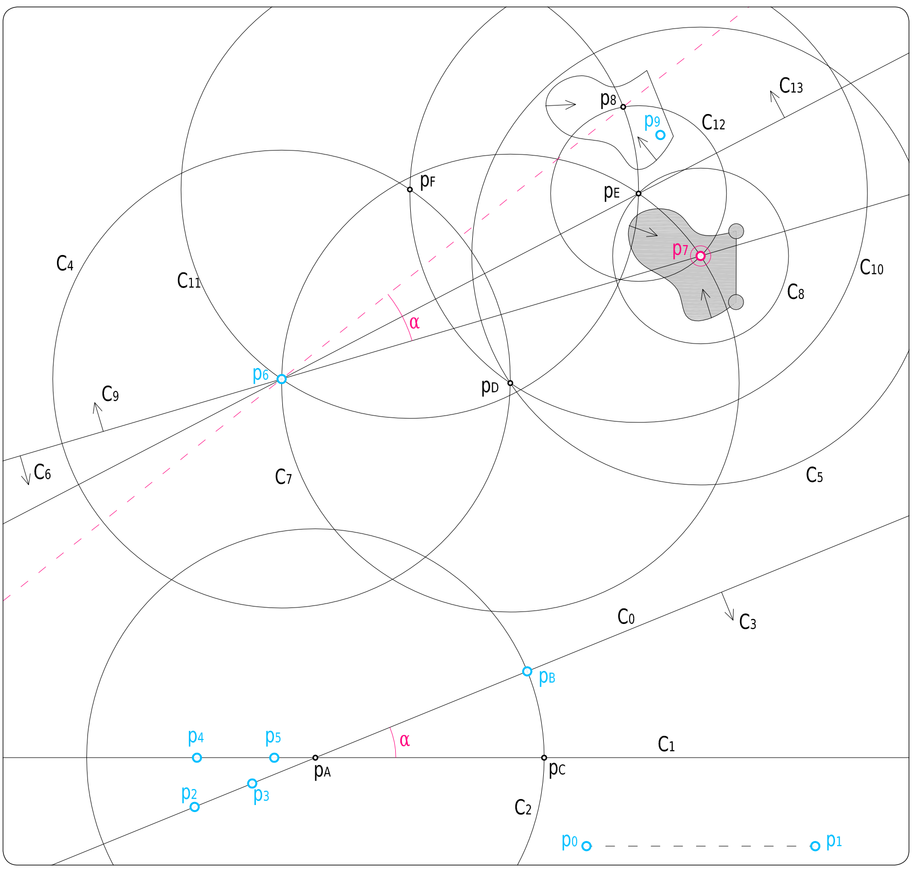
— *search for the constraint that defines hinge D*

C_{11} = Compass[p_F p_0 p_1]

C_{12} = Compass[p_E p_{10} p_{11}]

C_{13} = VeeringHalfplane[p_6 p_E]

figure 206
construction of $p_8 \in$ Plagiograph [p_0 p_1 p_2 p_3 p_4 p_5 p_6 p_7] constraint; the domain of p_7 is highlighted.



approximations of transcendental numbers by finite constructions · Some numbers (called transcendental numbers) cannot accurately be obtained by classical compass-and-straightedge constructions. However they can be approximate. For instance, the following constraint calculates the value of π to four decimal places. It is based on a construction of Kochanski (Kochanski·1685 and Kochanski/Fuks·2011, figure 207) that computes the square root of $2^2+(3\text{-}1/\sqrt{3})^2$ (figure 208):

KochanskiPiCompass[p_0] := Compass[p_0 p_G p_I]
where: $p_B \in$ UnitCompass[p_A]

— *search for* p_Cp_D = *perpendicular to* p_Ap_B*:*
C_0 = VeeringHalfplane[p_A p_B]
C_1 = VeeringHalfplane[p_B p_A]
C_2 = CollapsibleCompass[p_A p_B]
C_3 = CollapsibleCompass[p_B p_A]
$p_C \in C_0 \cap C_2 \cap C_3$
$p_D \in C_1 \cap C_2 \cap C_3$

— *search for* p_E = *intersection between circles* C_2 *and* C_6*:*
C_4 = HalfPlane[p_A p_D p_C]
C_5 = HalfPlane[p_A p_C p_D]
C_6 = Compass[p_C p_A p_B]
$p_E \in C_2 \cap C_4 \cap C_6$

— *search for* p_F = *intersection between line* p_Bp_E *and its perpendicular passing through* p_A :
C_7 = VeeringStraightedge[p_B p_E]
$p_F \in C_4 \cap C_5 \cap C_7$

— *search for* p_Gp_H = 3 *units of length:*
$p_G \in C_0 \cap C_1 \cap C_3 \cap \backslash C_4$
$p_H \in C_0 \cap C_1 \cap C_2 \cap C_4$

— *search for* p_Gp_I = 3 *units of length - distance* p_Ap_F*:*
C_8 = Compass[p_F p_G p_H]
$p_I \in C_8 \cap C_4 \cap C_5 \cap C_1$

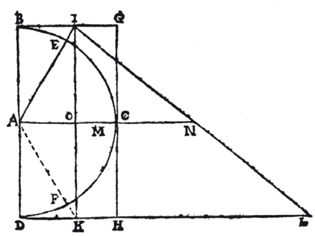


figure 207
approximation of Pi by Kochanski (in Kochanski·1685, page 397).

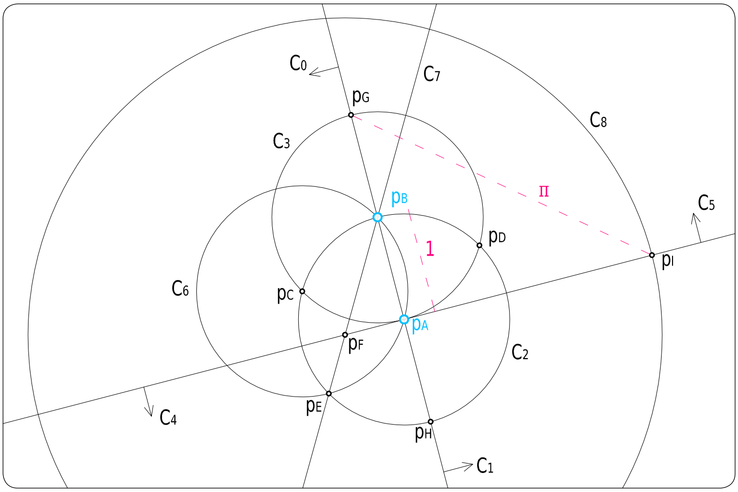


figure 208
construction of a PiCompass [p_0] constraint.

Other constructions of π are more accurate but also longer, *e.g.* the approximation provided by Srivanasa Ramanujan (Ramanujan·1914) calculates the value of π to eight decimal places, but requires 28 fundamental constraints.

constraints that go beyond the limitations of compass-and-straightedge constructions · Classical compass-and-straightedge constructions are unable to perform the following problems:

- quadrature of the circle, *i.e.* defining the value of π (Klein/Tägert·1897 page 78)
- doubling the cube, *i.e.* defining the cubic root of 2
- trisecting any angle (Holme·2010, page 420)
- drawing any regular polygons (Eeckhoff·1999)
- constructing transcendental lengths
- …

However, these problems can all be constructed with interdependent HalfPlane, DiscInside and DiscOutside constraints. The following examples closely mimic traditional linkages, *e.g.* Neusis constructions, Philo lines and Laisant's mechanism, in order to calculate the cubic root of 2 and to divide an angle into n equivalent parts.

- A Neusis construction rotates a given marked ruler around a fixed point until the distance between two curves equals a given length (measured on the marked ruler) (figure 209). Constructions of this kind are useful for doubling the cube, trisecting angles and constructing regular polygons with, for instance, 7, 9 or 13 sides. It is implemented here with the Neusis[p_0 p_1 p_2 p_3 p_4 p_5 p_6] constraint. This constraint is a Straightedge passing through p_4 and crossing the lines p_0p_1 and p_2p_3 by a distance p_5p_6 (figure 210).

figure 209
a traditional Neusis construction.

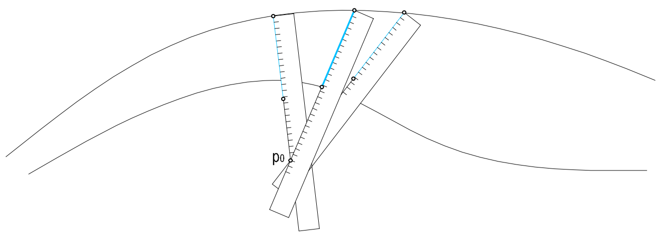
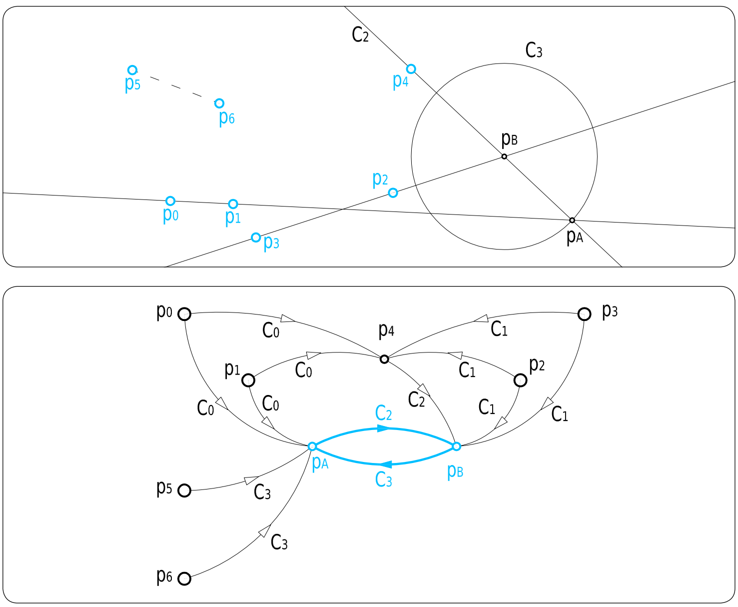


figure 210
construction of a Neusis [p_0 p_1 p_2 p_3 p_4 p_5 p_6] constraint and its graph of dependencies.



By successive displacements to their closest position, points p_A and p_B converge to the expected orientation of the Neusis line:

Neusis[p_0 p_1 p_2 p_3 p_4 p_5 p_6] := C_2
where: C_0 = VeeringStraightedge[p_0 p_1]
C_1 = VeeringStraightedge[p_2 p_3]
C_2 = VeeringStraightedge[p_4 p_A]

C_3 = Compass[p_B p_5 p_6]
$p_4 \in \backslash(C_0 \cap C_1)$
$p_A \in C_0 \cap C_3$
$p_B \in C_1 \cap C_2$

If p_A and p_B are two given points that belong to non-regular curves, *e.g.* unions of arcs of circles instead of straight lines, the Neusis is obtained by constraining p_A onto C_3 and p_8 onto C_2.

The following construction uses the Neusis constraint in order to calculate the cubic root of 2 (figure 211):

CubicRootOf2Compass[p_0] := Compass[p_0 p_C p_E]
where: C_0 = UnitCompass[p_A]
$p_B \in C_0$

— *search for the equilateral triangle* $p_Ap_Bp_C$*:*
C_1 = CollapsibleCompass[p_B p_A]
$p_C \in C_0 \cap C_1$ — *two available positions*

— *search for* p_Cp_D *= two units of lengths and parallel to* p_Bp_C*:*
C_4 = VeeringStraightedge[p_B p_C]
$p_D \in C_1 \cap C_2 \cap \backslash(p_C)$

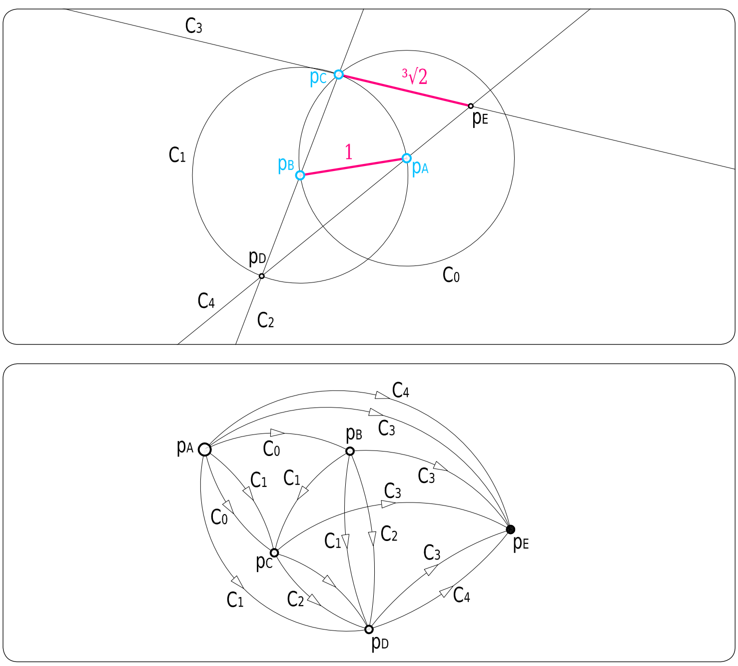


figure 211
construction of a CubicRootOfTwo Compass[p_5] constraint and its graph of dependencies.

— *search for a unit segment between lines* $p_A p_B$ *and* $p_A p_D$*:*

C_3 = Neusis[p_A p_B p_A p_D p_C p_A p_B]

— *search for* $p_C p_E = \sqrt[3]{2}$ *units of length:*

C_4 = VeeringStraightedge[p_A p_D]
$p_E \in C_3 \cap C_4$

The proof of this construction can be found in Hartshorne·2000.

- A Philo line is the closest segment between two given lines — p_0p_1 and p_0p_2 in figure 212 — and passing through a given point — p_3. If p_4 is defined as the base of the altitude of triangle $p_0p_1p_2$ passing through p_0, then the distance p_2p_4 is one third of the distance p_1p_2.

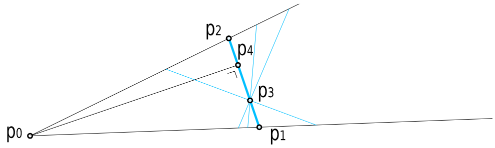


figure 212
rotation of a Philo line around p_3.

A Philo line joining two lines p_0p_1 and p_2p_3 and passing through p_4 can be implemented as follows (figure 213):

Philo[p_0 p_1 p_2 p_3 p_4] := C_2

where: C_0 = VeeringStraightedge[p_0 p_1]
C_1 = VeeringStraightedge[p_2 p_3]
C_2 = VeeringStraightedge[p_4 p_A]
$p_A \in C_0 \cap$ CollapsibleOutsideDisc[p_B p_4]
$p_B \in C_1 \cap C_2 \cap$ CollapsibleOutsideDisc[p_A p_4]

C_3 = CollapsibleCompass[p_4 p_B]
C_4 = CollapsibleCompass[p_B p_4]
$p_C \in C_3 \cap C_4$
$p_D \in C_3 \cap C_4 -$ [p_C]

$p_E = C_0 \cap C_1$
C_5 = Straightedge[p_E p_C p_D]
$p_F \in C_2 \cap C_5$

C_6 = Compass[p_4 p_B p_F]
$p_A \in C_6$

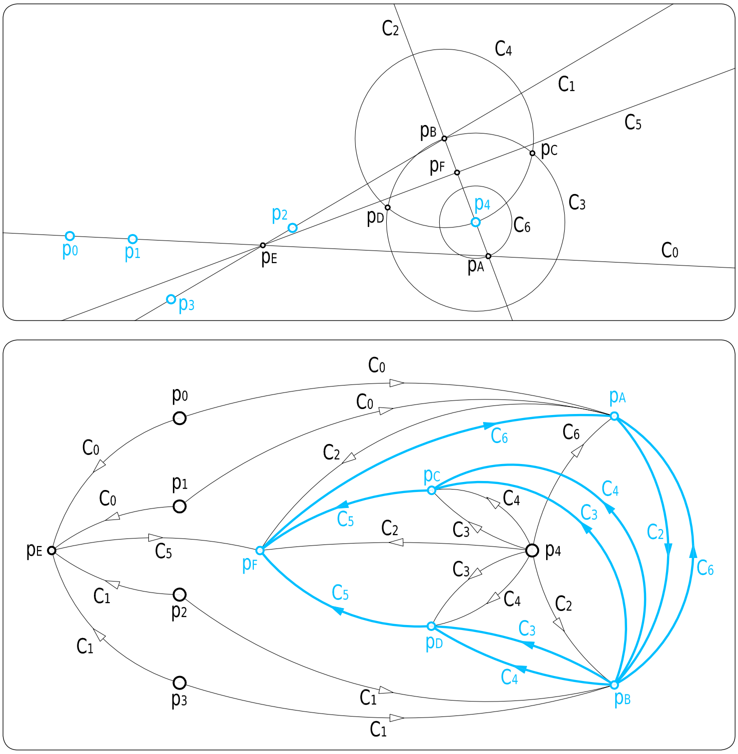


figure 213
construction of a PhiloLine[p_0 p_1 p_2 p_3 p_4] constraint and its graph of dependencies.

Philo of Byzantium (ca. 280 BC – ca. 220 BC) used this line in order to produce the cubic root of 2 on the basis of three similar triangles (figure 214).

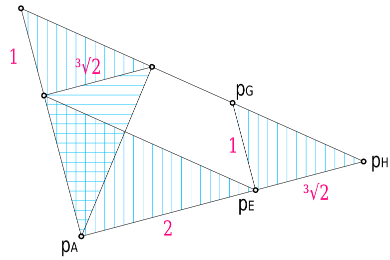


figure 214
proof of the construction of the cubic root of 2 using a Philo line.

Its construction can be implemented as follows (figure 215):

CubicRootOfTwoCompass[p_0] := Compass[p_0 p_E p_H]

where: C_0 = UnitCompass[p_A]

$p_B \in C_0$

C_1 = CollapsibleCompass[p_B p_A]
C_2 = VeeringStraightedge[p_A p_B]
$p_C \in C_0 \cap C_1$
$p_D \in C_0 \cap C_1 -$ [p_C]
C_3 = Straightedge[p_A p_C p_D]

$p_E \in C_0 \cap C_2 -$ [p_A]
C_4 = Straightedge[p_E p_C p_D]
$p_F \in C_1 \cap C_3$
C_5 = Straightedge[p_F p_A p_B]
$p_G \in C_4 \cap C_5$

C_6 = Philo[p_A p_B p_A p_F p_G]
$p_H \in C_0 \cap C_6$

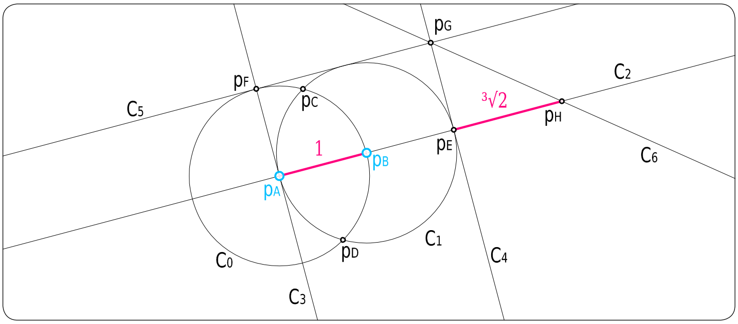


figure 215
construction of a CubicRootOfTwo Compass[p_8] constraint.

• The original purpose of a Laisant's mechanism (Brocard·1875 and Yates·1941) is to divide an angle into three equal sectors (figure 216) but it can be generalised to divide an angle into any number of sectors (figure 217).

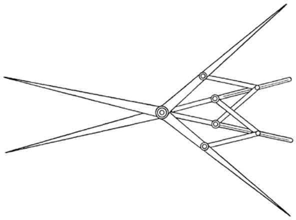


figure 216
a Laisant's mechanism (from Yates·1941).

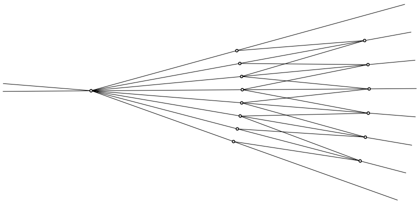


figure 217
generalisation of Laisant's mechanism.

If the angle to be divided is given by lines p_0p_1 and p_2p_3, the constraint LaisantTrisector[p_0 p_1 p_2 p_3] returns the union of the two straightedges that divide that angle into three equal parts (figure 218):

LaisantTrisector[p_0 p_1 p_2 p_3] := $C_6 \cup C_8$

where: C_0 = VeeringStraightedge[p_0 p_1]
C_1 = VeeringStraightedge[p_2 p_3]
$p_A \in C_0 \cap C_1$

$p_B \in C_1 - [p_A]$
C_2 = VeeringHalfplane[p_A p_B]
C_3 = CollapsibleCompass[p_A p_B]
$p_C \in C_0 \cap C_2 \cap C_3$
C_4 = CollapsibleCompass[p_C p_A]

$p_D \in C_0 \cap C_4 - [p_A]$
C_5 = CollapsibleCompass[p_A p_D]

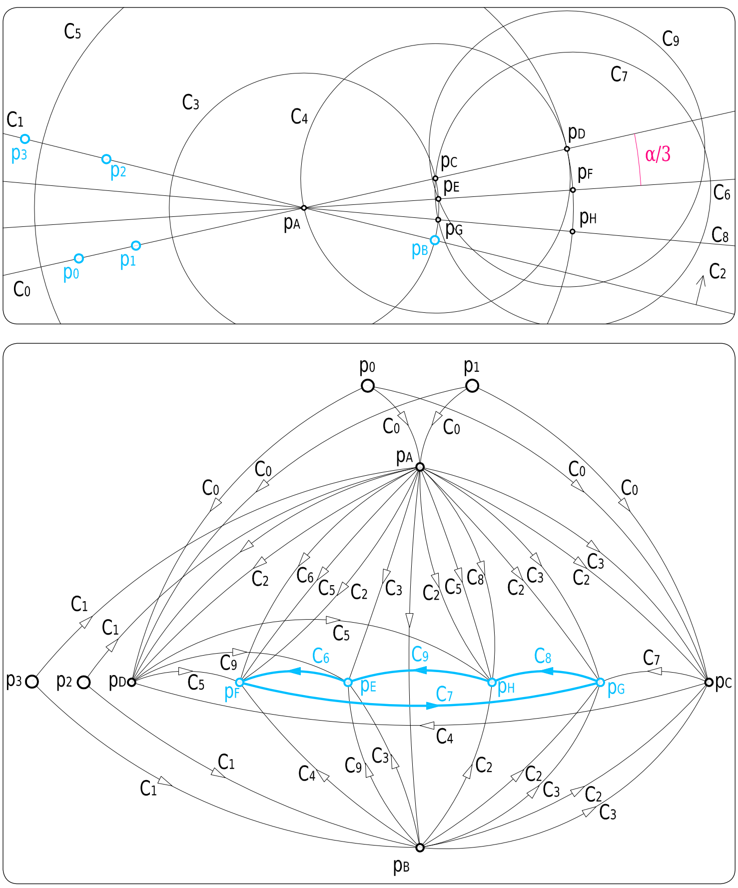


figure 218
construction of a LaisantTrisector [p_0 p_1 p_2 p_3] constraint and its graph of dependencies.

— first trisector:

$p_E \in C_3$
C_6 = VeeringStraightedge[p_A p_E]
$p_F \in C_5 \cap C_6 \cap C_2$

— second trisector:

C_7 = CollapsibleCompass[p_F p_C]
$p_G \in C_2 \cap C_3 \cap C_7$
C_8 = VeeringStraightedge[p_A p_G]
$p_H \in C_2 \cap C_5 \cap C_8$

— final adjustment:

C_9 = Compass[p_D p_B p_H]
$p_E \in C_9$

Other methods are available for trisecting an angle. These include Pascal's marked ruler (Archimedes/...·1808, page 396, Yates·1941 and Loy·2003) or the tomahawk.

constraining points on curves using geometric properties · The interdependencies of fundamental constraints can also be used to constrain points on curves other than the circle. Either the construction of the curves is based on particular geometric properties or they perform polynomial functions graphically. This paragraph develops three conic curves based on their geometric properties. Examples of polynomial functions will be given subsequently.

- The Ellipse[p_0 p_1 p_2 p_3 p_4] constraint compels a point to be along the ellipse defined by these properties: (1) the ellipse is centred on p_0; (2) the distance between focal points is twice the distance p_0p_1; (3) the major radius is of orientation p_0p_1; and (4) the major radius is of length p_2p_3. p_4 is the point that will be constrained, such that $p_4 \in$ Ellipse[p_0 p_1 p_2 p_3 p_4]. The construction makes use of the fact that the sum of distances p_1p_4 and p_Bp_4 is always equal to twice the distance p_2p_3 (figure 219):

Ellipse[p_0 p_1 p_2 p_3 p_4] := $C_2 \cap C_8$

where: — p_2p_A *is the length of the major radius:*

C_0 = VeeringStraightedge[p_2 p_3]
C_1 = CollapsibleCompass[p_3 p_2]
$p_A \in (C_0 \cap C_1) - [p_2]$

— the distance between the focal points is lower than or equal to the length of the major radius:

C_2 = InsideDisc[p_0 p_2 p_3]
$p_1 \in C_2$

— p_1 *and* p_B *are the two focal points:*

C_3 = VeeringStraightedge[p_0 p_1]

C_4 = CollapsibleCompass[p_0 p_1]

$p_B \in C_3 \cap C_4 - [p_1]$

— p_C *is constrained by the distance* p_1p_4 *and constrains the distance* p_4p_B*:*

C_5 = CollapsibleInsideDisc[p_2 p_3]

C_6 = CollapsibleInsideDisc[p_3 p_2]

C_7 = Compass[p_2 p_1 p_4]

$p_C \in C_0 \cap C_5 \cap C_6 \cap C_7$

— *constraining the distance* p_4p_B*:*

C_8 = Compass[p_B p_A p_C]

$p_4 \in$ Ellipse[p_0 p_1 p_2 p_3 p_4]

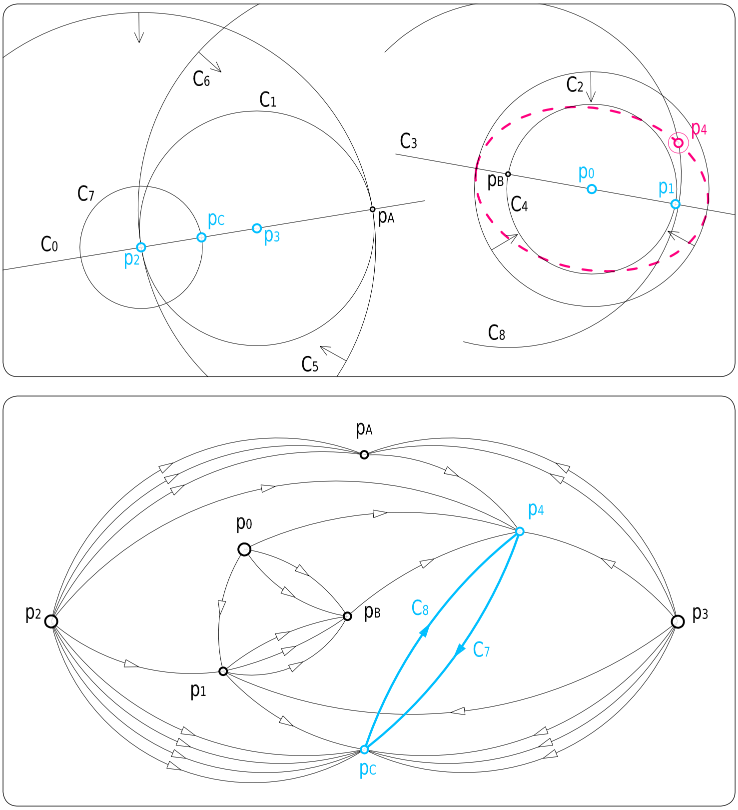


figure 219 construction of $p_4 \in$ Ellipse [p_0 p_1 p_2 p_3 p_4] constraint and its graph of dependencies.

- Constraining a point p_3 on a Parabola[p_0 p_1 p_2 p_3] constraint would force it to stay on a parabola such that: (1) the focus is p_0; (2) the axis of symmetry is parallel to p_1p_2; and (3) the distance between the focus and the directrix is equal to p_1p_2. The construction uses the fact that the distance between a point p_3 and the directrix is always equal to the distance between that point and the focus (figure 220):

Parabola[p_0 p_1 p_2 p_3] := C_9

where: C_0 = VeeringHalfplane[p_1 p_2]
C_1 = VeeringHalfplane[p_2 p_1]
C_2 = CollapsibleCompass[p_1 p_2]
C_3 = CollapsibleCompass[p_2 p_1]
$p_A \in C_0 \cap C_2 \cap C_3$
$p_B \in C_1 \cap C_2 \cap C_3$

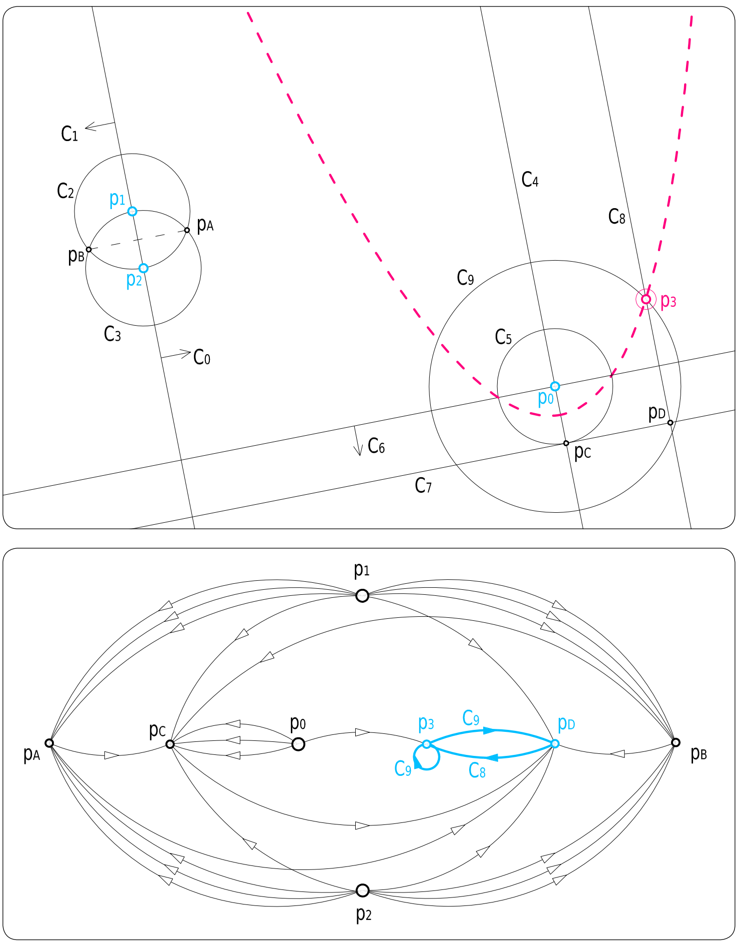


figure 220 construction of $p_3 \in$ Parabola [p_0 p_1 p_2 p_3] constraint and its graph of dependencies.

C_4 = Straightedge[p_0 p_1 p_2]
C_5 = Compass[p_0 p_1 p_2]
C_6 = HalfPlane[p_0 p_A p_B]
$p_C \in C_4 \cap C_5 \cap C_6$
C_7 = Straightedge[p_C p_A p_B]

C_8 = Straightedge[p_3 p_1 p_2]
$p_D \in C_7 \cap C_8$
C_9 = Compass[p_0 p_3 p_D]

$p_3 \in$ Parabola[p_0 p_1 p_2 p_3]

- Constraining points on curves may also be done using properties of mechanical instruments. The Hyperbola[p_0 p_1 p_2 p_3 p_4] constraint is constructed here using the mechanism in figure 221. Points p_0 and p_1 are the focus and the magnitude of the hyperbola is given by the distance p_2p_3. Point p_4 is the point to be constrained on the hyperbola (figure 222):

Hyperbola[p_0 p_1 p_2 p_3 p_4] := VeeringStraightedge[p_1 p_B]
where: C_0 = VeeringStraightedge[p_0 p_4]
C_1 = Compass[p_0 p_2 p_3]
C_2 = Compass[p_A p_0 p_1]
C_3 = Compass[p_1 p_2 p_3]
C_4 = Straightedge[p_1 p_0 p_A]
$p_A \in C_0 \cap C_1$
$p_B \in C_2 \cap C_3 - C_4$

$p_4 \in$ Hyperbola[p_0 p_1 p_2 p_3 p_4]

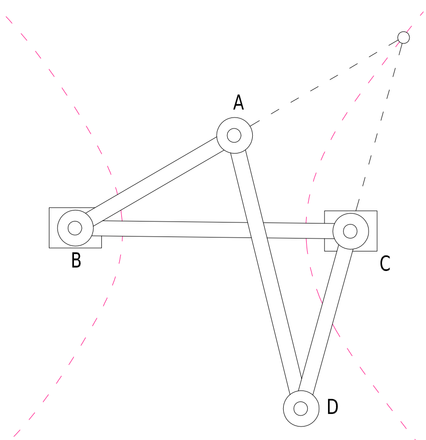
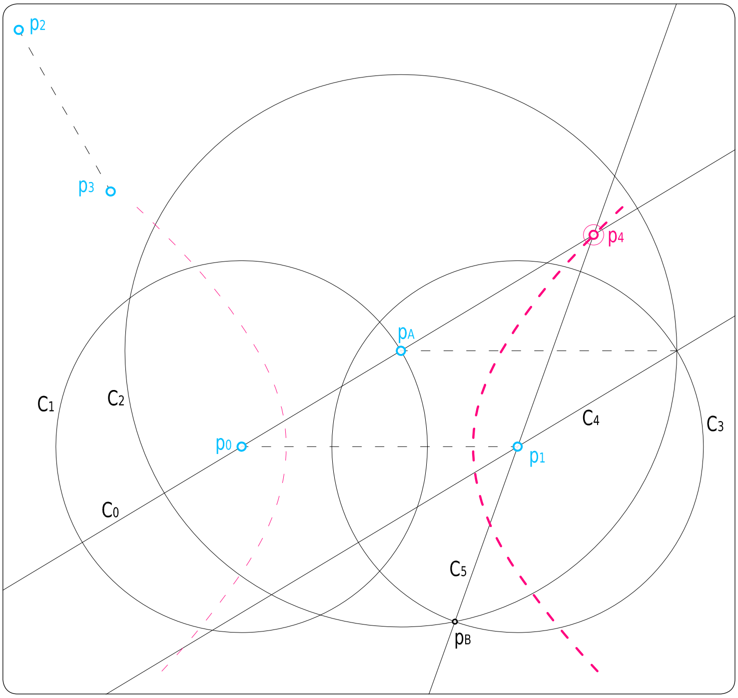


figure 221 mechanism used to draw a hyperbola (from Yates·1959, page 183, figure 168b).



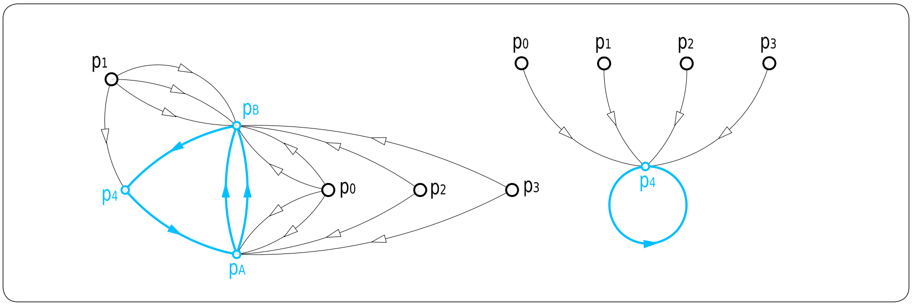


figure 222
construction of p_4 ∈ Hyperbola [p_0 p_1 p_2 p_3 p_4] constraint and its graph of dependencies.

constraining points on curves using analogies of polynomial functions · Finally, curves can be constructed using polynomial functions — *i.e.* by applying multiple arithmetic and trigonometric constraints. See the paragraphs entitled “arithmetic operations” (page 170) and “trigonometric operations” (page 174).

As an illustration, Hankinson’s curve (Hankinson·1921) is constructed. This curve is used to describe the variation of the oblique strength in an orthotropic material according to its two principal (orthogonal) allowable strengths. This strength is calculated for an angle α and is given by equation $(f_1 f_2) / (f_1 \sin^2[\alpha] + f_2 \cos^2[\alpha])$ where F_1 and F_2 are the two principal stresses.

$\text{Hankinson}[p_0\ p_1\ p_2\ p_3\ p_4\ p_5\ p_6]$ is a constraint that compels p_6 to be within the polar representation of Hankinson's curve. Point p_0 is the centre of this shape — moving this point means moving the entire shape. Points p_0 and p_1 give the orientation of the principal stress. The distance p_2p_3 is equal to F_1 and the distance p_4p_5 is equal to F_2. This construction is illustrated in figure 223:

$\text{Hankinson}[p_0\ p_1\ p_2\ p_3\ p_4\ p_5\ p_6] := \text{CollapsibleInsideDisc}[p_0\ p_I]$

where: — distance $p_0p_A = \cos[\alpha]$ *and distance* $p_0p_B = \sin[\alpha]$*:*

$C_0 = \text{CosineCompass}[p_0\ p_0\ p_1\ p_0\ p_6]$
$C_1 = \text{SineCompass}[p_0\ p_0\ p_1\ p_0\ p_6]$
$p_A \in C_0$
$p_B \in C_1$

— distance $p_0p_C = \cos^2[\alpha]$ *and distance* $p_0p_D = \sin^2[\alpha]$*:*

$C_2 = \text{ProductCompass}[p_0\ p_0\ p_A\ p_0\ p_A]$
$C_3 = \text{ProductCompass}[p_0\ p_0\ p_B\ p_0\ p_B]$
$p_C \in C_2$
$p_D \in C_3$

— distance $p_0p_E = f_2\times\cos^2[\alpha]$ *and distance* $p_0p_F = f_1\times\sin^2[\alpha]$*:*

$C_4 = \text{ProductCompass}[p_0\ p_0\ p_C\ p_4\ p_5]$
$C_5 = \text{ProductCompass}[p_0\ p_0\ p_D\ p_2\ p_3]$
$p_E \in C_4$
$p_F \in C_5$

— distance $p_0p_G = f_2\times\cos^2[\alpha] + f_1\times\sin^2[\alpha]$*:*

$C_6 = \text{SumCompass}[p_0\ p_0\ p_E\ p_0\ p_F]$
$p_G \in C_6$

— distance $p_0p_H + f_1\times f_2$*:*

$C_7 = \text{ProductCompass}[p_0\ p_2\ p_3\ p_4\ p_5]$
$p_H \in C_7$

— distance $p_0p_I = f_1\times f_2 / (\ f_2\times\cos^2[\alpha] + f_1\times\sin^2[\alpha]\ )$*:*

$C_8 = \text{QuotientCompass}[p_0\ p_0\ p_H\ p_0\ p_G]$
$p_I \in C_8$

$p_6 \in \text{Hankinson}[p_0\ p_1\ p_2\ p_3\ p_4\ p_5\ p_6]$

The same method can be used to constrain points on transcendental curves, *e.g.* cycloids and spirals, or on curves that are generated by other curves, *e.g.* cissoids.

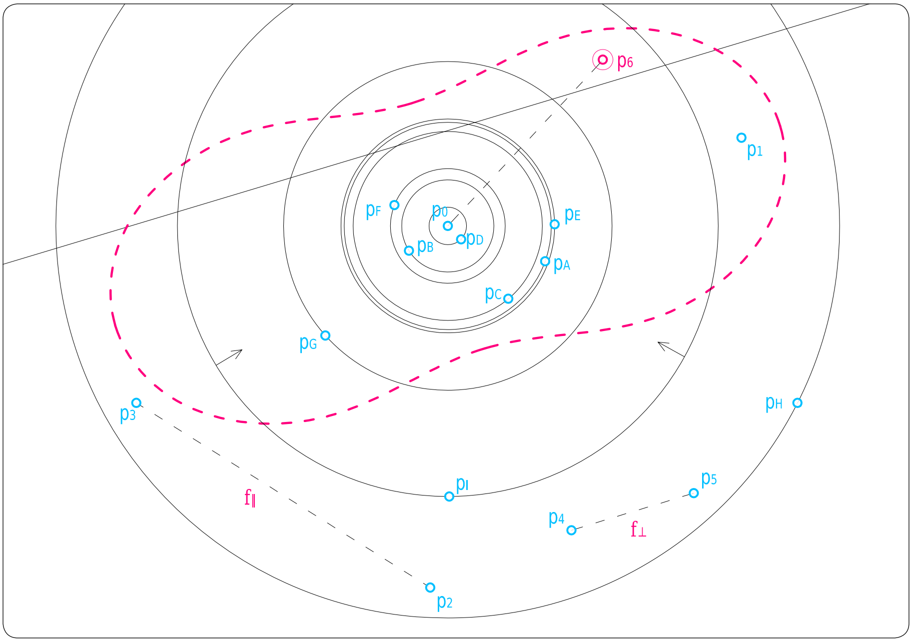
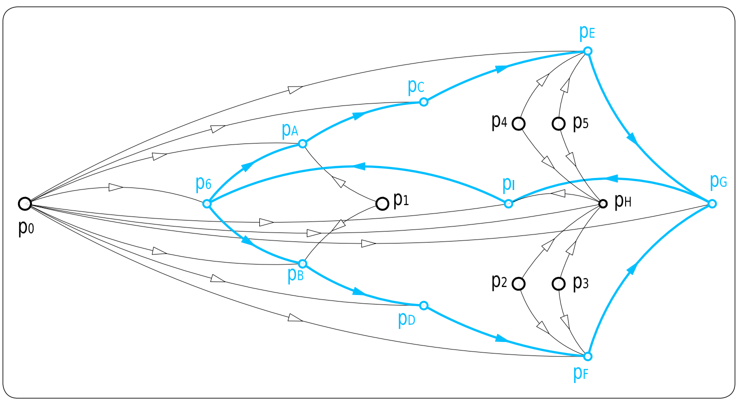


figure 223
construction of $p_6 \in$ Hankinson $[p_0\ p_1\ p_2\ p_3\ p_4\ p_5\ p_6]$ constraint and its graph of dependencies.



The last example solves the following system of inequations:

$r \leq f_1 f_2 / (f_1 \sin^2[\alpha] + f_2 \cos^2[\alpha]$
$r > F_3$
$\sin[\alpha] \geq (F_4 - \cos[\alpha]) / F_5$
$\cos[\alpha] \geq 0$

If r is the distance p_0p_2, if α is given by p_0p_1 and p_0p_2, and if the values $\{F_1, F_2, F_3, F_4, F_5\}$ are given by the distances $\{p_Ap_B, p_Cp_D, p_Ep_F, p_Gp_H, p_Ip_J\}$, then the system of inequations can be described by the following constraints:

$p_2 \in \mathrm{Hankinson}[p_0\ p_1\ p_A\ p_B\ p_C\ p_D\ p_2] \cap \backslash\mathrm{DI}[p_0\ p_E\ p_F]$

$\qquad \cap\ \mathrm{VHP}[p_4\ p_3] \cap \mathrm{VHP}[p_0\ p_4]$

$p_3 \in \mathrm{VSE}[p_0\ p_1] \cap \mathrm{HP}[p0\ p_5\ p_6] \cap \mathrm{C}[p_0\ p_G\ p_H]$

$p_4 \in \mathrm{QuotientCompass}[p_0\ p_G\ p_H\ p_I\ p_J] \cap \mathrm{SE}[p_0\ p_5\ p_6] \cap \mathrm{VHP}[p_1\ p_0]$

$p_5 \in \mathrm{CC}[p_0\ p_1] \cap \mathrm{CC}[p_1\ p_0] \cap \mathrm{VHP}[p_0\ p_1]$

$p_6 \in \mathrm{CC}[p_0\ p_1] \cap \mathrm{CC}[p_1\ p_0] \cap \mathrm{VHP}[p_1\ p_0]$

As a result, the set of positions that p_2 can hold is equivalent to the entire set of values that r and α can hold (figure 224).

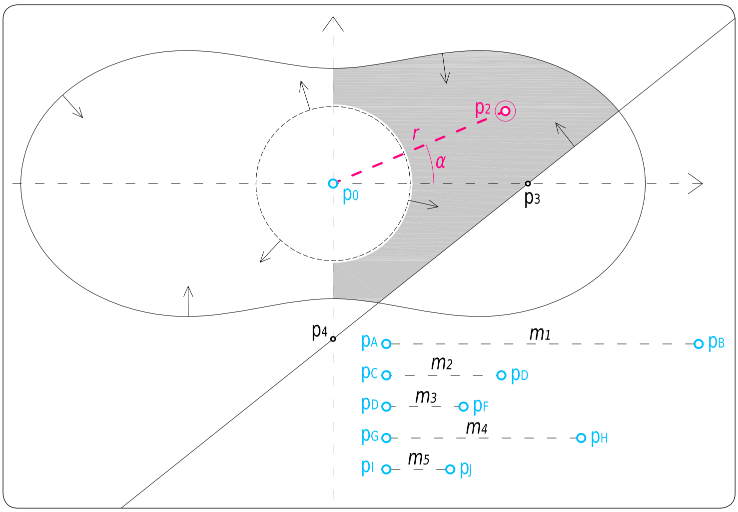


figure 224 construction of a system of inequations, the solution domain of p_2 is highlighted.

This last example also highlights the analogy that exists between the application of multiple geometric constraints and solving systems of non-linear inequations graphically. Only full implementation of constraint propagation — see sub-section 19 ("constraint propagations", page 201) — would guarantee that a solution always exists and would allow the solving of more complex systems of non-linear inequations, *e.g.* involving more than two variables or differentiations.

18 switching constraint dependencies

This subsection explains how to switch the hierarchy of dependencies between points — *i.e.* to make a child point its father's father and vice versa — without having to rebuild the entire construction. The consequence of this feature is that the user does not have to predict the parametric hierarchy in advance or rebuild the model whenever the parametric hierarchy has to be changed. If the user first analyses the behaviour of a certain result by varying the terms of the problem, this technique allows the user to alter this result explicitly and see how the terms of the problem would behave. The solution of a problem can become the statement of the inverse problem and vice versa.

using symmetry to switch dependencies · This technique takes advantage (1) of the fact that the only variables of the problem are positions of points in a plane and (2) of properties of symmetry in Proximity and Laterality relationships — see Ax.1 (page 62) and Ax.6 (page 64). As explained in sub-section 15 (“graphical regions and dynamic compliance with geometric relationships”, page 137), HalfPlane, DiscInside and DiscOutside constraints are directly linked to that symmetry. For instance, the following applications of constraints are all equivalent:

$$
\begin{aligned}
p_0 \in \text{HalfPlane}[p_1\ p_2\ p_3] &\longleftrightarrow p_1 \in \text{HalfPlane}[p_0\ p_3\ p_2] \\
&\longleftrightarrow p_2 \in \text{HalfPlane}[p_3\ p_1\ p_0] \\
&\longleftrightarrow p_3 \in \text{HalfPlane}[p_2\ p_0\ p_1]
\end{aligned}
$$

$$
\begin{aligned}
p_0 \in \text{DiscInside}[p_1\ p_2\ p_3] &\longleftrightarrow p_1 \in \text{DiscInside}[p_0\ p_2\ p_3] \\
&\longleftrightarrow p_2 \in \text{DiscOutside}[p_3\ p_0\ p_1] \\
&\longleftrightarrow p_3 \in \text{DiscOutside}[p_2\ p_0\ p_1]
\end{aligned}
$$

To replace a fundamental constraint with its symmetry means that the dependency between a child point and its father point are switched. Since any combined constraint is made of a symmetrical fundamental constraint, the dependencies of any pair of points that are relatives can consequently be switched by the domino effect of this symmetrical property.

As soon as two given points are relatives, the switch can be automated by an algorithm that (1) searches the set of fundamental constraints that link the dependencies between the two points and (2) replaces each of those constraints with its symmetrical counterpart.

example of switches · The following example shows how the process of this technique is made easier by analysing the graph of dependencies. The geometric construction of this example can be seen as the geometric skeleton of a funicular polyline passing through p_6 and bearing two forces that are applied on p_7 and p_8 and that have magnitudes equal to the distances p_1p_2 and p_2p_3 respectively. The initial dependencies are as follows (figure 225 and figure 226):

$p_7 \in$ Straightedge[p_4 p_1 p_2] ∩ Straightedge[p_6 p_0 p_1]
$p_8 \in$ Straightedge[p_5 p_2 p_3] ∩ Straightedge[p_7 p_0 p_2]
$p_9 \in$ Straightedge[p_8 p_0 p_3]

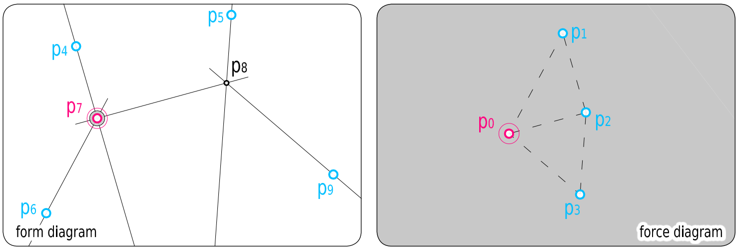


figure 225 prior to permutation; (left) the domain of p_7 is the shaded position; (right) the domain of p_0 is the shaded area.

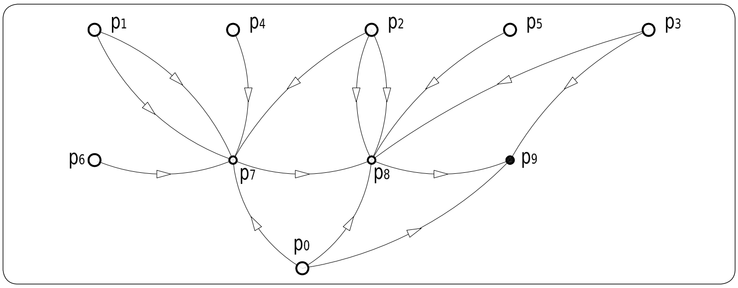


figure 226 directed graph of dependencies before permutation.

From this construction, it is deduced that the domain of p_7 is equal to a single position that is controlled by five points on the intersection of two secant straightedges. For this reason p_7 cannot be moved directly.

The first example of switching considers two direct relatives and consists in offering a degree of freedom to p_7. This would mean removing one of the two constraints applied to it and adding the symmetrical constraint to one of its parents. The set of its parents is {p_0 p_1 p_2 p_4 p_6}. All these points are good candidates since they can all still move within their domain. Point p_0 is chosen. It is a direct father of p_7 due to the Straightedge[p_6 p_0 p_1] constraint.

The Straightedge[p_6 p_0 p_1] constraint is actually the intersection of two fundamental constraints:

Straightedge[p_6 p_0 p_1] = HalfPlane[p_6 p_0 p_1] ∩ HalfPlane[p_6 p_1 p_0]

To apply this constraint on p_7 is therefore equivalent to the following symmetrical application:

p_0 ∈ HalfPlane[p_1 p_6 p_7] ∩ HalfPlane[p_1 p_7 p_6]
⟷ p_0 ∈ Straightedge[p_1 p_6 p_7]

After permutation, the resulting geometric construction is as follows (figure 227 and figure 228):

p_0 ∈ Straightedge[p_1 p_6 p_7]
p_7 ∈ Straightedge[p_4 p_1 p_2]
p_8 ∈ Straightedge[p_5 p_2 p_3] ∩ Straightedge[p_7 p_0 p_2]
p_9 ∈ Straightedge[p_8 p_0 p_3]

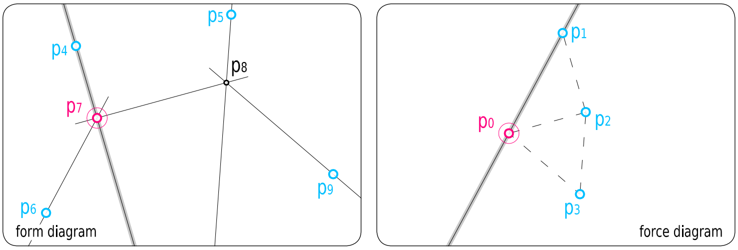


figure 227 after permutation; (left) the domain of p_7 is the shaded line; (right) the domain of p_0 is the shaded line.

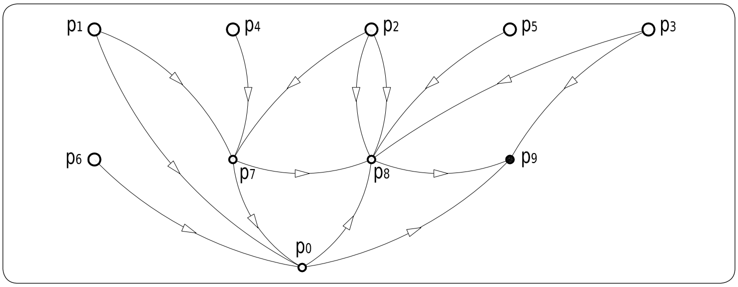


figure 228 directed graph of dependencies before permutation.

Point p_7 gained a degree of freedom and p_0 lost one of its degrees of freedom.

The second example of switching concerns the indirect relatives p_6 and p_9. On the initial construction (figure 225 and figure 226, page 196), point p_6 can be moved anywhere in the plane while point p_9 must stay on a Straightedge constraint:

$p_7 \in$ Straightedge[p_4 p_1 p_2] ∩ Straightedge[p_6 p_0 p_1]
$p_8 \in$ Straightedge[p_5 p_2 p_3] ∩ Straightedge[p_7 p_0 p_2]
$p_9 \in$ Straightedge[p_8 p_0 p_3]

It is decided to switch between these two degrees of freedom. Since p_9 is a great-grandson of p_6, three successive permutations have to be performed on the graph in figure 226, page 196: the first between p_9 and p_8, the second between p_8 and p_7 and the third between p_7 and p_6.

After the first switch of symmetrical applications, the construction is:

$p_7 \in$ Straightedge[p_4 p_1 p_2] ∩ Straightedge[p_6 p_0 p_1]
$p_8 \in$ Straightedge[p_5 p_2 p_3] ∩ Straightedge[p_7 p_0 p_2] ∩ Straightedge[p_9 p_0 p_3]

After the second switch of symmetrical applications, the construction is:

$p_7 \in$ Straightedge[p_4 p_1 p_2] ∩ Straightedge[p_6 p_0 p_1] ∩ Straightedge[p_8 p_0 p_2]
$p_8 \in$ Straightedge[p_5 p_2 p_3] ∩ Straightedge[p_9 p_0 p_3]

After the third switch of symmetrical applications, the final construction is (figure 229):

$p_6 \in$ Straightedge[p_7 p_0 p_1]
$p_7 \in$ Straightedge[p_4 p_1 p_2] ∩ Straightedge[p_8 p_0 p_2]
$p_8 \in$ Straightedge[p_5 p_2 p_3] ∩ Straightedge[p_9 p_0 p_3]

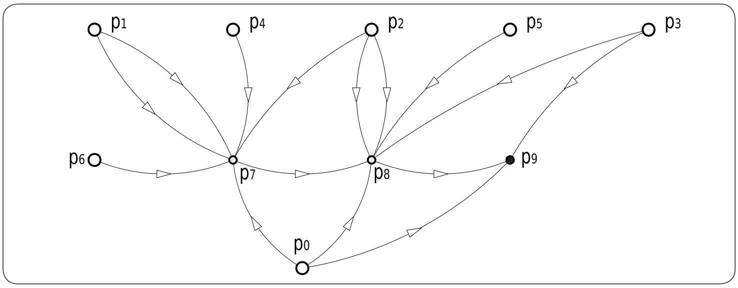


figure 229
directed graph of dependencies after three permutations.

Point p_9 is now the one that controls the geometry of the funicular polyline. It is important to note that all these intermediate operations are fully systematised. The user only needs to specify the two points that have to switch their degrees of freedom.

Furthermore, no computation of solutions is required by these permutations because the positions remain unchanged.

generation of interdependency by switching dependencies · Multiple genealogical paths sometimes exist between two relatives. If these paths are not all switched at the same time, a cycle of dependencies would occur and both points would become interdependent.

For example, if only one of the two dependences between p_1 and p_7 is chosen to be switched from the initial construction of figure 225 and figure 226, page 196, *e.g.* if only the first Straightedge constraint is replaced by its symmetrical counterpart —, the construction would become:

$p_1 \in$ Straightedge[p_2 p_4 p_7]
$p_7 \in$ Straightedge[p_6 p_0 p_1]
$p_8 \in$ Straightedge[p_5 p_2 p_3] ∩ Straightedge[p_7 p_0 p_2]
$p_9 \in$ Straightedge[p_8 p_0 p_3]

The corresponding directed graph of dependencies (figure 230) clearly shows that p_1 and p_7 are now interdependent. The interdependency is avoided if the application of $p_7 \in$ Straightedge[p_6 p_0 p_1] is also replaced by its symmetry.

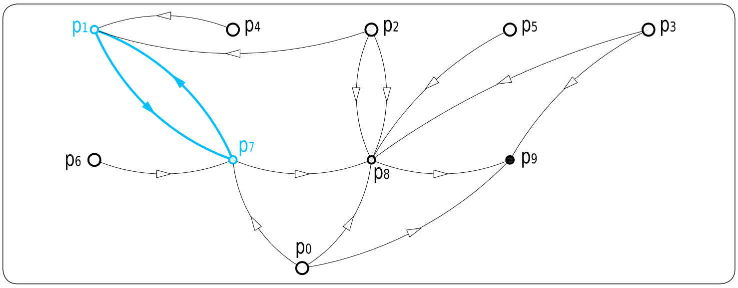


figure 230 directed graph of dependencies showing the emergence of interdependency.

For another example, the dependencies between p_0 and p_9 are chosen for switching from the initial construction of figure 225 and figure 226, page 196. Three genealogical paths link p_0 and p_9: (1) a direct path {$p_0 \rightarrow p_9$}; (2) an indirect path {$p_0 \rightarrow p_8 \rightarrow p_9$}; (3) another indirect path {$p_0 \rightarrow p_7 \rightarrow p_8 \rightarrow p_9$}. If these paths are not all switched at the same time, cycles of interdependencies would occur. The directed graph of dependencies resulting from the permutation of the third path only is shown in figure 231.

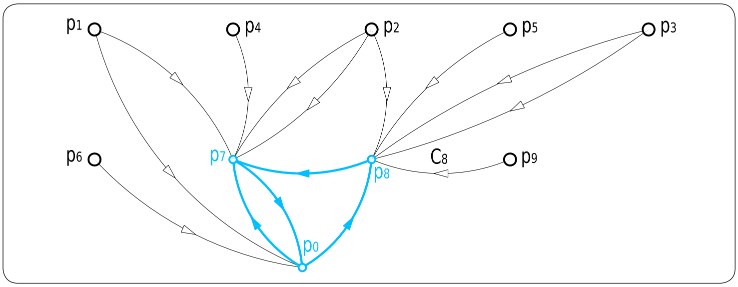


figure 231 directed graph of dependencies after permutation of the third path only.

automatic deletion of superfluous interdependencies · The beneficial consequence of this is that the switching of interdependencies can be automatised in order to delete all the superfluous interdependencies, *i.e.* all interdependent constraints that can be replaced by symmetrical non-interdependent equivalents, *e.g.* the interdependencies found in sub-section 17 ("examples of graphical computations", page 165) are not superfluous.

As a result, the user does not have to worry about whether or not the construction being carried out will induce superfluous interdependency. Algorithms can be performed afterwards in order to remove any superfluous interdependency.

switching the dependencies of veering and collapsible fundamental constraints · Switching the dependencies of two points that are linked by a VeeringHalfplane, a CollapsibleDiscOutside or a CollapsibleDiscInside constraint may either provoke interdependency or not. It depends on the chosen point of application of these constraints.

For instance, if $p_0 \in$ VeeringHalfplane[p_1 p_2], either $p_0 \in$ HalfPlane[p_1 p_1 p_2] or $p_0 \in$ HalfPlane[p_2 p_1 p_2] are equivalent constraints.

(1) If $p_0 \in$ HalfPlane[p_1 p_1 p_2], then the switch between p_0 and p_1 would produce either the application $p_1 \in$ HalfPlane[p_0 p_2 p_1] or the application $p_1 \in$ HalfPlane[p_2 p_1 p_0]; in both case, the switch creates the interdependency of p_1 by itself — see paragraph entitled "inner self-constraining" (page 162).

(2) If $p_0 \in$ HalfPlane[p_2 p_1 p_2], then the switch between p_0 and p_1 would produce the application $p_1 \in$ HalfPlane[p_2 p_2 p_0]. This does not create a new interdependency.

19 constraint propagations

the role of constraint propagation · Generally speaking, the role of constraint propagation is to restrict the number of values that variables of a given problem can hold. If the remaining set of values is a solution to the problem, this set is said to be consistent. A set of positions is consistent here if every point has a non-empty graphical region of solutions. Because of the constraint dependencies, this means that every father point must have a position that does not force the domain of its child points to be empty. Therefore, the role of constraint propagation is to restrict the domain of solutions of each father point so that it can never be placed in a position that empties the domain of one of its child points.

For instance, the following construction has to be made consistent (figure 165, page 152):

$p_0 \in \text{VeeringHalfplane}[p_1\ p_2] \cap \text{CollapsibleCompass}[p_1\ p_2]$
$p_6 \in \text{VeeringHalfplane}[p_1\ p_2] \cap \text{VeeringHalfplane}[p_4\ p_5] \cap \text{VeeringHalfplane}[p_3\ p_0]$

Because points p_1, p_2, p_3, p_4 and p_5 are orphans, their domain will never be made empty. However, points p_0 and p_6 may have empty domains. The role of constraint propagation is consequently to ensure (1) that p_1 and p_2 cannot be placed on positions that empty the domain of p_0 and (2) that p_0, p_1, p_2, p_3, p_4 and p_5 cannot be placed on positions that empty the domain of p_6.

Because of the nature of the constraints applied on p_0, it is actually impossible for p_1 and p_2 to empty the domain of p_0. There is consequently no domain of p_0 to propagate on p_1 or p_2. However, the constraints applied on p_6 might form an empty region for some particular positions of its father points, meaning that their domain must be restricted. For example, the domain of p_0 must be restricted so that there is always an intersection between the constraint VeeringHalfplane[p_3 p_0], *i.e.* the constraint that links p_0 with its child

p_6, and the constraint VeeringHalfplane[p_1 p_2] ∩ VeeringHalfplane[p_4 p_5], *i.e.* the other constraints applied on p_6. The propagation domain of p_0 is illustrated in figure 168, page 152.

The constraint that links the child point to the father point is called "the constraint that propagates", and the other constraint applied on the child is called "the constraint to be propagated".

Constraint propagation is not the only way to prevent empty domains. Since (1) geometric statements are expected to be constructed constraint by constraint, (2) points are expected to be dragged one by one, and (3) the initial set of positions is expected to be consistent, then any operation, either the application of a new constraint or the movement of a point, that would empty a domain can indeed be cancelled systematically. The role of constraint propagation is therefore mostly to display consistent domains, *i.e.* to ensure that all the positions inside a displayed domain are solutions of the geometric construction.

In the case of interdependent constructions, constraint propagation only makes sense if the child point and the father point are not identical. If they are, the point would actually always converge somewhere in its domain.

approaches to constraint propagation · Literature provides numerous generic methods to reduce the field of solutions to a problem (Bouma/Fudos/…·1995, Hoffmann/…·2005, Rossi/…·2006, Joan-Arinyo·2009, Mathis/Thierry·2010 and Bettig/…·2011). However, although some of them discuss parameters that can vary according to one degree of freedom (Hoffmann/Kim·2001, vanderMeiden/·2006 and Hidalgo/…·2012), no method exists as yet to handle parameters that have two degrees of freedom, *i.e.* to consider constraints that are, in the plane, as general as HalfPlane, DiscInside and DiscOutside.

Hopefully, this drawback is mitigated by the fact that there is only one type of variable here, *i.e.* only positions of points — there are no length or angle — and all the constraints are expressed by means of graphical regions, *i.e.* no external algebraic (in)equation. Initial attempts at constraint propagation might consequently be as follows:

(1) The first approach is mainly numerical and maybe the most straightforward. It would first generate a discretisation of all the potential positions that the father point could hold in the plane. It would then examine the resulting

domain of the child point for each position of the father point, perhaps through converging loops. It would finally display the entire finite set of positions (or a smooth approximation of it) that did not involve an empty domain of the child point. The main advantage of this approach is that a unique method fits all cases. One drawback is that the boundaries of the created domain are not accurate and small entities of this domain, *e.g.* isolated positions, may remain completely unknown. Another drawback is that the created domain has to be recalculated from scratch whenever a child point has moved, which might be slow and detrimental to the interactive handling of the geometric constructions.

(2) Another approach, also numerical but taking advantage of geometric reasoning, would first identify each mechanism produced by the constraints. This would then associate each mechanism with a known locus of positions already studied in literature, *e.g.* Artobolevski·1964. Together these loci form the boundaries of the propagation domain. The computation of well chosen positions would then determine which regions are inside the propagation domain and which regions are not. The advantage of this approach is that the locus can be parameterised and does not have to be recalculated from scratch whenever a child point has moved. The main drawback is that there is no general method: only known loci can be propagated.

(3) A third approach would construct the propagation domain symbolically, by means of Boolean combinations of HalfPlane, DiscInside and DiscOutside constraints. This approach presents two great advantages. Firstly, it uses a similar grammar to that already developed, meaning that any technique developed in the previous sub-sections can be directly applied to it: graphs of dependencies, automatic displacements to closest positions, permutations of constraints, *etc.* Secondly, propagation domains can be defined by combinations of constraints in such a way that their geometrical behaviour, *i.e.* the resulting region, remains valid for any position. Therefore they do not have to be recalculated at each move. Their update is only required when new constraints are applied or old ones deleted. However, the benefits of this approach also reduce its scope of application: it only handles domains that can be described by HalfPlane, DiscInside and DiscOutside constraints. As a consequence, it is restricted to propagations where the path of paternal filiation between the child point and the father point is unique, *e.g.* if a point is the father of another point, it can be its father several times (by several constraints) but it cannot be its grandfather or its child.

For instance, this last approach cannot perform the propagation domain of p_0 within the following construction (figure 232):

$p_{10} \in \text{Straightedge}[p_3\ p_0\ p_1] \cap \text{Straightedge}[p_4\ p_4\ p_5]$
$p_{11} \in \text{Straightedge}[p_{10}\ p_0\ p_2] \cap \text{Straightedge}[p_6\ p_6\ p_7]$
$\qquad \cap\ \text{HalfPlane}[p_6\ p_8\ p_9] \cap \text{HalfPlane}[p_7\ p_9\ p_8]$
$p_8\ \in \text{Compass}[p_6\ p_6\ p_7] \cap \text{Compass}[p_7\ p_7\ p_6] \cap \text{HalfPlane}[p_6\ p_6\ p_7]$
$p_9\ \in \text{Compass}[p_6\ p_6\ p_7] \cap \text{Compass}[p_7\ p_7\ p_6] \cap \text{HalfPlane}[p_7\ p_7\ p_6]$

The shaded area in figure 232 is the propagation domain that p_0 must hold in order to ensure that the domain of p_{11} is not empty. Since p_{11} is its child and its grandchild at the same time, two paths of paternal filiation exist and the propagation domain of p_0 cannot be expressed in terms of fundamental constraints.

This field of study still requires further research. The best approach would probably be a mix of those mentioned above.

By way of introduction, the paragraphs that follow develop a general procedure and a sub-procedure of it for the symbolic propagation of particular dependencies (third approach).

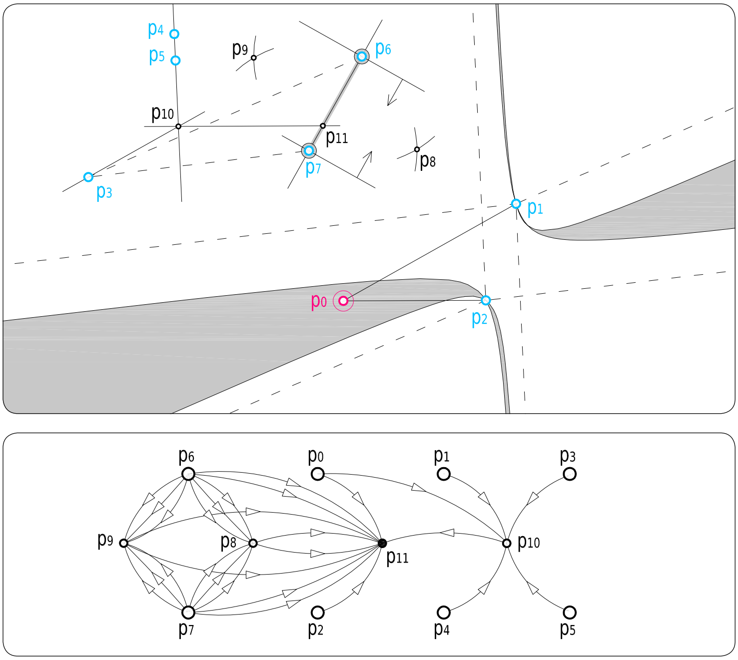


figure 232
point p_{11} will remain on the line segment p_6p_7 as long as p_0 moves inside the grey area; (below) corresponding directed graph of dependencies.

general method for symbolic propagation · Graphs of dependencies are usually more complex than what has been explained for figure 165, page 152. The point whose propagation domain is sought might have many children that are not direct relatives — p_0 in figure 233 —, meaning that domains must be propagated from child to father incrementally.

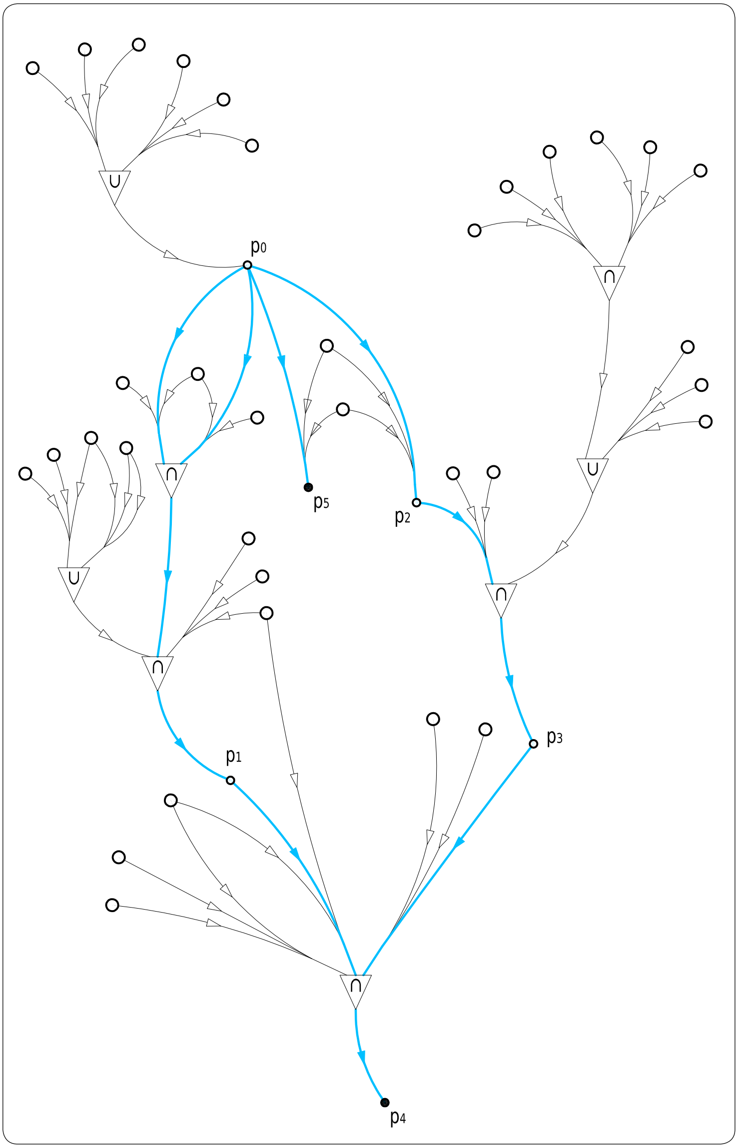


figure 233
a fictitious directed graph of dependencies.

Since the domain of a child cannot be propagated if it does not reflect the domain propagation of its own children, the following general procedure should be used to obtain the propagation domain of a point p_0:

(1) create P_{child} = the set of all child points and grandchild points of p_0, plus p_0 itself

(2) rewrite the constraints applied on each point of P_{child} in terms of fundamental constraints.

(3) for each point p* of P_{child} (p_0 excluded) and whose domain has not yet been propagated onto its parents that are also in P_{child}:

- (3.1) if p* is childless, its "propagation domain" is equal to the entire plane and is considered "completed".
- (3.2) if the constraints applied on all the children and grandchildren of p* have been propagated on it, the propagation domain of p* is "completed"
- (3.3) when the propagation domain of p* becomes completed, its "do main of solutions" can be propagated on all its parents that also belong to P_{child}, like this:
  - (3.3.1) get the "domain of solutions" of p*, *i.e.* the intersection of its "propagation domain" and its "input domain"
  - (3.3.2) get the pure equivalent union of the domain of solutions of p* — see paragraph entitled "pure equivalent unions and intersections" (page 139)
  - (3.3.3) get all the parent points of p* that also belong to P_{child}
  - (3.3.4) for every parent of p* that also belongs to P_{child}:
    - (3.3.4.1) for each sub-domain of the pure equivalent union of the domain of solutions of p*:
      - (3.3.4.1.1) get the constraints that propagate and the constraint(s) to be propagated
      - (3.3.4.1.2) split up the sub-domain into as many intersections as propagating constraints, so that each new intersection only includes one propagating constraint
      - (3.3.4.1.3) for each new intersection:
        - (3.3.4.1.3.1) given the nature of the constraint that propagates and the constraints to be propagated, identify the pattern of propagation to be applied — see next paragraphs

(3.3.4.1.3.2) if a pattern exists, construct the local propagation domain associated with this intersection

(3.3.4.1.4) get the intersection of all the generated propagation domains

(3.3.4.2) get the union of the propagation of each sub-domain

(3.3.4.3) apply this union onto the uncompleted propagation domain of the current parent of p_*

(4) Once the "propagation domain" of p_0 is completed, get its "domain of solutions" by intersecting its "propagation domain" with its "input domain".

As stated above, this procedure assumes that there is only one way of paternal filiation between the child point and the father point. The following example meets this condition.

On the other hand, the quality of the propagation procedure depends on the chronology in which the domains of the child points are propagated. In the worst case scenario, it is probable that the resulting propagation domain or that the resulting domain of solutions becomes empty after this procedure, meaning that there is no position on which a point can go without emptying the domain of one of its children. It is also likely that the domain of a child point remains empty after the propagation of its domain onto its father points. However, this does not mean that there are no sets of positions for which no domain is empty. The solution is to wait for the domains of other child points to be propagated before updating the propagation domain that cause the problem.

Given a new geometric construction, the goal of the following example is to obtain the propagation domain of p_0 (figure 234) using the aforementioned procedure. The construction is as follows:

$p_5 \in \text{CollapsibleCompass}[p_1\ p_2] \cap \text{CollapsibleCompass}[p_2\ p_1] \cap \text{VeeringHalfPlane}[p_1\ p_2]$

$p_6 \in \text{CollapsibleCompass}[p_1\ p_2] \cap \text{CollapsibleCompass}[p_2\ p_1] \cap \text{VeeringHalfPlane}[p_2\ p_1]$

$p_7 \in \text{Straightedge}[p_0\ p_1\ p_2] \cap \text{CollapsibleCompass}[p_3\ p_4] \cap \text{HalfPlane}[p_3\ p_6\ p_5]$

$p_{13} \in (\ (\text{VeeringHalfPlane}[p_{10}\ p_{11}] \cap \text{VeeringHalfPlane}[p_9\ p_8]) \cup \text{Compass}[p_{12}\ p_1\ p_2]\ ) \cap \text{VeeringHalfPlane}[p_3\ p_7]$

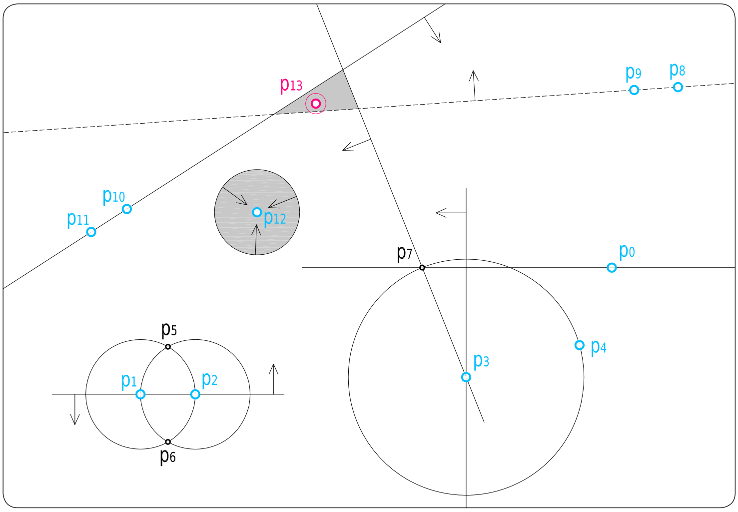


figure 234
the solution domain of p_{13}.

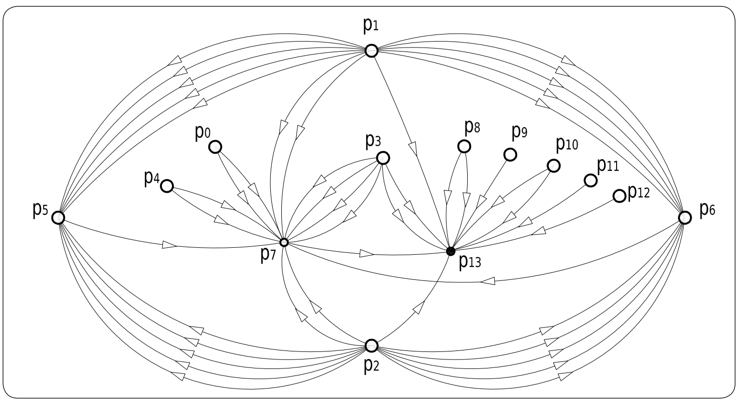


figure 235
directed graph of dependencies for the construction of figure 234.

(1) On the basis of the directed graph of dependencies (figure 235), the P_{child} set of p_0 is equivalent to $\{p_0\ p_7\ p_{13}\}$.

(2) Their domain expressed in terms of fundamental constraints is:

$p_5 \in C_2 \cap C_3 \cap C_4 \cap C_5 \cap C_6$
$p_7 \in C_0 \cap C_1 \cap C_8 \cap C_9 \cap C_{10}$
$p_{13} \in (\ (C_{11} \cap C_{12}) \cup C_{13}\ ) \cap C_{14}$

where: C_0 = HalfPlane$[p_0\ p_1\ p_2]$
C_1 = HalfPlane$[p_0\ p_2\ p_1]$

C_2 = InsideDisc[p_1 p_1 p_2]
C_3 = OutsideDisc[p_1 p_1 p_2]
C_4 = InsideDisc[p_2 p_1 p_2]
C_5 = OutsideDisc[p_2 p_1 p_2]
C_6 = HalfPlane[p_1 p_1 p_2]
C_7 = HalfPlane[p_1 p_2 p_1]

C_8 = InsideDisc[p_3 p_3 p_4]
C_9 = OutsideDisc[p_3 p_3 p_4]
C_{10} = HalfPlane[p_3 p_6 p_5]

C_{11} = HalfPlane[p_{10} p_{10} p_{11}]
C_{12} = \HalfPlane[p_8 p_8 p_9]
C_{13} = InsideDisc[p_{12} p_1 p_2]
C_{14} = HalfPlane[p_3 p_3 p_7]

(3) The next instructions consider each point of P_{child} (p_0 excluded) individually:

a p_{13}

b p_7

a (3.1) Since p_{13} is childless, its domain of solutions is equal to the entire plane Ω.

a (3.3) Since the domain of solutions of p_{13} is completed, it can consequently be propagated on all its parents that are also in P_{child} — *i.e.* point p_7.

a (3.3.1) The domain of solutions of p_{13} is therefore computed as follows:

$$\begin{aligned} \text{SolutionDom}[p_{13}] &= \text{InputDom}[p_{13}] \cap \text{PropagationDom}[p_{13}] \\ &= \text{InputDom}[p_{13}] \cap \Omega \\ &= (\ (C_{11} \cap C_{12}) \cup C_{13}\ ) \cap C_{14} \end{aligned}$$

a (3.3.2) The pure equivalent union of this domain is:

$$\text{PureEquivalentUnion of SolutionDom}[p_{13}] = (C_{11} \cap C_{12} \cap C_{14}) \cup (C_{13} \cap C_{14})$$

a (3.3.3) The only parent point of p_{13} that is also in P_{child} is p_7.

a (3.3.4.1) The various sub-domains of the pure equivalent union of the domain of solutions of p_{13} are:

a.a $(C_{11} \cap C_{12} \cap C_{14})$

a.b $(C_{13} \cap C_{14})$

a.a (3.3.4.1.1) The constraint that propagates the first sub-domain of p_{13} onto p_7 is C_{14} since p_7 is a parameter of C_{14}.

a.a (3.3.4.1.2) There is only one constraint that propagates the first sub-domain of p_{13}.

a.a (3.3.4.1.3.1) The pattern of propagation is the one that will be presented in next paragraph entitled "first symbolic pattern: propagation of convex intersections" (page 217).

a.a (3.3.4.1.3.2) The result of this local propagation is (figure 236, left):

$$(\text{Dom}[p_{13} \in C_{11} \cap C_{12} \cap C_{14}]\ \textit{propagated onto}\ p_7\ \textit{by}\ C_{14}) = C_{15} \cup C_{16}$$

where : $C_{15} = \backslash\text{HalfPlane}[p_3\ p_3\ p_A]$

$C_{16} = \backslash\text{Halfplane}[p_3\ p_{11}\ p_{10}]$

with: $p_A = \text{VeeringStraightedge}[p_{10}\ p_{11}] \cap \text{VeeringStraightedge}[p_8\ p_9]$

a.b (3.3.4.1.1) The constraint that propagates the second sub-domain of p_{13} onto p_7 is also C_{14} since p_7 is a parameter of C_{14}.

a.b (3.3.4.1.2) There is only one constraint that propagates the second sub-domain of p_{13}.

a.b (3.3.4.1.3.1) The pattern of propagation is the one that will be presented in next paragraph entitled "first symbolic pattern: propagation of convex intersections" (page 217).

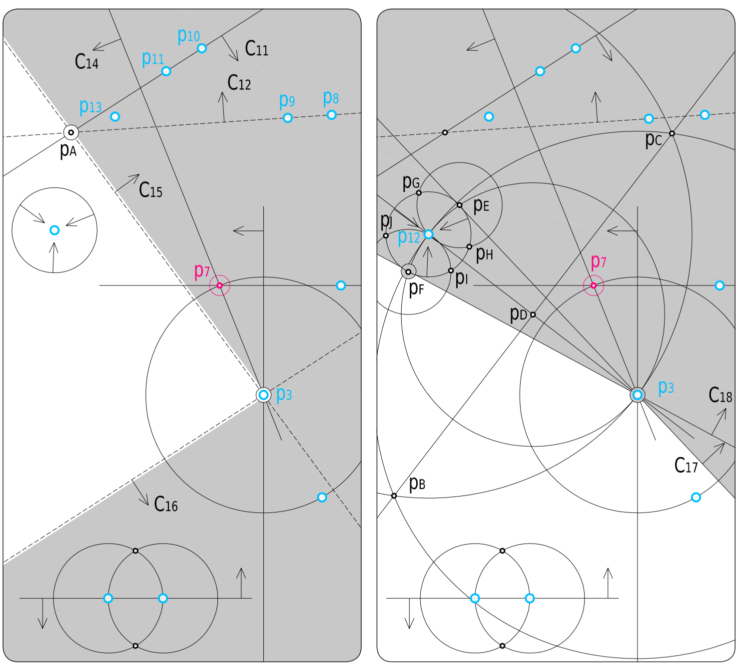


figure 236 (left) first propagation of the sub-domain of p_{13} onto p_7; (right) second propagation of the sub-domain of p_{13} onto p_7.

a.b (3.3.4.1.3.2) The result of this local propagation is (figure 236, right):

$(Dom[p_{13} \in C_{11} \cap C_{12} \cap C_{14}]$ *propagated onto* p_7 *by* $C_{14}) = C_{17} \cup C_{18}$

where : $C_{17} = HalfPlane[p_E\ p_G\ p_H]$
$C_{18} = HalfPlane[p_F\ p_J\ p_I]$

with: $p_B = Compass[p_3\ p_3\ p_{12}] \cap Compass[p_{12}\ p_{12}\ p_3] \cap HalfPlane[p_3\ p_3\ p_{12}]$
$p_C = Compass[p_3\ p_3\ p_{12}] \cap Compass[p_{12}\ p_{12}\ p_3] \cap HalfPlane[p_{12}\ p_{12}\ p_3]$
$p_D = Straightedge[p_3\ p_3\ p_{12}] \cap Straightedge[p_B\ p_B\ p_C]$
$p_E = Compass[p_{12}\ p_1\ p_2] \cap Compass[p_D\ p_D\ p_3] \cap HalfPlane[p_{12}\ p_{12}\ p_3]$
$p_F = Compass[p_{12}\ p_1\ p_2] \cap Compass[p_D\ p_D\ p_3] \cap HalfPlane[p_3\ p_3\ p_{12}]$
$p_G = Compass[p_{12}\ p_{12}\ p_E] \cap Compass[p_E\ p_E\ p_{12}] \cap HalfPlane[p_{12}\ p_{12}\ p_E]$
$p_H = Compass[p_{12}\ p_{12}\ p_E] \cap Compass[p_E\ p_E\ p_{12}] \cap HalfPlane[p_E\ p_E\ p_{12}]$
$pI = Compass[p_{12}\ p_{12}\ p_F] \cap Compass[p_F\ p_F\ p_{12}] \cap HalfPlane[p_{12}\ p_{12}\ p_F]$
$pJ = Compass[p_{12}\ p_{12}\ p_F] \cap Compass[p_F\ p_F\ p_{12}] \cap HalfPlane[p_F\ p_F\ p_{12}]$

a (3.3.4.2) $PropagationDom[p_7] = (C_{15} \cup C_{16}) \cup (C_{17} \cup C_{18})$.

b (3.2) The propagation domain of p_7 is now completed.

b (3.3) Since the domain of solutions of p_7 is completed, it can consequently be propagated on all its parents that also belong to P_{child}, *i.e.* point p_0.

b (3.3.1) The domain of solutions of p_7 is computed as follows:

$$\begin{aligned} SolutionDom[p_7] &= InputDom[p_7] \cap PropagationDom[p_7] \\ &= InputDom[p_7] \cap ((C_{15} \cup C_{16}) \cup (C_{17} \cup C_{18})) \\ &= C_0 \cap C_1 \cap C_8 \cap C_9 \cap C_{10} \cap (C_{15} \cup C_{16} \cup C_{17} \cup C_{18}) \end{aligned}$$

b (3.3.2) The pure equivalent union of this domain is:

$$\begin{aligned} PureEquivalentUnion\ of\ SolutionDom[p_7] = &(C_0 \cap C_1 \cap C_8 \cap C_9 \cap C_{10} \cap C_{15}) \\ &\cup (C_0 \cap C_1 \cap C_8 \cap C_9 \cap C_{10} \cap C_{16}) \\ &\cup (C_0 \cap C_1 \cap C_8 \cap C_9 \cap C_{10} \cap C_{17}) \\ &\cup (C_0 \cap C_1 \cap C_8 \cap C_9 \cap C_{10} \cap C_{18}) \end{aligned}$$

b (3.3.3) The only parent point of p_7 that also belongs to P_{child} is p_0.

b (3.3.4.1) The various sub-domains of the pure equivalent union of the domain of solutions of p_{13} are:

b.a $(C_0 \cap C_1 \cap C_8 \cap C_9 \cap C_{10} \cap C_{15})$

b.b $(C_0 \cap C_1 \cap C_8 \cap C_9 \cap C_{10} \cap C_{16})$

b.c $(C_0 \cap C_1 \cap C_8 \cap C_9 \cap C_{10} \cap C_{16})$

b.d $(C_0 \cap C_1 \cap C_8 \cap C_9 \cap C_{10} \cap C_{16})$

b.a, b.b, b.c and b.d (3.3.4.1.1) The constraints that propagate each sub-domain of p_7 onto p_0 are always C_0 and C_1 — *i.e.* p_0 is a parameter of C_0 and C_1.

b.a (3.3.4.1.2) The propagations concerning the first sub-domain are as follows:

b.a.a $(C_0 \cap C_8 \cap C_9 \cap C_{10} \cap C_{15})$

b.a.b $(C_1 \cap C_8 \cap C_9 \cap C_{10} \cap C_{15})$

b.b (3.3.4.1.2) The propagations concerning the second sub-domain are as follows:

b.b.a $(C_0 \cap C_8 \cap C_9 \cap C_{10} \cap C_{16})$

b.b.b $(C_1 \cap C_8 \cap C_9 \cap C_{10} \cap C_{16})$

b.c (3.3.4.1.2) The propagations concerning the third sub-domain are as follows:

b.c.a $(C_0 \cap C_8 \cap C_9 \cap C_{10} \cap C_{17})$

b.c.b $(C_1 \cap C_8 \cap C_9 \cap C_{10} \cap C_{17})$

b.d (3.3.4.1.2) The propagations concerning the fourth sub-domain are as follows:

b.d.a $(C_0 \cap C_8 \cap C_9 \cap C_{10} \cap C_{18})$

b.d.b $(C_1 \cap C_8 \cap C_9 \cap C_{10} \cap C_{18})$

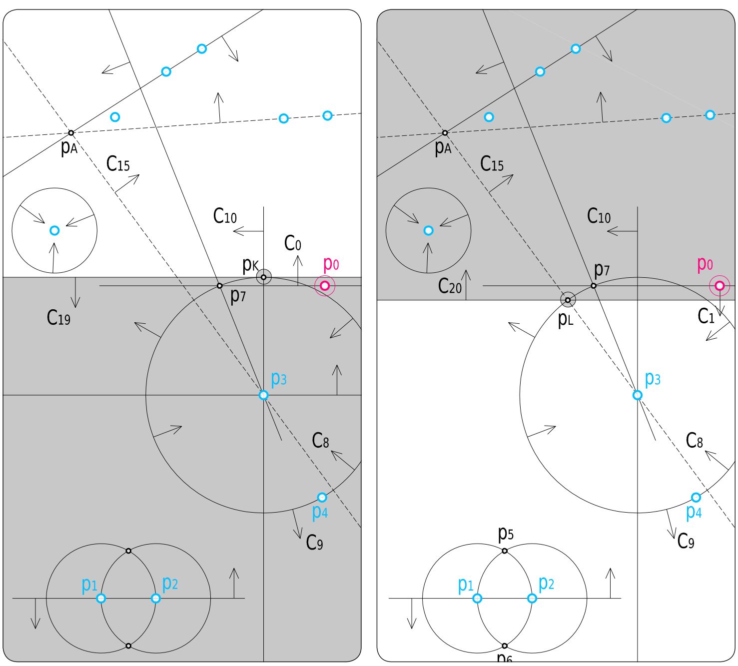


figure 237 (left) the propagation constraint C_{19}; (right) the propagation constraint C_{20}.

b.a, b.b, b.c and b.d (3.3.4.1.3.1) The pattern of propagation is in all cases the one that will be presented in next paragraph entitled "first symbolic pattern: propagation of convex intersections" (page 217).

b.a.a (3.3.4.1.3.2) The result of the first local propagation is (figure 237, left):

$(\text{Dom}[p_7 \in C_0 \cap C_8 \cap C_9 \cap C_{10} \cap C_{15}]$ *propagated onto* p_0 *by* $C_0) = C_{19}$

where : $C_{19} = \text{Halfplane}[p_K\ p_2\ p_1]$

with: $p_K = \text{Compass}[p_3\ p_3\ p_4] \cap \text{Straightedge}[p_3\ p_5\ p_6] \cap \text{Halfplane}[p_3\ p_1\ p_2]$

b.a.b (3.3.4.1.3.2) The result of the second local propagation is (figure 237, right):

$(\text{Dom}[p_7 \in C_1 \cap C_8 \cap C_9 \cap C_{10} \cap C_{15}]$ *propagated onto* p_0 *by* $C_1) = C_{20}$

where : $C_{20} = \backslash\text{Halfplane}[p_L\ p_2\ p_1]$

with: $p_L = \text{Compass}[p_3\ p_3\ p_4] \cap \text{Straightedge}[p_3\ p_3\ p_A] \cap \text{Halfplane}[p_3\ p_6\ p_5]$

b.b.a (3.3.4.1.3.2) The result of the third local propagation is (figure 238, left):

$(\text{Dom}[p_7 \in C_0 \cap C_8 \cap C_9 \cap C_{10} \cap C_{16}]$ *propagated onto* p_0 *by* $C_0) = C_{21}$

where : $C_{21} = \text{Halfplane}[p_K\ p_2\ p_1]$

with: $p_M = \text{Compass}[p_3\ p_3\ p_4] \cap \text{Straightedge}[p_3\ p_{10}\ p_{11}] \cap \text{Hp}[p_3\ p_6\ p_6]$

b.b.b (3.3.4.1.3.2) The result of the fourth local propagation is (figure 238, right):

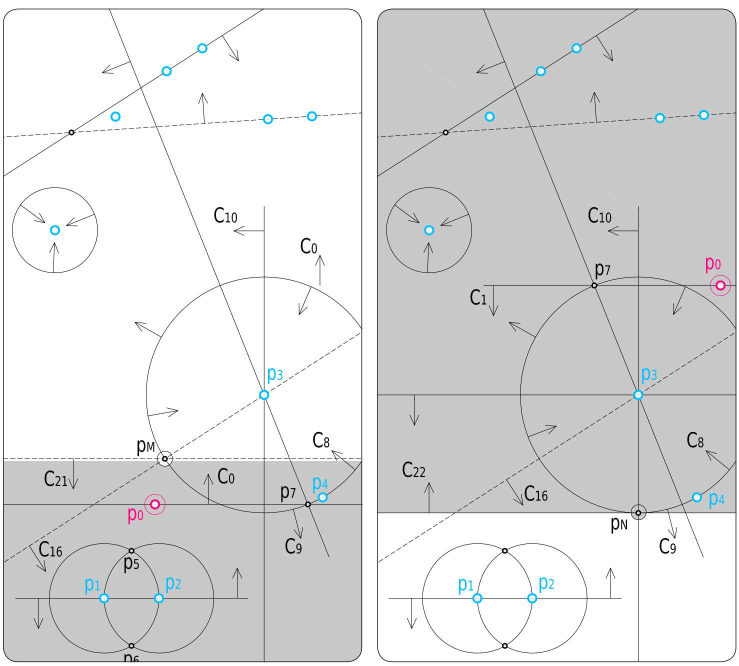


figure 238 (left) the propagation constraint C_{21}; (right) the propagation constraint C_{22}.

$(\mathrm{Dom}[p_7 \in C_1 \cap C_8 \cap C_9 \cap C_{10} \cap C_{16}]$ *propagated onto* p_0 *by* $C_1\,) = C_{22}$

where : $C_{22} = \mathrm{Halfplane}[p_K\ p_2\ p_1]$

with: $p_N = \mathrm{Compass}[p_3\ p_3\ p_4] \cap \mathrm{Straightedge}[p_3\ p_5\ p_6] \cap \mathrm{Hp}[p_3\ p_2\ p_1]$

b.c.a (3.3.4.1.3.2) The result of the fifth local propagation is (figure 237, left):

$(\mathrm{Dom}[p_7 \in C_0 \cap C_8 \cap C_9 \cap C_{10} \cap C_{17}]$ *propagated onto* p_0 *by* $C_0\,) = C_{19}$

b.c.b (3.3.4.1.3.2) The result of the sixth local propagation is (figure 239, left):

$(\mathrm{Dom}[p_7 \in C_1 \cap C_8 \cap C_9 \cap C_{10} \cap C_{17}]$ *propagated onto* p_0 *by* $C_1\,) = C_{23}$

where : $C_{23} = \mathrm{Halfplane}[p_K\ p_2\ p_1]$

with: $p_Q = \mathrm{Compass}[p_3\ p_3\ p_4] \cap \mathrm{Straightedge}[p_E\ p_G\ p_H] \cap \mathrm{Hp}[p_3\ p_6\ p_5]$

b.d.a (3.3.4.1.3.2) The result of the seventh local propagation is (figure 237, left):

$(\mathrm{Dom}[p_7 \in C_0 \cap C_8 \cap C_9 \cap C_{10} \cap C_{18}]$ *propagated onto* p_0 *by* $C_0\,) = C_{19}$

b.d.b (3.3.4.1.3.2) The result of the eighth local propagation is (figure 239, right):

$(\mathrm{Dom}[p_7 \in C_1 \cap C_8 \cap C_9 \cap C_{10} \cap C_{18}]$ *propagated onto* p_0 *by* $C_1\,) = C_{24}$

where : $C_{24} = \mathrm{Halfplane}[p_R\ p_1\ p_2]$

with: $p_R = \mathrm{Compass}[p_3\ p_3\ p_4] \cap \mathrm{Straightedge}[p_F\ p_J\ p_I] \cap \mathrm{Hp}[p_3\ p_6\ p_5]$

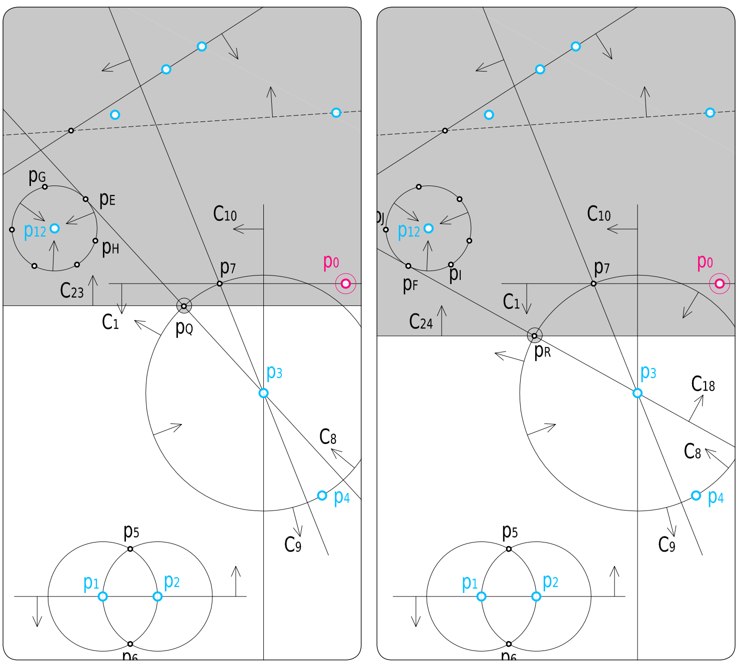


figure 239 (left) the propagation constraint C_{23}; (right) the propagation constraint C_{24}.

b.a (3.3.4.1.4) The intersection resulting from the propagation of the first subdomain is:

$$(\text{Dom}[p_7 \in C_0 \cap C_1 \cap C_8 \cap C_9 \cap C_{10} \cap C_{15}] \textit{ propagated onto } p_0) = C_{19} \cap C_{20}$$

b.b (3.3.4.1.4) The intersection resulting from the propagation of the second subdomain is:

$$(\text{Dom}[p_7 \in C_0 \cap C_1 \cap C_8 \cap C_9 \cap C_{10} \cap C_{16}] \textit{ propagated onto } p_0) = C_{21} \cap C_{22}$$

b.c (3.3.4.1.4) The intersection resulting from the propagation of the third subdomain is:

$$(\text{Dom}[p_7 \in C_0 \cap C_1 \cap C_8 \cap C_9 \cap C_{10} \cap C_{17}] \textit{ propagated onto } p_0) = C_{19} \cap C_{23}$$

b.d (3.3.4.1.4) The intersection resulting from the propagation of the fourth subdomain is:

$$(\text{Dom}[p_7 \in C_0 \cap C_1 \cap C_8 \cap C_9 \cap C_{10} \cap C_{18}] \textit{ propagated onto } p_0) = C_{19} \cap C_{24}$$

b (3.3.4.2) The union of all these intersections is:

$$\begin{aligned}(\text{Dom}[p_7] \textit{ propagated onto } p_0) = {} & (C_{19} \cap C_{20}) \\ & \cup (C_{21} \cap C_{22}) \\ & \cup (C_{19} \cap C_{23}) \\ & \cup (C_{19} \cap C_{24})\end{aligned}$$

(4) The propagation domain of p_0 is now completed since p_7 was its only child. Its domain of solution is the following one:

$$\begin{aligned}\text{SolutionDomain}[p_0] &= \text{InputDomain}[p_0] \cap \text{PropagationDomain}[p_0] \\ &= \Omega \cap (C_{19} \cap C_{20}) \cup (C_{21} \cap C_{22}) \cup (C_{19} \cap C_{23}) \cup (C_{19} \cap C_{24})\end{aligned}$$

where:

$C_{19} = \text{HalfPlane}[p_K\ p_2\ p_1]$
$C_{20} = \backslash\text{HalfPlane}[p_L\ p_2\ p_1]$
$C_{21} = \backslash\text{HalfPlane}[p_M\ p_1\ p_2]$
$C_{22} = \text{HalfPlane}[p_N\ p_1\ p_2]$
$C_{23} = \text{HalfPlane}[p_Q\ p_1\ p_2]$
$C_{24} = \text{HalfPlane}[p_R\ p_1\ p_2])$

with:

$p_A = \text{Straightedge}[p_{10}\ p_{10}\ p_{11}] \cap \text{Straightedge}[p_8\ p_8\ p_9]$
$p_B = \text{Compass}[p_3\ p_3\ p_{12}] \cap \text{Compass}[p_{12}\ p_{12}\ p_3] \cap \text{HalfPlane}[p_3\ p_3\ p_{12}]$
$p_C = \text{Compass}[p_3\ p_3\ p_{12}] \cap \text{Compass}[p_{12}\ p_{12}\ p_3] \cap \text{HalfPlane}[p_{12}\ p_{12}\ p_3]$
$p_D = \text{Straightedge}[p_3\ p_3\ p_{12}] \cap \text{Straightedge}[p_B\ p_B\ p_C]$
$p_E = \text{Compass}[p_{12}\ p_1\ p_2] \cap \text{Compass}[p_D\ p_D\ p_3] \cap \text{HalfPlane}[p_{12}\ p_{12}\ p_3]$
$p_F = \text{Compass}[p_{12}\ p_1\ p_2] \cap \text{Compass}[p_D\ p_D\ p_3] \cap \text{HalfPlane}[p_3\ p_3\ p_{12}]$
$p_G = \text{Compass}[p_{12}\ p_{12}\ p_E] \cap \text{Compass}[p_E\ p_E\ p_{12}] \cap \text{HalfPlane}[p_{12}\ p_{12}\ p_E]$
$p_H = \text{Compass}[p_{12}\ p_{12}\ p_E] \cap \text{Compass}[p_E\ p_E\ p_{12}] \cap \text{HalfPlane}[p_E\ p_E\ p_{12}]$

p_I = Compass[p_{12} p_{12} p_F] ∩ Compass[p_F p_F p_{12}] ∩ HalfPlane[p_{12} p_{12} p_F]
p_J = Compass[p_{12} p_{12} p_F] ∩ Compass[p_F p_F p_{12}] ∩ HalfPlane[p_F p_F p_{12}]
p_K = Compass[p_3 p_3 p_4] ∩ Straightedge[p_3 p_5 p_6] ∩ Halfplane[p_3 p_1 p_2]
p_L = Compass[p_3 p_3 p_4] ∩ Straightedge[p_3 p_3 p_A] ∩ Halfplane[p_3 p_6 p_5]
p_M = Compass[p_3 p_3 p_4] ∩ Straightedge[p_3 p_{10} p_{11}] ∩ Hp[p_3 p_6 p_6]
p_N = Compass[p_3 p_3 p_4] ∩ Straightedge[p_3 p_5 p_6] ∩ Hp[p_3 p_2 p_1]
p_Q = Compass[p_3 p_3 p_4] ∩ Straightedge[p_E p_G p_H] ∩ Hp[p_3 p_6 p_5]
p_R = Compass[p_3 p_3 p_4] ∩ Straightedge[p_F p_J p_I] ∩ Hp[p_3 p_6 p_5]

This domain is illustrated in figure 240. As this will be explained in the next paragraph entitled “first symbolic pattern: propagation of convex intersections” (page 217), this solution domain remains valid for any position of p_0, p_1, ..., p_{13}. If one of these points moves, the domain of solution of p_0 remains described by (C_{19} ∩ C_{20}) ∪ (C_{21} ∩ C_{22}) ∪ (C_{19} ∩ C_{23}) ∪ (C_{19} ∩ C_{24}) but will have a different geometry.

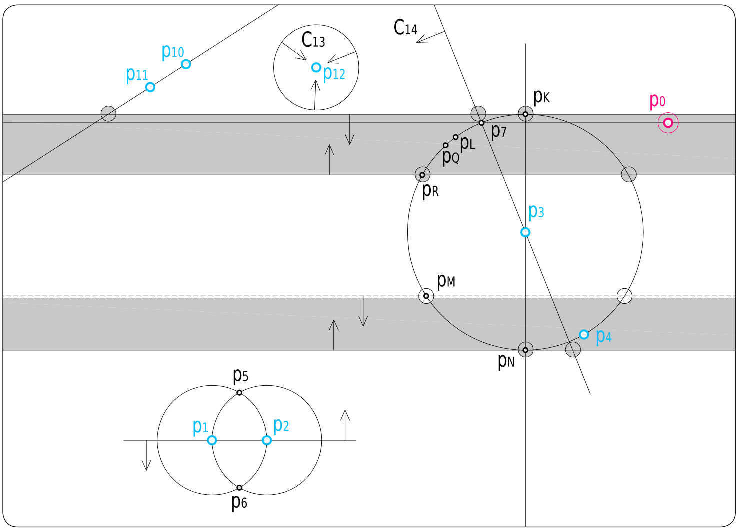


figure 240
the solution domain of p_0 (so that the domain of p_{13} is never empty).

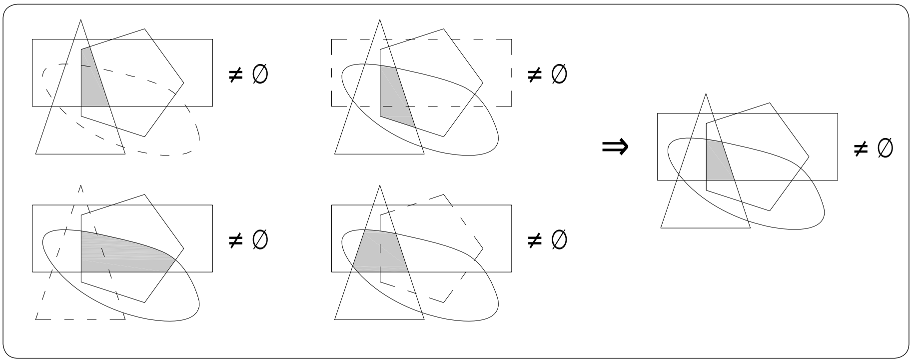
first symbolic pattern: propagation of convex intersections · The core part of the procedure described in the previous paragraph is the local propagation of a pure intersection ($\bigcap$*) of fundamental constraints (C_0, C_1, ..., C_n) onto the domain of a given father point (p_0) — see step (3.3.4.1.3.1), page 206. This local (symbolic) propagation uses here different methods according to the pattern of the constraints to be propagated and the constraint that propagates (C_i where $i \in [0,n]$). This paragraph describes a first method and the beginnings of other methods are introduced in the next one.

The required pattern for the first method of local propagation is as follows: every constraint included in the set {C_0, C_1, ..., C_n} must be either a HalfPlane, \Halfplane, DiscInside or \DiscOutside constraint. These four constraints have the property of being convex — *i.e.* if two points are included in their region, the line joining them is totally included in that region. Since the intersection of convex subsets is also a convex set, the pure intersection $\bigcap$* is therefore also convex.

Among all the properties of convex shapes, there is one proven by Eduard Helly in 1914 (Radon·1921) that is of particular interest for the purpose of propagation (figure 241):

figure 241 basic illustration of Helly's theorem.

"Suppose that C_0, C_1, ..., C_n is a finite collection of convex subsets of $\Re^d$, where $n>d$, if the intersection of every $d+1$ of these sets is non-empty, then the whole collection has a nonempty intersection, that is : $\bigcap_{0 \leq j \leq n} C_j \neq \emptyset$."



Since only the plane is considered here, d is equal to 2. The theorem consequently means that if there are more than two constraints in the intersection $\bigcap^*$ and if every sub-intersection made of three constraints of $\{C_0, C_1, ..., C_n\}$ is not empty, then the intersection $\bigcap^*$ is not empty. Therefore, each of these sub-intersections can be propagated on an individual basis. The propagated domain resulting from the intersection $\bigcap^*$ is equal to the intersection of the propagated domains resulting from each sub-intersection.

The great advantage of this property is that the propagation of only sixty typical intersections have to be known by the method, whatever the number n of fundamental constraints included in the intersection $\bigcap^*$. And these sixty intersections are made by a maximum of 3 fundamental constraints. They are all listed in the following table.

		constraints to be propagated														
		none	Hp	\Hp	DI	\DO	Hp ∩ Hp	\Hp ∩ \Hp	DI ∩ DI	\DO ∩ \DO	Hp ∩ \Hp	Hp ∩ DI	Hp ∩ \DO	\Hp ∩ DI	\Hp ∩ \DO	DI ∩ \DO
constraint that propagates	Hp	1	5	9	13	17	21	25	29	33	37	41	45	49	53	57
	\Hp	2	6	10	14	18	22	26	30	34	38	42	46	50	54	58
	DI	3	7	11	15	19	23	27	31	35	39	43	47	51	55	59
	\DO	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60

For example, if a point p_0 has to stay on the following intersection:

$p_0 \in C_0 \cap C_1$
where : C_0 = Straightedge[p_1 p_3 p_4]
C_1 = Straightedge[p_2 p_3 p_4]

this will have an empty domain when the two straightedges are not superimposed (figure 242). Although the domain of p_0 includes four HalfPlane constraints, the propagation domain of its father p_1 can actually be obtained systematically by intersecting the propagation domain of typical intersections that include a maximum of three HalfPlane constraints:

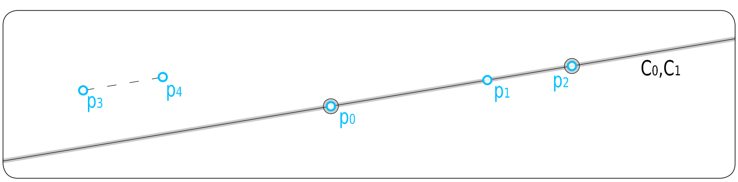


figure 242
Point p_0 is constrained on the intersection of two parallels.

$$
\begin{aligned}
\text{PropagationDomain}[p_1] &= \text{Dom}[p_0 \in C_2 \cap C_3 \cap C_4 \cap C_5] \textit{ propagated on } p_1 \textit{ by } C_2 \textit{ and } C_3 \\
&\quad \textit{where: } C_2 = \text{HalfPlane}[p_1\ p_3\ p_4] \\
&\quad\quad C_3 = \text{HalfPlane}[p_1\ p_4\ p_3] \\
&\quad\quad C_4 = \text{HalfPlane}[p_2\ p_3\ p_4] \\
&\quad\quad C_5 = \text{HalfPlane}[p_2\ p_4\ p_3] \\
&= \text{Dom}[p_0 \in C_2 \cap C_4 \cap C_5] \textit{ propagated on } p_1 \textit{ by } C_2 \\
&\quad \cap \text{Dom}[p_0 \in C_3 \cap C_4 \cap C_5] \textit{ propagated on } p_1 \textit{ by } C_3 \\
&= \text{TypicalDom\#1}\ [p_0 \in C_2] \textit{ propagated on } p_1 \textit{ by } C_2 \\
&\quad \cap \text{TypicalDom\#5}\ [p_0 \in C_2 \cap C_4] \textit{ propagated on } p_1 \textit{ by } C_2 \\
&\quad \cap \text{TypicalDom\#5}\ [p_0 \in C_2 \cap C_5] \textit{ propagated on } p_1 \textit{ by } C_2 \\
&\quad \cap \text{TypicalDom\#21}\ [p_0 \in C_2 \cap C_4 \cap C_5] \textit{ propagated on } p_1 \textit{ by } C_2 \\
&\quad \cap \text{TypicalDom\#1}\ [p_0 \in C_3] \textit{ propagated on } p_1 \textit{ by } C_3 \\
&\quad \cap \text{TypicalDom\#5}\ [p_0 \in C_3 \cap C_4] \textit{ propagated on } p_1 \textit{ by } C_3 \\
&\quad \cap \text{TypicalDom\#5}\ [p_0 \in C_3 \cap C_5] \textit{ propagated on } p_1 \textit{ by } C_3 \\
&\quad \cap \text{TypicalDom\#21}\ [p_0 \in C_3 \cap C_4 \cap C_5] \textit{ propagated on } p_1 \textit{ by } C_3
\end{aligned}
$$

The three typical propagations #1, #5 and #21 are defined later. They produce the following domains:

TypicalDom#1 $[p_0 \in C_2]$ *propagated on* p_1 *by* C_2	$= \Omega$	
TypicalDom#5 $[p_0 \in C_2 \cap C_4]$ *propagated on* p_1 *by* C_2	$= \Omega$	figure 243*, left*
TypicalDom#5 $[p_0 \in C_2 \cap C_5]$ *propagated on* p_1 *by* C_2	$= C_5$	figure 243*, right*
TypicalDom#21 $[p_0 \in C_2 \cap C_4 \cap C_5]$ *propagated on* p_1 *by* C_2	$= C_5$	figure 244*, left*
TypicalDom#1 $[p_0 \in C_3]$ *propagated on* p_1 *by* C_3	$= \Omega$	
TypicalDom#5 $[p_0 \in C_3 \cap C_4]$ *propagated on* p_1 *by* C_3	$= C_4$	figure 244*, right*
TypicalDom#5 $[p_0 \in C_3 \cap C_5]$ *propagated on* p_1 *by* C_3	$= \Omega$	figure 245*, left*
TypicalDom#21 $[p_0 \in C_3 \cap C_4 \cap C_5]$ *propagated on* p_1 *by* C_3	$= C_4$	figure 245*, right*

This means that the propagation domain of p_1 is $C_4 \cap C_5$, which compels the two straightedges to be superimposed.

The construction of the sixty typical propagations is mainly made complex by the fact that they must remain true for every positions of points. In other words, the graphical region that they produce must remain correct when the constraints of the sub-intersection have particular positions, orientations and radii. These special cases are usually due to coincidence of parameters, parallelism and tangencies.

Moreover, it is expected that typical propagations do not create interdependency. And if typical propagations create empty domains, these are assumed to have no effect and are therefore automatically changed by the entire plane domain Ω. This special device is helpful for some typical propagation — *e.g.* typical propagation #13b.

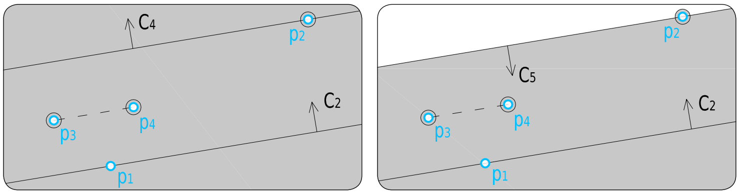


figure 243
(left) $p_0 \in (C_2 \cap C_4)$ propagated on p_1; (right) $p_0 \in (C_2 \cap C_5)$ propagated on p_1.

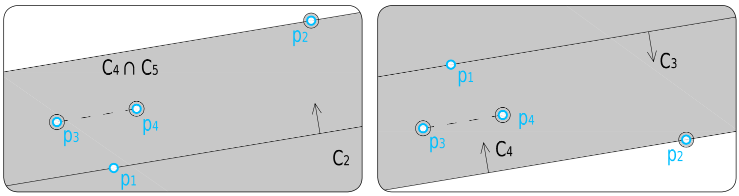


figure 244
(left) $p_0 \in (C_2 \cap C_4 \cap C_5)$ propagated on p_1; (right) $p_0 \in (C_3 \cap C_4)$ propagated on p_1.

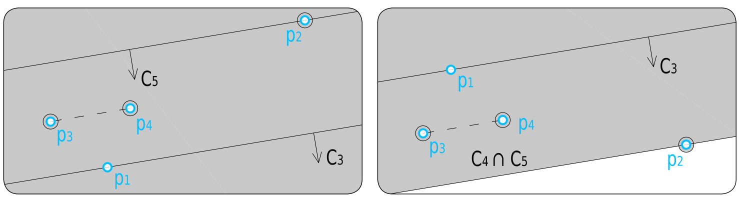


figure 245
(left) $p_0 \in (C_3 \cap C_5)$ propagated on p_1; (right) $p_0 \in (C_3 \cap C_4 \cap C_5)$ propagated on p_1.

Because each parameter of a fundamental constraint has a different role, it is finally expected that the propagation of each typical intersection returns a different domain depending on the place of the father parameter point in the typical intersection. For example, the typical intersection #5 involving two HalfPlane constraints should return a different domain for each of these cases:

#5a: PropagationDomain[p_1] *if* $p_0 \in$ HalfPlane[p_1 p_2 p_3] ∩ HalfPlane[p_4 p_5 p_6]
#5b: PropagationDomain[p_2] *if* $p_0 \in$ HalfPlane[p_1 p_2 p_3] ∩ HalfPlane[p_4 p_5 p_6]
#5c: PropagationDomain[p_3] *if* $p_0 \in$ HalfPlane[p_1 p_2 p_3] ∩ HalfPlane[p_4 p_5 p_6]
#5d: PropagationDomain[p_1] *if* $p_0 \in$ HalfPlane[p_1 p_1 p_2] ∩ HalfPlane[p_4 p_5 p_6]
#5e: PropagationDomain[p_1] *if* $p_0 \in$ HalfPlane[p_0 p_1 p_1] ∩ HalfPlane[p_4 p_5 p_6]
#5f: PropagationDomain[p_1] *if* $p_0 \in$ HalfPlane[p_1 p_1 p_1] ∩ HalfPlane[p_4 p_5 p_6]

Some of the sixty typical propagations are presented in the following lines.

#1 a, b, c • If a point p_0 must belong to the only constraint HalfPlane[p_1 p_2 p_3], its domain will always be non-empty. Therefore:

PropagationDomain[p_1]=Ω
PropagationDomain[p_2]=Ω
PropagationDomain[p_3]=Ω

#2 a, b, c • If a point p_0 must belong to the only constraint \HalfPlane[p_1 p_2 p_3], its domain is empty if p_2 and p_3 are coincident. Therefore:

PropagationDomain[p_1]=Ω
PropagationDomain[p_2]=\DiscInside[p_3 p_3 p_3]
PropagationDomain[p_3]=\DiscInside[p_2 p_2 p_2]

#3 a, b, c • If a point p_0 must belong to the only constraint DiscInside[p_1 p_2 p_3], its domain will always be non-empty. Therefore:

PropagationDomain[p_1]=Ω
PropagationDomain[p_2]=Ω
PropagationDomain[p_3]=Ω

#4 a, b, c • If a point p_0 must belong to the only constraint \DiscOutside[p_1 p_2 p_3], its domain is non-empty if p_2 and p_3 are coincident. Therefore:

PropagationDomain[p_1]=Ω
PropagationDomain[p_2]=\DiscInside[p_3 p_3 p_3]
PropagationDomain[p_3]=\DiscInside[p_2 p_2 p_2]

#5a • If a point p_0 must belong to the only constraint HalfPlane[p_1 p_2 p_3] $\cap$ HalfPlane[p_4 p_5 p_6] (figure 246), its domain will be non-empty if the constraints are not parallel and opposed. Therefore, its propagation on p_1 is:

PropagationDomain[p_1] = VHP[p_G p_4] $\cup$ VHP[p_H p_4] $\cup$ SE[p_4 p_5 p_6]
where: $p_A \in$ CC[p_2 p_3] $\cap$ CC[p_3 p_2] $\cap$ VHP[p_2 p_3]
$p_B \in$ CC[p_2 p_3] $\cap$ CC[p_3 p_2] $\cap$ VHP[p_3 p_2]
$p_C \in$ CC[p_5 p_6] $\cap$ CC[p_6 p_5] $\cap$ VHP[p_5 p_6]
$p_D \in$ CC[p_5 p_6] $\cap$ CC[p_6 p_5] $\cap$ VHP[p_6 p_5]
$p_E \in$ C[p_4 p_2 p_3] $\cap$ SE[p_4 p_5 p_6] $\cap$ HP[p_4 p_C p_D]
$p_F \in$ C[p_4 p_2 p_3] $\cap$ SE[p_4 p_5 p_6] $\cap$ HP[p_4 p_D p_C]
$p_G \in$ C[p_4 p_2 p_3] $\cap$ SE[p_4 p_2 p_3] $\cap$ HP[p_4 p_A p_B]
$p_H \in$ C[p_4 p_2 p_3] $\cap$ SE[p_4 p_2 p_3] $\cap$ \[p_E] $\cap$ (HP[p_4 p_B p_A] $\cup$ [p_F])

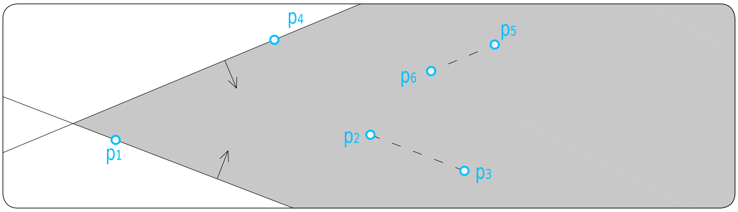


figure 246 the domain HalfPlane [p_1 p_2 p_3] $\cap$ HalfPlane [p_4 p_5 p_6].

Different regions resulting from this domain are illustrated in figure 247, figure 248, and figure 249. When p_2 and p_3 are coincident, p_4, p_E, p_F, p_G and pH are all coincident, which means that VHP[p_G p_4] $\cup$ VHP[p_H p_4] is equal to the entire plane — what is expected. When p_5 and p_6 are coincident, SE[p_4 p_5 p_6] is equal to the entire plane — what is expected.

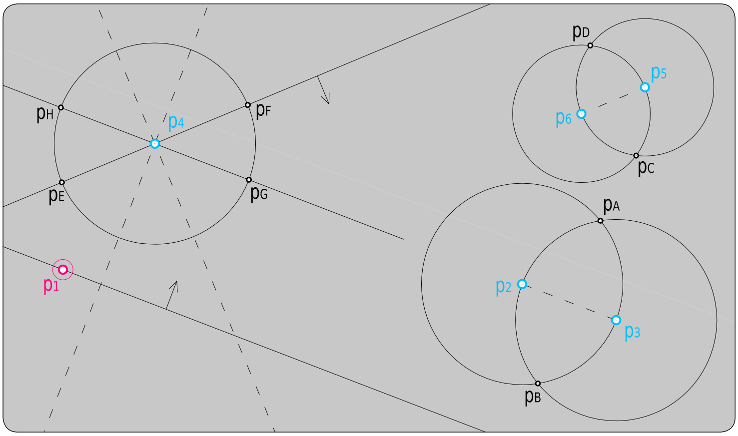


figure 247
the domain of p_1 so that HalfPlane $[p_1\ p_2\ p_3] \cap$ HalfPlane $[p_4\ p_5\ p_6]$ is not empty.

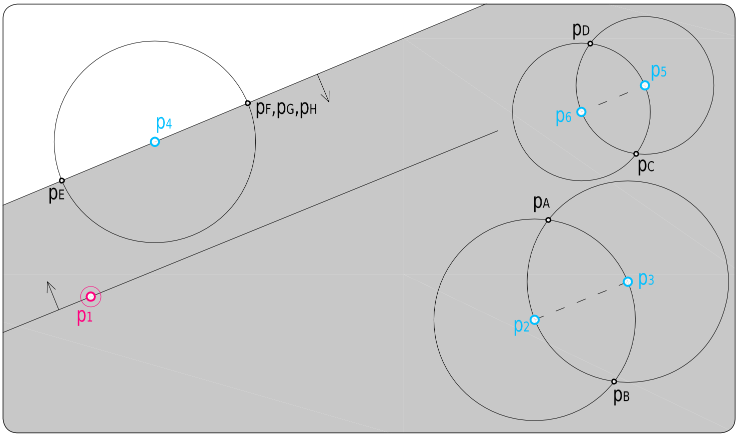


figure 248
the domain of p_1 so that HalfPlane $[p_1\ p_2\ p_3] \cap$ HalfPlane $[p_4\ p_5\ p_6]$ is not empty.

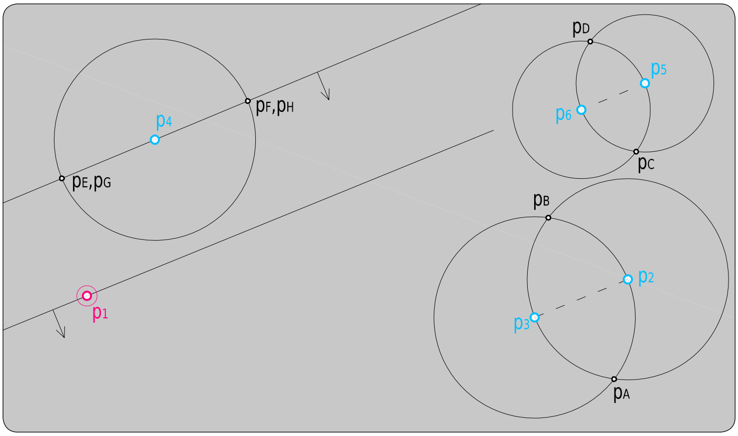


figure 249
the domain of p_1 so that HalfPlane $[p_1\ p_2\ p_3] \cap$ HalfPlane $[p_4\ p_5\ p_6]$ is not empty.

#5b • If a point p_0 must belong to the same constraint HalfPlane[p_1 p_2 p_3] ∩ HalfPlane[p_4 p_5 p_6], its propagation on p_2 is:

$$\text{PropagationDomain}[p_2] = \backslash SE[p_3\ p_5\ p_6] \cup HP[p_3\ p_1\ p_4] \cup HP[p_3\ p_4\ p_A]$$
$$\textit{where: } p_A \in C[p_4\ p_5\ p_6] \cap VSE[p_1\ p_4] \cap HP[p_4\ p_5\ p_6]$$

Different regions resulting from this domain are illustrated in figure 250 and figure 251. The resulting regions remain valid when p_2 and p_3 are coincident and/or when p_5 and p_6 are coincident.

figure 250
the domain of p_2 so that HalfPlane [p_1 p_2 p_3] ∩ HalfPlane [p_4 p_5 p_6] is not empty.

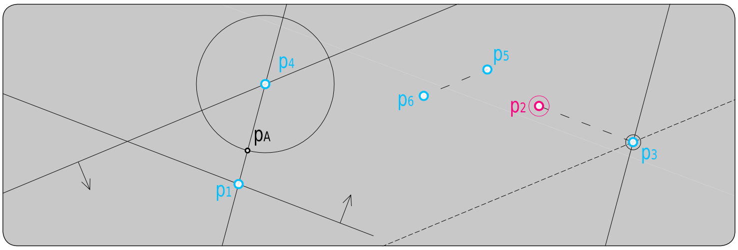
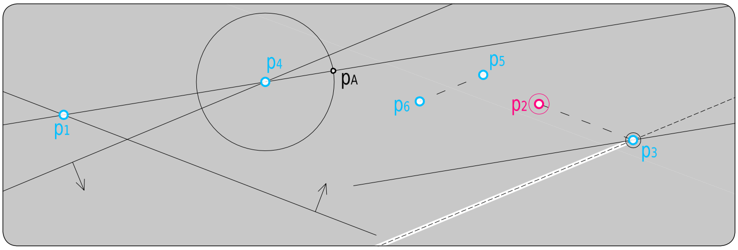


figure 251
the domain of p_2 so that HalfPlane [p_1 p_2 p_3] ∩ HalfPlane [p_4 p_5 p_6] is not empty.

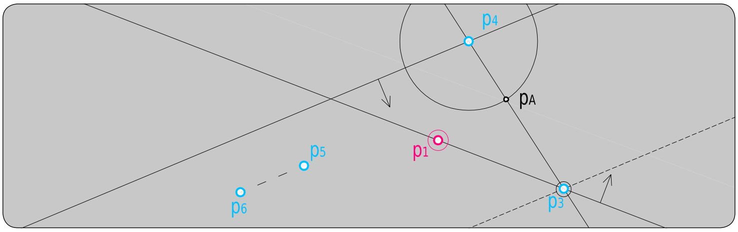


#5d • If a point p_0 must belong to the constraint HalfPlane[p_1 p_1 p_3] ∩ HalfPlane[p_4 p_5 p_6], its propagation on p_1 is:

$$\text{PropagationDomain}[p_1] = \backslash SE[p_3\ p_5\ p_6] \cup VHP[p_3\ p_4] \cup HP[p_3\ p_4\ p_A]$$
$$\textit{where: } p_A \in C[p_4\ p_5\ p_6] \cap VSE[p_3\ p_4] \cap HP[p_4\ p_5\ p_6]$$

Different regions resulting from this domain are illustrated in figure 252 and figure 253. The resulting regions remain valid when p_1 and p_3 are coincident and/or when p_5 and p_6 are coincident.

figure 252
the domain of p_1 so that HalfPlane [p_1 p_1 p_3] ∩ HalfPlane [p_4 p_5 p_6] is not empty.



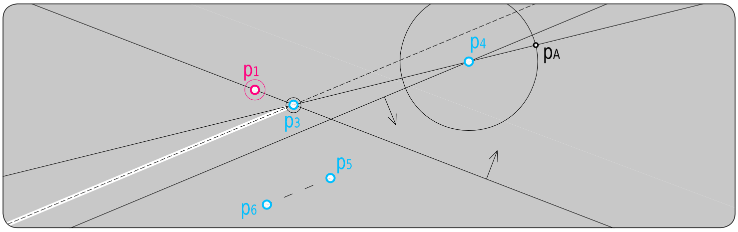


figure 253
the domain of p_1 so that HalfPlane $[p_1\ p_1\ p_3] \cap$ HalfPlane $[p_4\ p_5\ p_6]$ is not empty.

#9a • If a point p_0 must belong to the constraint HalfPlane$[p_1\ p_2\ p_3] \cap$ \HalfPlane$[p_4\ p_5\ p_6]$, its propagation on p_1 is:

$$\text{PropagationDomain}[p_1] = \backslash\text{VHP}[p_4\ p_G] \cup \backslash\text{VHP}[p_4\ p_H] \cup \backslash\text{HP}[p_I\ p_G\ p_H] \cup \text{SE}[p_I\ p_2\ p_3]$$

where:
$p_A \in \text{CC}[p_2\ p_3] \cap \text{CC}[p_3\ p_2] \cap \text{VHP}[p_2\ p_3]$
$p_B \in \text{CC}[p_2\ p_3] \cap \text{CC}[p_3\ p_2] \cap \text{VHP}[p_3\ p_2]$
$p_C \in \text{CC}[p_5\ p_6] \cap \text{CC}[p_6\ p_5] \cap \text{VHP}[p_6\ p_5]$
$p_D \in \text{CC}[p_5\ p_6] \cap \text{CC}[p_6\ p_5] \cap \text{VHP}[p_5\ p_6]$
$p_E \in \text{C}[p_4\ p_2\ p_3] \cap \text{SE}[p_4\ p_5\ p_6] \cap \text{HP}[p_4\ p_C\ p_D]$
$p_F \in \text{C}[p_4\ p_2\ p_3] \cap \text{SE}[p_4\ p_5\ p_6] \cap \text{HP}[p_4\ p_D\ p_C]$
$p_G \in \text{C}[p_4\ p_2\ p_3] \cap \text{SE}[p_4\ p_2\ p_3] \cap \text{HP}[p_4\ p_A\ p_B]$
$p_H \in \text{C}[p_4\ p_2\ p_3] \cap \text{SE}[p_4\ p_2\ p_3] \cap \backslash[p_E] \cap (\ \text{HP}[p_4\ p_B\ p_A] \cup [p_F]\ )$
$p_I \in \text{CC}[p_4\ p_5\ p_6] \cap \text{SE}[p_4\ p_A\ p_B] \cap \text{HP}[p_4\ p_3\ p_2]$

Different regions resulting from this domain are illustrated in figure 254 and figure 255. The resulting regions remain valid when p_1 and p_3 are coincident. Since the typical propagations #2b and #2c already deal with the same constraint, p_5 and p_6 will never be coincident.

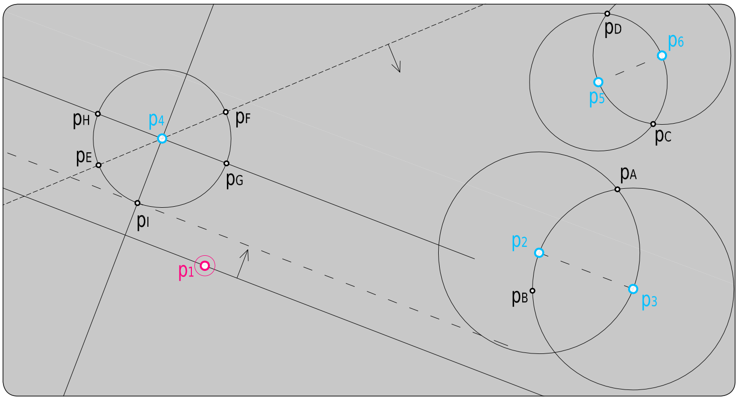


figure 254
the domain of p_1 so that HalfPlane $[p_1\ p_2\ p_3] \cap$ \HalfPlane $[p_4\ p_5\ p_6]$ is not empty.

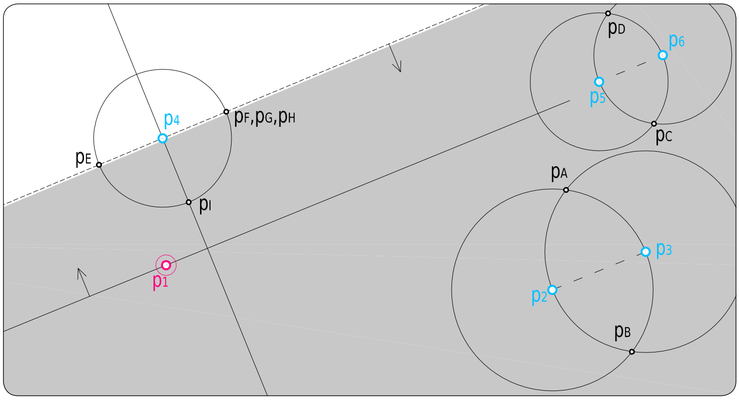


figure 255
the domain of p_1 so that HalfPlane $[p_1\ p_2\ p_3] \cap$ \HalfPlane $[p_4\ p_5\ p_6]$ is not empty.

#13a • If a point p_0 must belong to the constraint HalfPlane$[p_1\ p_2\ p_3] \cap$ DiscInside$[p_4\ p_5\ p_6]$, its propagation on p_1 is:

PropagationDomain$[p_1]$ = HP$[p_C\ p_3\ p_2]$

where: $p_A \in$ CC$[p_2\ p_3] \cap$ CC$[p_3\ p_2] \cap$ VHP$[p_2\ p_3]$

$p_B \in$ CC$[p_2\ p_3] \cap$ CC$[p_3\ p_2] \cap$ VHP$[p_3\ p_2]$

$p_C \in$ C$[p_4\ p_5\ p_6] \cap$ SE$[p_4\ p_A\ p_B] \cap$ HP$[p_4\ p_2\ p_3]$

A region resulting from this domain is illustrated in figure 256. The resulting regions remain valid when p_1 and p_3 are coincident and/or when p_5 and p_6 are coincident.

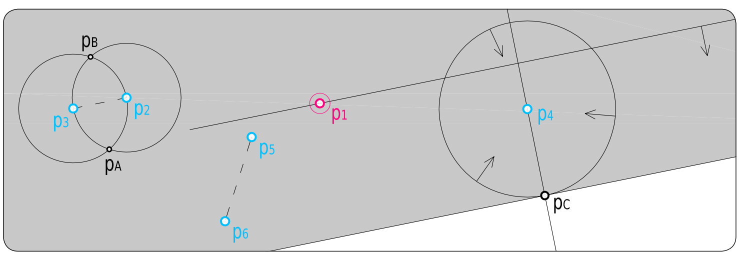


figure 256
the domain of p_1 so that HalfPlane $[p_1\ p_2\ p_3] \cap$ DiscInside $[p_4\ p_5\ p_6]$ is not empty.

#13b • If a point p_0 must belong to the constraint HalfPlane$[p_1\ p_2\ p_3] \cap$ DiscInside$[p_4\ p_5\ p_6]$, its propagation on p_2 is:

PropagationDomain$[p_2]$ = HP$[p_3\ p_F\ p_G] \cup$ HP$[p_3\ p_I\ p_H]$

where: $p_A \in$ CC$[p_1\ p_4] \cap$ CC$[p_4\ p_1] \cap$ VHP$[p_1\ p_4]$

$p_B \in$ CC$[p_1\ p_4] \cap$ CC$[p_4\ p_1] \cap$ VHP$[p_4\ p_1]$

$p_C \in$ VSE$[p_1\ p_4] \cap$ VSE$[p_A\ p_B] \cap$ CC$[p_1\ p_4]$

$p_D \in$ CC$[p_C\ p_4] \cap$ C$[p_4\ p_5\ p_6] \cap$ VHP$[p_C\ p_4]$

$p_E \in$ CC$[p_C\ p_4] \cap$ C$[p_4\ p_5\ p_6] \cap$ VHP$[p_4\ p_C]$

$p_F \in CC[p_4\ p_D] \cap CC[p_D\ p_4] \cap VHP[p_4\ p_D]$
$p_G \in CC[p_4\ p_D] \cap CC[p_D\ p_4] \cap VHP[p_D\ p_4]$
$p_H \in CC[p_4\ p_E] \cap CC[p_E\ p_4] \cap VHP[p_4\ p_E]$
$p_I \in CC[p_4\ p_E] \cap CC[p_E\ p_4] \cap VHP[p_D\ p_E]$

Different regions resulting from this domain are illustrated in figure 257 and figure 258. The resulting regions remain valid when p_1 and p_3 are coincident and/or when p_5 and p_6 are coincident.

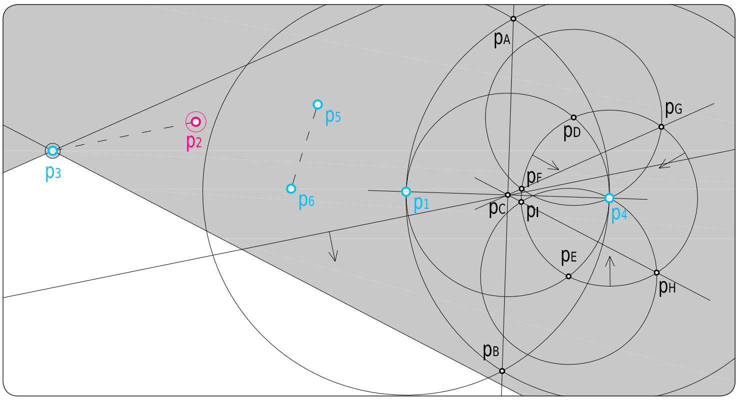


figure 257
the domain of p_2 so that HalfPlane $[p_1\ p_2\ p_3] \cap$ DiscInside $[p_4\ p_5\ p_6]$ is not empty.

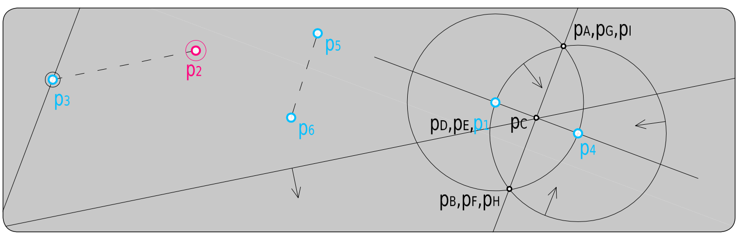


figure 258
the domain of p_2 so that HalfPlane $[p_1\ p_2\ p_3] \cap$ DiscInside $[p_4\ p_5\ p_6]$ is not empty.

#15a • If a point p_0 must belong to the constraint DiscInside$[p_1\ p_2\ p_3] \cap$ DiscInside$[p_4\ p_5\ p_6]$ (figure 259), its propagation on p_1 is:

PropagationDomain$[p_1]$ = CDI$[p_4\ p_D]$
where: $p_A \in C[p_4\ p_5\ p_6]$
$p_B \in CC[p_4\ p_A] \cap CC[p_A\ p_4] \cap VHP[p_4\ p_A]$
$p_C \in CC[p_4\ p_A] \cap CC[p_A\ p_4] \cap VHP[p_A\ p_4]$
$p_D \in CC[p_A\ p_2\ p_3] \cap VSE[p_4\ p_A] \cap VHP[p_B\ p_C] \cap DO[p_4\ p_5\ p_6]$

A region resulting from this domain is illustrated in figure 260. The resulting regions remain valid when p_1 and p_3 are coincident and/or when p_5 and p_6 are coincident.

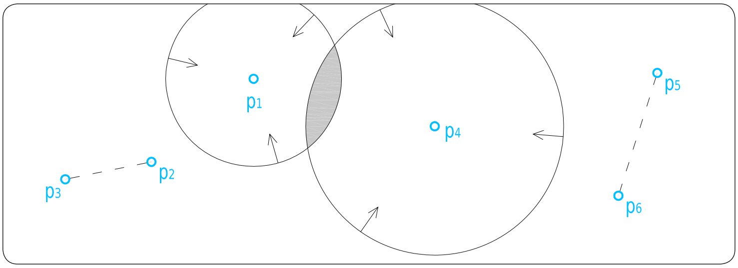


figure 259
the domain
DiscInside $[p_1\ p_2\ p_3]$
$\cap$ DiscInside
$[p_4\ p_5\ p_6]$.

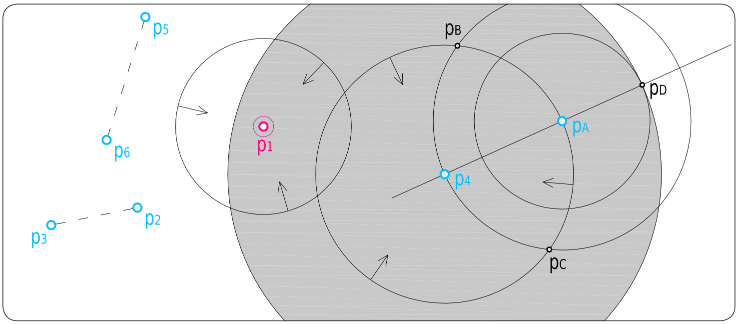


figure 260
the domain of p_1
so that DiscInside
$[p_1\ p_2\ p_3] \cap$
DiscInside $[p_4\ p_5\ p_6]$
is not empty.

#37a • If a point p_0 must belong to the constraint HalfPlane$[p_1\ p_2\ p_3] \cap$ HalfPlane$[p_4\ p_5\ p_6] \cap$ HalfPlane$[p_7\ p_8\ p_9]$ (figure 261), its propagation on p_1 is:

$$\mathrm{PropagatedDomain}[p_1] = \mathrm{VHP}[p_F\ p_A] \cup \mathrm{VHP}[p_G\ p_A] \cup \mathrm{VSE}[p_A\ p_H] \cup \mathrm{VSE}[p_A\ p_I]$$

where: $p_A \in \mathrm{SE}[p_4\ p_5\ p_6] \cap \mathrm{SE}[p_7\ p_8\ p_9]$

$p_B \in \mathrm{C}[p_A\ p_5\ p_6] \cap \mathrm{SE}[p_4\ p_5\ p_6] \cap \mathrm{HP}[p_7\ p_8\ p_9]$

$p_C \in \mathrm{C}[p_A\ p_8\ p_9] \cap \mathrm{SE}[p_7\ p_8\ p_9] \cap \mathrm{HP}[p_4\ p_5\ p_6] \cap \backslash[p_B]$

$p_D \in \mathrm{CC}[p_2\ p_3] \cap \mathrm{CC}[p_3\ p_2] \cap \mathrm{VHP}[p_2\ p_3]$

$p_E \in \mathrm{CC}[p_2\ p_3] \cap \mathrm{CC}[p_3\ p_2] \cap \mathrm{VHP}[p_3\ p_2]$

$p_F \in \mathrm{C}[p_A\ p_2\ p_3] \cap \mathrm{SE}[p_A\ p_2\ p_3] \cap \mathrm{HP}[p_A\ p_D\ p_E]$

$p_G \in \mathrm{C}[p_A\ p_2\ p_3] \cap \mathrm{VSE}[p_A\ p_F] \cap (\backslash\mathrm{VHP}[p_B\ p_A] \cup \backslash\mathrm{VHP}[p_C\ p_A])$
$\cap (\backslash[p_F] \cup (\mathrm{VHP}[p_A\ p_B] \cap \mathrm{VHP}[p_A\ p_C]))$

$p_H \in \mathrm{C}[p_A\ p_5\ p_6] \cap \mathrm{VSE}[p_A\ p_F]$

$p_I \in \mathrm{C}[p_A\ p_8\ p_9] \cap \mathrm{VSE}[p_A\ p_F]$

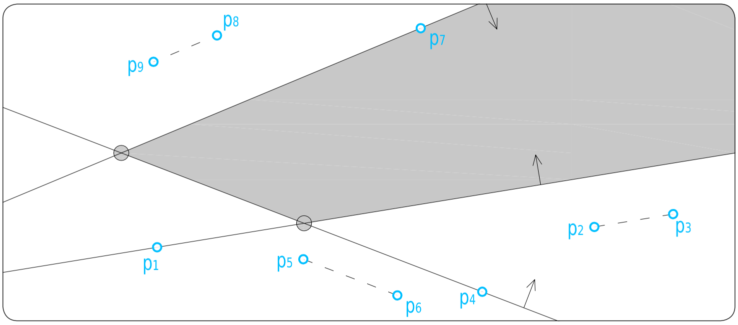


figure 261
the domain
HalfPlane $[p_1\ p_2\ p_3]$
$\cap$ HalfPlane
$[p_4\ p_5\ p_6] \cap$
HalfPlane $[p_7\ p_8\ p_9]$.

Different regions resulting from this domain are illustrated in the following figures (figure 262, figure 263 and figure 264). The resulting regions remain valid when p_1 and p_3 are coincident and/or when p_5 and p_6 are coincident.

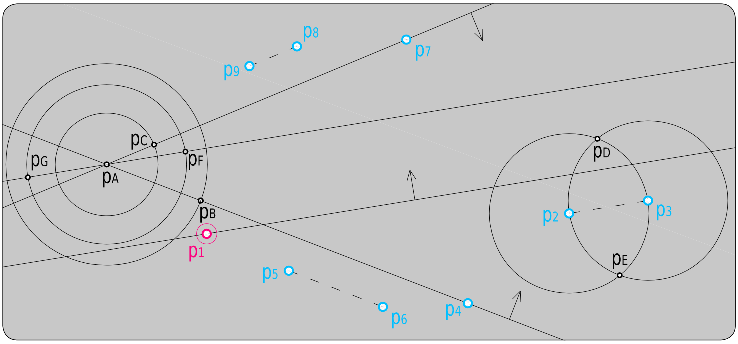


figure 262
the domain of p_1 so that HalfPlane [p_1 p_2 p_3] ∩ HalfPlane [p_4 p_5 p_6] ∩ HalfPlane [p_7 p_8 p_9] is not empty.

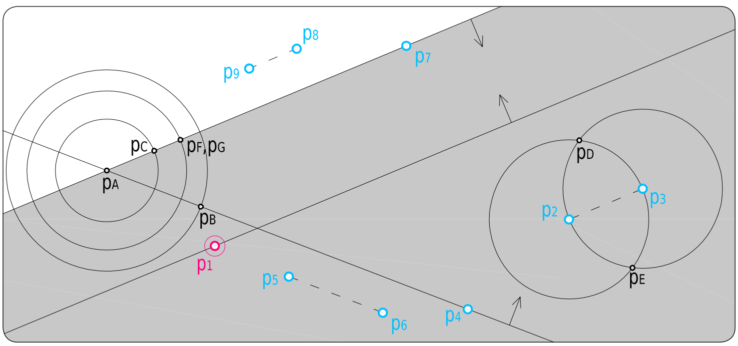


figure 263
the domain of p_1 so that HalfPlane [p_1 p_2 p_3] ∩ HalfPlane [p_4 p_5 p_6] ∩ HalfPlane [p_7 p_8 p_9] is not empty.

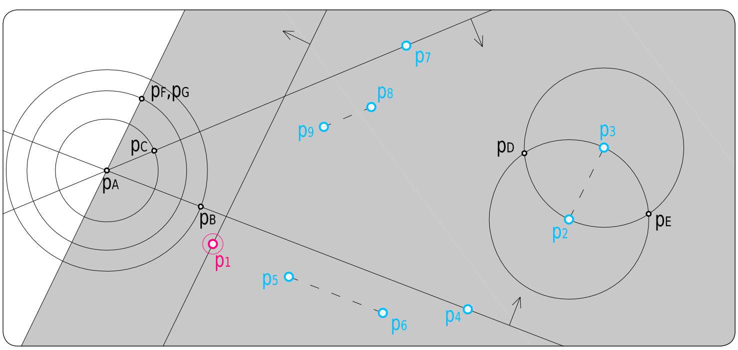


figure 264
the domain of p_1 so that HalfPlane [p_1 p_2 p_3] ∩ HalfPlane [p_4 p_5 p_6] ∩ HalfPlane [p_7 p_8 p_9] is not empty.

second symbolic pattern: propagation by outsides of discs · Other symbolic propagations appear to be feasible. Some of them are illustrated in this paragraph.

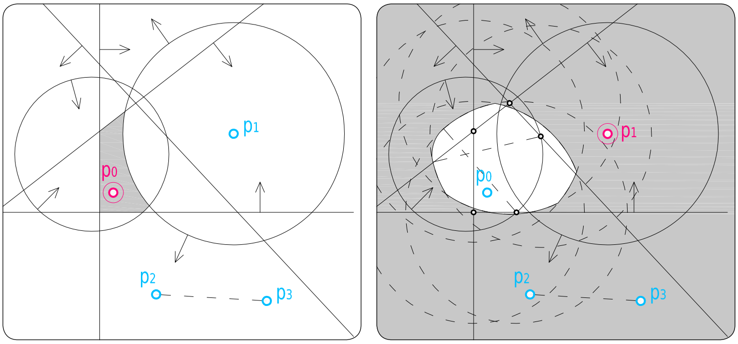
A first pattern that is very easily automated comprises no more than one Halfplane and an undetermined number of DiscOutside or \DiscInside constraints. Since the intersection of these constraints always include points at infinity, the propagation domain resulting from this pattern is the entire plane Ω.

Another pattern that may be propagated with symbolic algorithms is the one for which the constraint that propagates is a DiscOutside or a \DiscInside constraint and the intersection to be propagated includes only HalfPlane, DiscInside and \DiscOutside constraints. This means that the intersection to be propagated is a convex shape whose border is made of segments of lines and arc of circles. The resulting domain is a union of (1) DiscOutside and \DiscInside constraints that are centred on each vertex and that have a radius equal to the radius of the constraint that propagates and (2) intersections of DiscOutside and \DiscInside constraints with sectors — *i.e.* intersection of two secant half-planes — that are centred on each arc of circle, that have a radius equal to the subtraction of the radius of the constraint that propagates by the radius of the arc of circle — if the difference is negative, the radius is equal to zero —, and that are oriented opposite to the arc of circle, with respect of its centre.

Since the number of vertices can vary depending on the orientation of the half-planes, the propagation domain must use dynamic conditional statements developed in sub-section 20 ("dynamic conditional geometric statements", page 233), which complicates its construction.

A propagation domain that matches this pattern is illustrated in figure 265. The constraint that propagates is a DiscOutside[p_1 p_2 p_3] constraint and all the constraints are applied onto point p_0.

figure 265 (left) the domain of p_0 and (right) the propagation domain of p_1 (so that the domain of p_0 is not empty).



similarity between the locus and the propagation domain · There is a noteworthy property between the locus of points — see paragraph entitled "locus of positions" (page 164) —, the permutation of dependencies and the propagation domain (when the geometric construction is not interdependent): the locus of a point p_0 resulting from the displacement of its father point p_1 is identical to the PropagationDomain[p_0] once all required permutations of dependencies have been done in order to make p_1 the child or grandchild of p_0.

This property can be understood with the geometric construction of a piston (figure 266):

$p_1 \in C_1$
$p_2 \in C_0 \cap C_2$
where: C_0 = Straightedge[p_0 p_3 p_4]
C_1 = Compass[p_0 p_5 p_6]
C_2 = Compass[p_1 p_7 p_8]

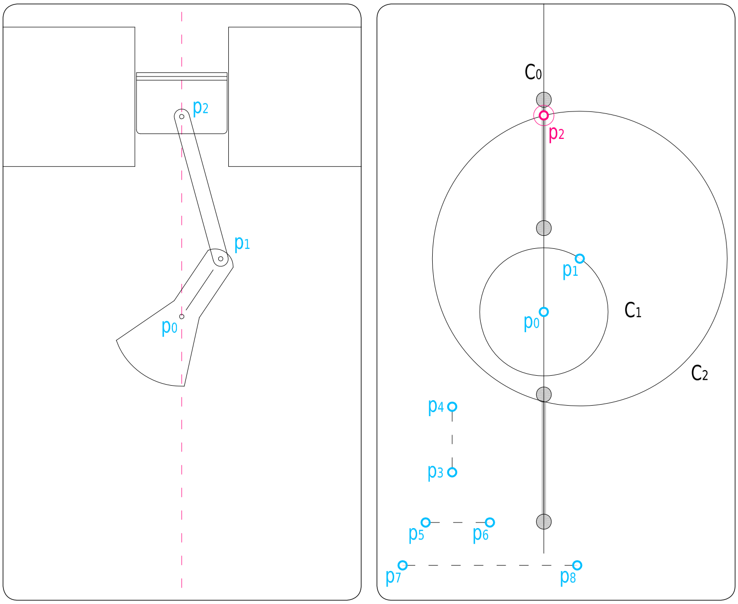


figure 266
(left) schematic functioning of a piston and (right) its geometrical construction with the locus domain of p_2 when p_1 moves.

The goal is to describe the locus of the centre of the piston — *i.e.* p_2 — when the crankpin — *i.e.* p_1 — rotates around the crankshaft — *i.e.* p_0. The directed graph of dependencies in figure 267 shows that p_2 is effectively the child of p_1.

figure 267
directed graph of dependencies prior to permutation.

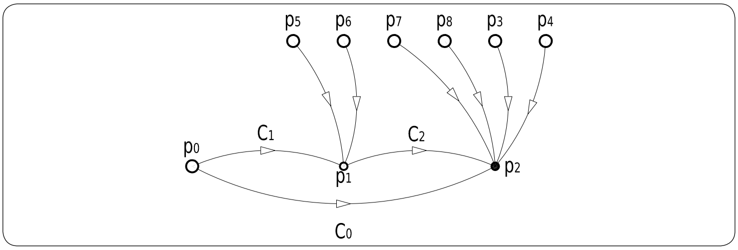
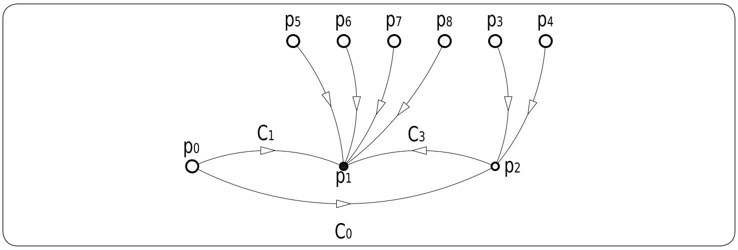


figure 268
directed graph of dependencies after permutation.



The first step is to switch the dependencies between p_1 and p_2 (figure 268):

$p_1 \in C_1 \cap C_3$
$p_2 \in C_0$
where: C_0 = Straightedge[p_0 p_3 p_4]
C_1 = Compass[p_0 p_5 p_6]
C_3 = Compass[p_2 p_7 p_8]

The second step calculates the propagation domain of p_2, which is equivalent to:

PropDomain[p_2] = Domain[$p_1 \in C_4 \cap C_5 \cap C_6 \cap C_7$) *propagated on* p_2 *by* C_6 *and* C_7
where : C_4 = DiscInside[p_0 p_5 p_6]
C_5 = DiscOutside[p_0 p_5 p_6]
C_6 = DiscInside[p_2 p_7 p_8]
C_7 = DiscOutside[p_2 p_7 p_8]

PropDomain[p_2] = Domain[$p_1 \in C_4 \cap C_5 \cap C_6$) *propagated on* p_2 *by* C_6
∩ Domain[$p_1 \in C_4 \cap C_5 \cap C_7$) *propagated on* p_2 *by* C_7

PropDomain[p_2] = Domain[$p_1 \in C_4 \cap C_6$) *propagated on* p_2 *by* C_6
∩ Domain[$p_1 \in C_4 \cap C_7$) *propagated on* p_2 *by* C_7
∩ Domain[$p_1 \in C_5 \cap C_6$) *propagated on* p_2 *by* C_6
∩ Domain[$p_1 \in C_4 \cap C_5 \cap C_6$) *propagated on* p_2 *by* C_6
∩ Domain[$p_1 \in C_4 \cap C_5 \cap C_7$) *propagated on* p_2 *by* C_7

PropDomain[p_2] = CollapsibleDiscInside[p_0 p_A] ∩ CollapsibleDiscOutside[p_0 p_B]

The propagation domain is illustrated in figure 269. Its intersection with the input domain of p_2 gives the locus of p_2 when p_1 moves:

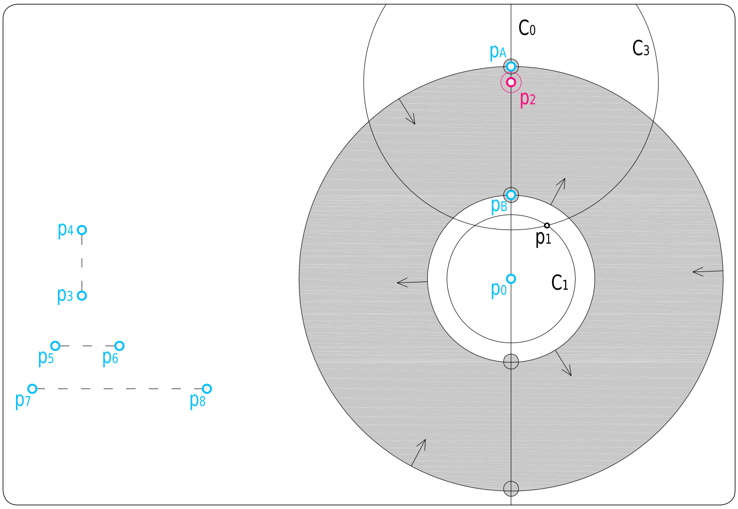


figure 269
the propagation domain of p_2.

$$\text{SolutionDomain}[p_2] = \text{CollapsibleDiscInside}[p_0\ p_A] \cap \text{CollapsibleDiscOutside}[p_0\ p_B] \cap \text{Straightedge}[p_0\ p_3\ p_4]$$

This purely symbolic resolution clearly differs from other methods proposed in constraint-based geometry literature — see Freixas/...·2008 and Hidalgo/...·2012 for comparison.

20 dynamic conditional geometric statements

This section shows how conditional statements such as If/Else conditions can be replaced graphically by combinations of HalfPlane and DiscInside constraints. Although these constraints are constructed symbolically as usual, their inner behaviour gives conditional results that are dynamically updated when points are moving.

In other words, the role of a constraint of this kind is to tell whether a point presently meets a certain geometric relationship or not. If the answer is true, the domain of the constraint is a single position. If the answer is false, the domain of the constraint is the inverse of the position — *i.e.* the entire plane without the position. These constraints are constructed in such a way that the answer is always correct whatever the position of the parameter points.

The following paragraphs presents five conditional constraints that can be combined together in order to construct more complex conditions. They produce the following results:

CoincidenceCondition[p_0 p_1 p_2]
  = [p_0] *IF* p_1 *and* p_2 *are coincident*
  = \[p_0] *otherwise*

LateralityCondition[p_0 p_1 p_2 p_3 p_4]
  = [p_0] *IF* p_1 *is on the left of or in line with* p_2 *according to the direction* p_3p_4
  = \[p_0] *otherwise*

ProximityCondition[p_0 p_1 p_2 p_3 p_4]
  = [p_0] *IF the distance* p_1p_2 *is smaller than or equal to the distance* p_3p_4
  = \[p_0] *otherwise*

ConjunctiveCondition[p_0 p_1 p_2 p_3 p_4]
  = [p_0] *IF (*p_1 *and* p_2 *are coincident) AND (*p_3 *and* p_4 *are coincident)*
  = \[p_0] *otherwise*

DisjunctiveCondition[p_0 p_1 p_2 p_3 p_4]
= [p_0] *IF (*p_1 *and* p_2 *are coincident) OR (*p_3 *and* p_4 *are coincident)*
= \[p_0] *otherwise*

the CoincidenceCondition constraint · The construction of the CoincidenceCondition constraint is drawn from the TranslatedPosition constraint — see paragraph entitled "dynamic compass-and-straightedge constructions" (page 165) — and takes advantage of the fact that the domain of a Straightedge[p_A p_B p_C] constraint is the entire plane when p_B and p_C are coincident. The two possible cases are shown in figure 270 and figure 271:

CoincidenceCondition[p_0 p_1 p_2] :=
\(SE[p_0 p_1 p_2] ∩ SE[p_0 p_A p_B]) ∪ (C[p_1 p_2 p_0] ∩ SE[p_1 p_2 p_0] ∩ HP[p_1 p_C p_D])

where: p_A ∈ (CC[p_1 p_2] ∩ CC[p_2 p_1] ∩ VHP[p_1 p_2])
p_B ∈ (CC[p_1 p_2] ∩ CC[p_2 p_1] ∩ VHP[p_2 p_1])
p_C ∈ (CC[p_2 p_0] ∩ CC[p_0 p_2] ∩ VHP[p_2 p_0])
p_D ∈ (CC[p_2 p_0] ∩ CC[p_0 p_2] ∩ VHP[p_0 p_2])

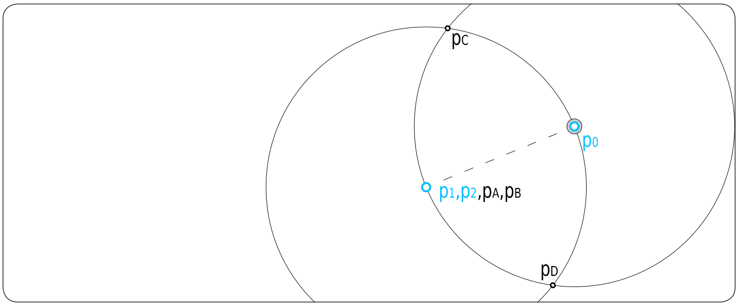


figure 270
the domain of Coincidence Condition [p_0 p_1 p_2] when p_1 and p_2 are coincident.

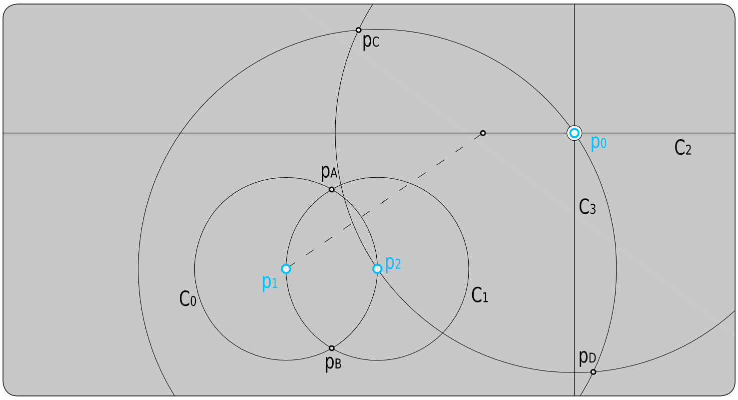


figure 271
the domain of Coincidence Condition [p_0 p_1 p_2] when p_1 and p_2 are not coincident.

When p_1 and p_2 are coincident, the domain of CoincidenceCondition[p_0 p_1 p_2] is equal to the position of p_0. When p_1 and p_2 are not coincident, the domain is equal to the inverse of the position p_0.

the LateralityCondition constraint · The LateralityCondition[p_0 p_1 p_2 p_3 p_4] constraint should present the following domains:

- *if* Laterality[p_1 p_2 p_3 p_4]*:* LateralityCondition[p_0 p_1 p_2 p_3 p_4] = [p_0]
- *if not:* LateralityCondition[p_0 p_1 p_2 p_3 p_4] = \[p_0]

As a result, this constraint can be constructed as follows (figure 272 and figure 273):

LateralityCondition[p_0 p_1 p_2 p_3 p_4] := CoincidenceCondition[p_0 p_A p_1]
where: $p_A \in$ ([p_1] ∪ [p_2]) ∩ (\HP[p_1 p_4 p_3] ∪ [p_1]) ∩ HP[p_2 p_3 p_4]

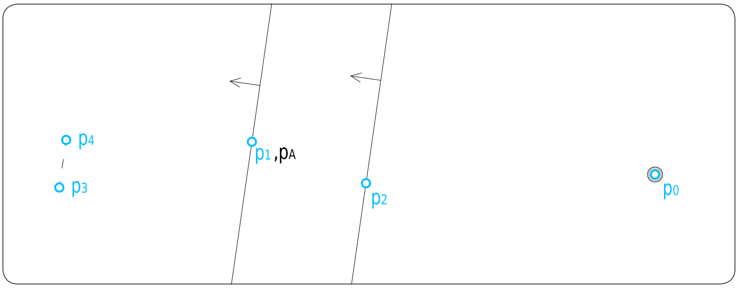


figure 272
the domain of the Laterality Condition [p_0 p_1 p_2 p_3 p_4] constraint when the Laterality [p_1 p_2 p_3 p_4] relationship is true.

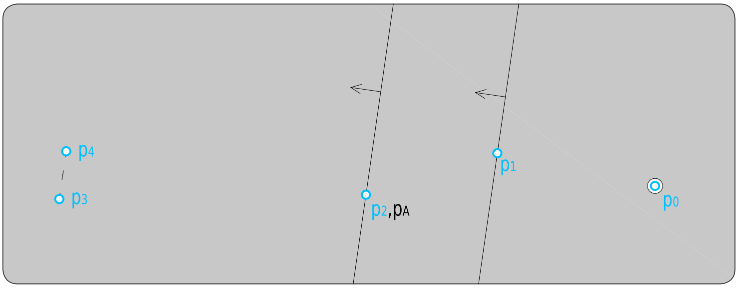


figure 273
the domain of the Laterality Condition [p_0 p_1 p_2 p_3 p_4] constraint when the Laterality [p_1 p_2 p_3 p_4] relationship is false.

The domain of this constraint remains relevant when the direction p_3p_4 does not exist:

($p_3 = p_4$) ⟶ (\HP[p_1 p_4 p_3] = ∅) ⟶ ($p_A \in$ [p_0]) ⟶ *the condition is verified*

The domain remains valid when p_1, p_2, p_3 or p_4 change position.

the ProximityCondition constraint · The ProximityCondition[p_0 p_1 p_2 p_3 p_4] constraint should present the following domains:

- *if* Proximity[p_1 p_2 p_3 p_4]: ProximityCondition[p_0 p_1 p_2 p_3 p_4] = [p_0]
- *if not:* ProximityCondition[p_0 p_1 p_2 p_3 p_4] = \[p_0]

The construction of this constraint first copies the two distances onto circles that are centred on the same point p_5. Point p_7 is defined in order to be positioned on the expected greatest distance — *i.e.* the distance p_3p_4. Another point (p_8) is then defined in order to be positioned on the actual greatest distance. The positions of p_7 and p_8 are finally compared with the CoincidenceCondition constraint (figure 274 and figure 275):

ProximityCondition[p_0 p_1 p_2 p_3 p_4] := CoincidenceCondition[p_0 p_D p_C]
where: $p_B \in$ C[p_A p_1 p_2]
$p_C \in$ C[p_A p_3 p_4]
$p_D \in$ ([p_B] ∪ [p_C]) ∩ DO[p_A p_1 p_2] ∩ (\DI[p_A p_3 p_4] ∪ [p_C])

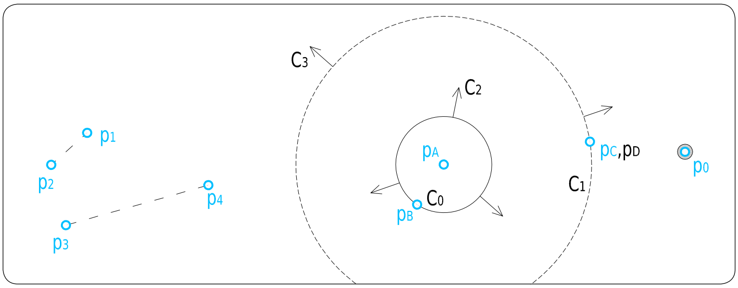


figure 274
the domain of the Proximity Condition [p_0 p_1 p_2 p_3 p_4] constraint when the Proximity [p_1 p_2 p_3 p_4] relationship is true.

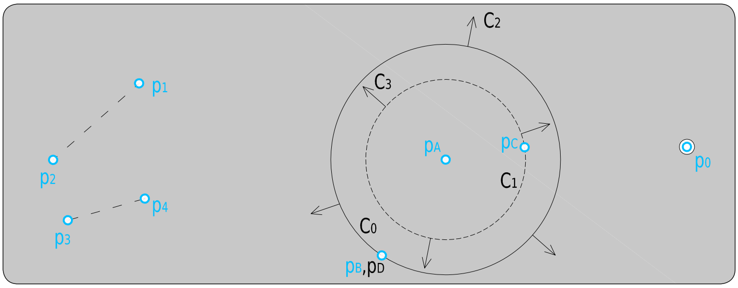


figure 275
the domain of the Proximity Condition [p_0 p_1 p_2 p_3 p_4] constraint when the Proximity [p_1 p_2 p_3 p_4] relationship is false.

This constraint remains valid when (a) p_1 and p_2 are coincident, (b) p_3 and p_4 are coincident and (c) p_1, p_2, p_3 and p_4 are all coincident. It also remains valid when a greatest distance decreases progressively and becomes smaller than the other distance.

It can be noted that the same construction can be used to find the greatest of two given distances. If the constraint MaxDistanceCompass[p_0 p_1 p_2 p_3 p_4] is applied to a point p_5, then the distance p_0p_5 will always be the greatest of the two distances p_1p_2 and p_3p_4, whatever they are or are becoming (figure 276):

MaxDistanceCompass[p_0 p_1 p_2 p_3 p_4] := Compass[p_0 p_A p_D]
where: $p_B \in$ C[p_A p_1 p_2]
$p_C \in$ C[p_A p_3 p_4]
$p_D \in$ ([p_B] ∪ [p_C]) ∩ DO[p_A p_1 p_2] ∩ (\DI[p_A p_3 p_4] ∪ [p_C])

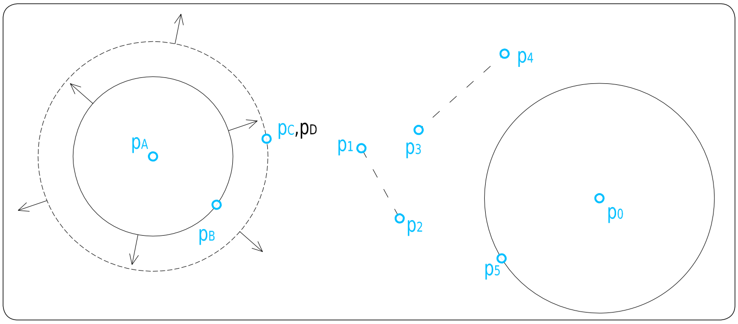


figure 276
application of a MaxDistance Compass [$p_0p_1p_2p_3p_4$] constraint onto p_5.

the ConjunctiveCondition constraint · The domain of a ConjunctiveCondition [p_0 p_1 p_2 p_3 p_4] constraint is directly deduced from the logical conjunction ∧:

· *if* Coincidence[p_1p_2] ∧ Coincidence[p_3p_4]*:* ConjunctiveCond[$p_0p_1p_2p_3p_4$] = [p_0]
· *if* Coincidence[p_1p_2] ∧ ¬Coincidence[p_3p_4]*:* ConjunctiveCond[$p_0p_1p_2p_3p_4$] = \[p_0]
· *if* ¬Coincidence[p_1p_2] ∧ Coincidence[p_3p_4]*:* ConjunctiveCond[$p_0p_1p_2p_3p_4$] = \[p_0]
· *if* ¬Coincidence[p_1p_2] ∧ ¬Coincidence[p_3p_4]*:* ConjunctiveCond[$p_0p_1p_2p_3p_4$] = \[p_0]

The construction of this constraint is as follows (figure 277 and figure 278):

ConjunctiveCondition[p_0 p_1 p_2 p_3 p_4] := CoincidenceCondition[p_B p_C p_0]
where: $p_B \in$ C[p_A p_1 p_2]
$p_C \in$ C[p_A p_3 p_4] ∩ CC[p_B p_A]

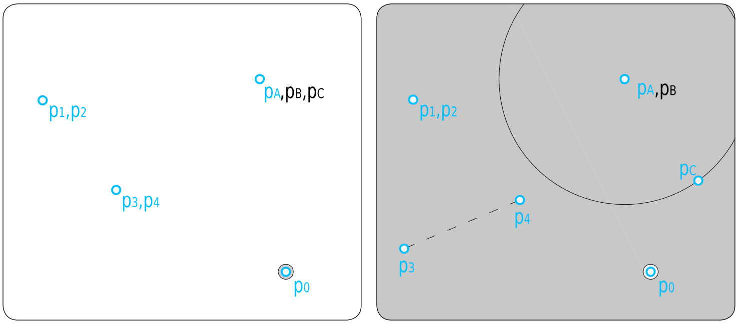


figure 277 the domain of the Conjunctive Condition constraint (left) when Coincidence $[p_1\ p_2] \wedge$ Coincidence $[p_3\ p_4]$ and (right) when Coincidence $[p_1\ p_2] \wedge \neg$Coincidence $[p_3\ p_4]$.

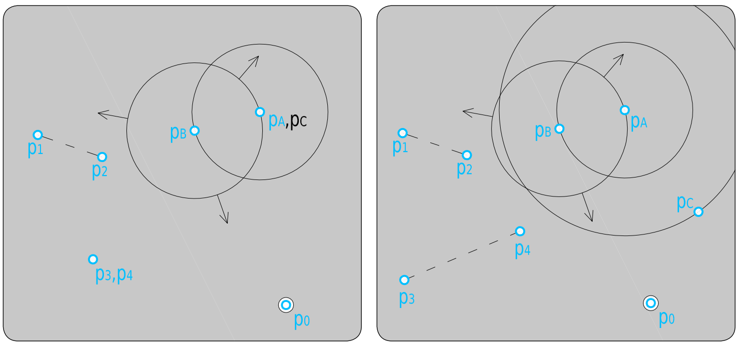


figure 278 the domain of the Conjunctive Condition constraint (left) when ¬Coincidence $[p_1\ p_2] \wedge$ Coincidence $[p_3\ p_4]$ and (right) when ¬Coincidence $[p_1\ p_2] \wedge$ ¬Coincidence $[p_3\ p_4]$.

the DisjunctiveCondition constraint · The domain of a DisjunctiveCondition $[p_0\ p_1\ p_2\ p_3\ p_4]$ constraint is directly deduced from the logical disjunction ∨:

- *if* Coincidence$[p_1p_2]$ ∨ Coincidence$[p_3p_4]$: DisjunctiveCond$[p_0p_1p_2p_3p_4] = [p_0]$
- *if* Coincidence$[p_1p_2]$ ∨ ¬Coincidence$[p_3p_4]$: DisjunctiveCond$[p_0p_1p_2p_3p_4] = [p_0]$
- *if* ¬Coincidence$[p_1p_2]$ ∨ Coincidence$[p_3p_4]$: DisjunctiveCond$[p_0p_1p_2p_3p_4] = [p_0]$
- *if* ¬Coincidence$[p_1p_2]$ ∨ ¬Coincidence$[p_3p_4]$: DisjunctiveCond$[p_0p_1p_2p_3p_4] = \setminus[p_0]$

The construction of this constraint is as follows (figure 279 and figure 280):

DisjunctiveCondition$[p_0\ p_1\ p_2\ p_3\ p_4]$:= $\setminus$CoincidenceCondition$[p_B\ p_E\ p_0]$
where: $p_B \in \setminus[p_A]$
$p_C \in$ CoincidenceCondition$[p_1\ p_2\ p_A] \cap ([p_A] \cup [p_B])$
$p_D \in$ CoincidenceCondition$[p_3\ p_4\ p_A] \cap ([p_A] \cup [p_B])$
$p_E \in$ MidPoint$[p_C\ p_D]$

The MidPoint constraint as been defined in the paragraph entitled "dynamic compass-and-straightedge constructions" (page 165).

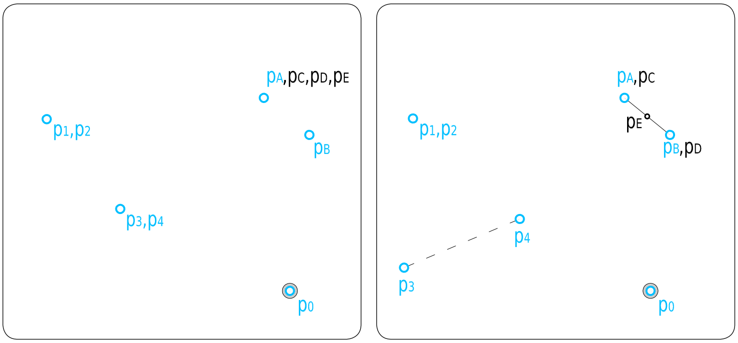


figure 279
the domain of the Disjunctive Condition constraint (left) when Coincidence [p_1 p_2] ∨ Coincidence [p_3 p_4] and (right) when Coincidence [p_1 p_2] ∨ ¬Coincidence [p_3 p_4].

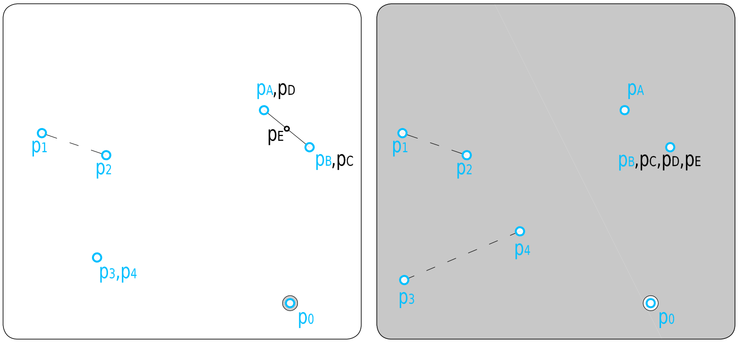


figure 280
the domain of the Disjunctive Condition constraint (left) when ¬Coincidence [p_1 p_2] ∨ Coincidence [p_3 p_4] and (right) when ¬Coincidence [p_1 p_2] ∨ ¬Coincidence [p_3 p_4].

logical conditions that return other results · The previous five constraints can be combined with other constraints in order to match other behaviours. This paragraph shows three different uses of the resulting domains of conditional constraints.

(1) The previous five constraints act like logical bijective equivalence — *i.e.* the domain is a position if the condition is true and the domain is the inverse of a position if the condition is false. In other words, its application onto a point p* induces a change of p*'s domain when the "IF" condition is observed and another change of p*'s domain when the "ELSE" condition is observed. In order to make a NewCondition constraint act like a logical implication — *i.e.* the application of this constraint is superfluous if the condition is false, there is no change of p*'s domain for any "ELSE" condition —, the NewCondition constraint can be constructed as follows:

NewCondition[p_0 ...] := OriginalCondition[p_0 ...] ∪ [p_0]

As a consequence, the domain of the NewCondition is the entire plane when the OriginalCondition is not observed, meaning that, in this case, the application of the NewCondition constraint is superfluous.

(2) The resulting domain of a condition constraint can be changed in order to make it equivalent to a position (p_A) when the condition is true and equivalent to another position (p_B) when the condition is false (figure 281):

NewCondition[p_A p_B ...] := OriginalCondition[p_A ...] ∩ ([p_A] ∪ [p_B])
where: $p_B \in$ \[p_A]

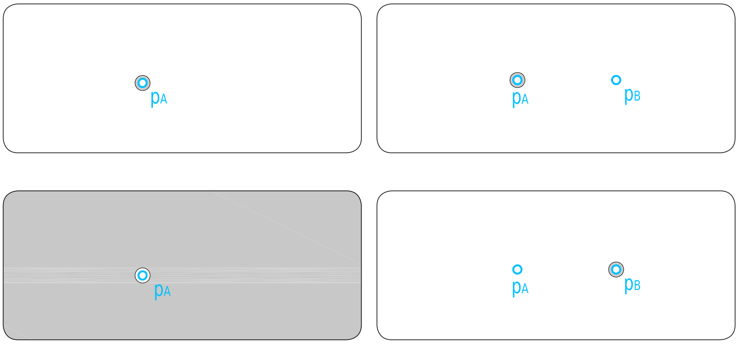


figure 281
(left) the domains of the OldCondition constraint and (right) the corresponding domains of the NewCondition constraint.

(3) If three points p_A, p_B and p_C define two perpendicular directions, the domain of the following NewCondition constraint can be a line parallel to p_Ap_B when the OriginalCondition constraint is satisfied and can be a line parallel to p_Ap_C when the OriginalCondition constraint is not satisfied (figure 282):

NewCondition[p_A p_B p_C ...] = VeeringStraightedge[p_A p_D]
where: $p_D \in$ OriginalCondition[p_B ...] ∩ ([p_B] ∪ [p_C])

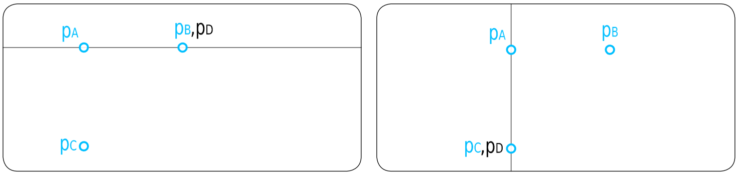


figure 282
(left) the domain of the NewCondition constraint when the condition is met and (right) the domain of the NewCondition constraint when the condition is not met.

example of conditional construction · The following construction presents a simple application using dynamic conditional statements. Given to distances p_2p_3 and p_4p_5, a rectangle is defined in such a way that its longer side is always parallel to p_0p_1, regardless of whether p_2p_3 is greater than p_4p_5 or not (figure 283):

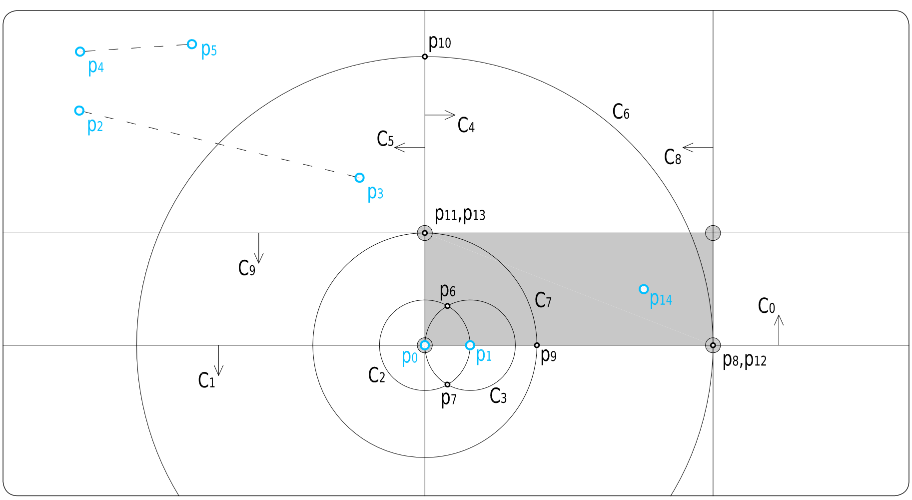
$C_0 = \text{VeeringHalfplane}[p_0\ p_1]$
$C_1 = \text{VeeringHalfplane}[p_1\ p_0]$
$C_2 = \text{CollapsibleCompass}[p_0\ p_1]$
$C_3 = \text{CollapsibleCompass}[p_1\ p_0]$
$p_6 \in C_0 \cap C_2 \cap C_3$
$p_7 \in C_1 \cap C_2 \cap C_3$
$C_4 = \text{Halfplane}[p_0\ p_6\ p_7]$
$C_5 = \text{Halfplane}[p_0\ p_7\ p_6]$

$C_6 = \text{Compass}[p_0\ p_2\ p_3]$
$C_7 = \text{Compass}[p_0\ p_4\ p_5]$
$p_8 \in C_0 \cap C_1 \cap C_6$
$p_9 \in C_0 \cap C_1 \cap C_7$
$p_{10} \in C_4 \cap C_5 \cap C_6$
$p_{11} \in C_4 \cap C_5 \cap C_7$

$p_{12} \in \text{ProximityCondition}[p_9\ p_2\ p_3\ p_4\ p_5] \cap ([p_8] \cup [p_9])$
$p_{13} \in \text{ProximityCondition}[p_{10}\ p_2\ p_3\ p_4\ p_5] \cap ([p_{10}] \cup [p_{11}])$

$C_8 = \text{Halfplane}[p_{12}\ p_7\ p_6]$
$C_9 = \text{Halfplane}[p_{13}\ p_1\ p_0]$
$p_{14} \in C_0 \cap C_4 \cap C_8 \cap C_9$

figure 283
a rectangle whose longest side is always horizontal.



21 constraints for a uniform reading cycle of forces

This sub-section details which constraints should be applied on each point defining a force in order to ensure that forces applied clockwise on the same point in the form diagram are always read in the same order in the corresponding force polygon in the force diagram. These constraints consequently guarantee the verification of axiom Ax.27 (page 109) — see sub-section 13 ("uniform reading cycle", page 107).

avoiding interdependency · Axiom Ax.27 can be implemented in different manners. The most straightforward is to constrain the point that defines the type of the force — *i.e.* the point p_2 of a Force[F_0 p_0 p_2 p_4 p_5] relationship — and let the propagation apply equivalent constraints onto the points that define the force in the force diagram — *i.e.* points p_4 and p_5 of a Force[F_0 p_0 p_2 p_4 p_5] relationship. However, this implementation always involves interdependency.

This can be understood with the basic example in figure 284. Forces F_0, F_1 and F_2 are all applied on point p_0. According to Ax.27, forces F_0, F_1 and F_2 must be read clockwise in this order in the form diagram — *i.e.* p_2 must stay in the wedge defined by the half-lines p_0p_1 and p_0p_3. In terms of constraints, Ax.27 is written as follows when F_0 and F_2 are not zero forces:

$$\begin{aligned}\text{ReadingCycleDom}[p_2] = \; & (\text{VeeringHalfplane}[p_1\ p_0] \cap \text{VeeringHalfplane}[p_0\ p_B]) \\ & \cup (\text{VeeringHalfplane}[p_1\ p_0] \cap \text{VeeringHalfplane}[p_0\ p_3]) \\ & \cup (\text{VeeringHalfplane}[p_A\ p_0] \cap \text{VeeringHalfplane}[p_0\ p_3])\end{aligned}$$

$$\begin{aligned}\textit{where: } p_A \in \; & \text{VeeringStraightedge}[p_0\ p_1] \cap \text{CollapsibleCompass}[p_0\ p_1] \\ & \cap \backslash\text{VeeringHalfPlane}[p_3\ p_0] \\ p_B \in \; & \text{VeeringStraightedge}[p_0\ p_3] \cap \text{CollapsibleCompass}[p_0\ p_3] \\ & \cap \backslash\text{VeeringHalfPlane}[p_0\ p_1]\end{aligned}$$

Using automated methods of constraint propagation, this sector would be propagated onto the domains of p_4 and p_5 since the orientations p_0p_2 and p_4p_5 are linked by Ax.19 (page 83):

$$\text{ForceDom}[p_2] = \text{Straightedge}[p_0\ p_4\ p_5] \cap \text{CoincidenceCondition}[p_4\ p_5\ p_0]$$

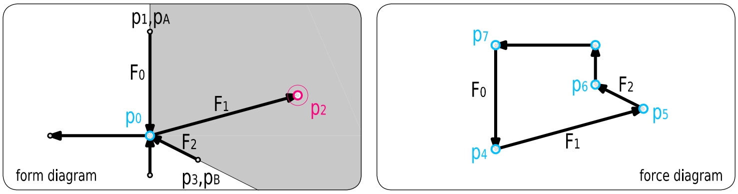


figure 284
the domain that p_2 must hold in order to guarantee Ax.27.

This propagation would make p_4 and p_5 dependent of p_2. However, in order to rotate F_1, either p_4 or p_5 must move. This means that, in most cases, F_0 or F_2 would rotate as well. As a result, either p_1 or p_3 — *i.e.* the points that define the type of application of F_0 and F_2 — would be moved, which would modify the constraints applied on p_2. A loop of dependencies consequently occurs.

This interdependency can be avoided if the uniform reading cycle requirement is defined specifically for each point that defines the force. This approach is chosen. For each Force[F_i p_A p_B p_C p_D] relationship, four ReadingCycleDom domains have to be constructed. The one concerning p_A is actually equal to the entire plane Ω and is therefore superfluous. The second — *i.e.* ReadingCycleDom[p_B] — is the direct application of Ax.27 already encountered — see figure 284. The third — *i.e.* ReadingCycleDom[p_2] — is obtained by describing the geometric conditions that p_C fulfils when Ax.27 is simultaneously applied on the previous and the next force that p_C links in the force diagram. The fourth— *i.e.* ReadingCycleDom[p_D] — is equivalent to the third regarding the subsequent force. As a consequence, the reading cycle domain of p_C is actually the only one that requires study.

It should be noted that, since Ax.27 is applied explicitly on the four points defining the force, their reading cycle domain should not be propagated onto themselves. It must only be propagated onto father points others than p_A, p_B, p_C and p_D.

construction of the reading cycle domain · Given four subsequent forces F_0, F_1, F_2, F_3:

Force[F_0 p_0 p_1 p_5 p_6]
Force[F_1 p_0 p_2 p_6 p_7]
Force[F_2 p_0 p_3 p_7 p_8]
Force[F_3 p_0 p_4 p_8 p_9]

and an undefined resultant R between F_3 and F_0 (figure 285 or figure 286), this paragraph defines the symbolic construction of the reading cycle domain of point p_7 in such a way that it remains valid whatever the positions of points

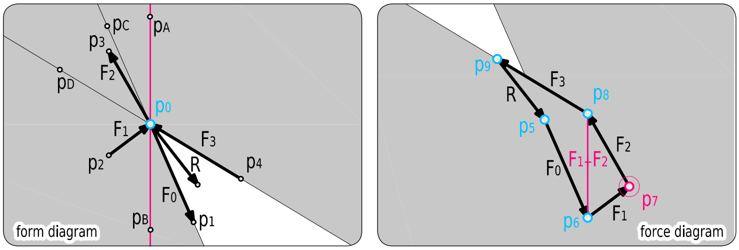


figure 285
(right) the domain that p_7 must hold in order to ensure that the cycle {F_0, F_1, F_2, F_3, R} is read clockwise in the form diagram.

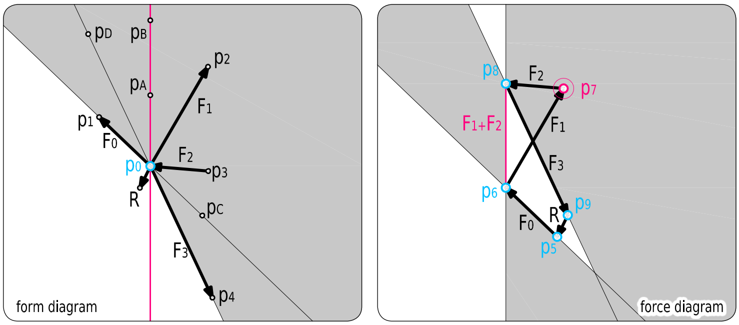


figure 286
five other forces; (right) the domain that p_7 must hold in order to ensure that the cycle {F_0, F_1, F_2, F_3, R} is read clockwise in the form diagram.

— *i.e.* whatever the new orientations, types of application, and magnitudes of forces. Therefore, the ReadingCycleDom does not have to be rebuilt when a point moves. It only has to be rebuilt when one of the four forces — F_0, F_1, F_2 or F_3 — is resolved or cancelled.

- When point p_7 moves, the orientations and magnitudes of F_1 and F_2 vary, meaning that Ax.27 must be satisfied for F_1 and F_2. But, instead of guaranteeing the cycles $\circlearrowright[F_0\ F_1\ F_2] \wedge \circlearrowright[F_1\ F_2\ F_3]$ separately (because it produces interdependency), Ax.27 is here satisfied with guaranteeing the cycles $\circlearrowright[F_0\ F_1\ F_3] \wedge \circlearrowright[F_0\ F_2\ F_3] \wedge \circlearrowright[F_1\ F_2]$, in other words, F_1 and F_2 are both read clockwise after F_0 and before F_3 and F_1 is read clockwise before F_2. These three conditions are developed individually and the resulting constraints are finally intersected.

- A constant that is of great help for this development is the axis of the resultant F_1+F_2. It only depends on p_6 and p_8 and is hence independent of p_7.

- In order to characterise the orientation of F_0 and F_3 with respect to the resultant F_1+F_2, points p_A and p_B are constructed as follows:

$$p_A \in SE[p_0\ p_6\ p_8] \cap VHP[p_0\ p_4] \cap CC[p_0\ p_1] \cap (\backslash[p_1] \cup [p_0])$$
$$p_B \in (SE[p_0\ p_6\ p_8] \cap VSE[p_0\ p_A]) \cap VHP[p_1\ p_0] \cap CC[p_0\ p_4] \cap (\backslash[p_4] \cup [p_0])$$

The role of these constraints is as follows.

Because of $SE[p_0\ p_6\ p_8]$ and $(SE[p_0\ p_6\ p_8] \cap VSE[p_0\ p_A])$, p_A and p_B are always positioned on the axis of the resultant F_1+F_2 (passing through p_0). And p_A and p_B always form on a line passing through p_0, even if p_6 and p_8 are coincident — *i.e.* if the resultant F_1+F_2 is zero.

Because of $VHP[p_1\ p_0]$, F_0 is always on the left of the direction going from p_0 to p_B. Because of $VHP[p_0\ p_4]$, F_3 is always on the left of the direction going from p_A to p_0.

Because of $CC[p_0\ p_1] \cap (\backslash[p_1] \cup [p_0])$, p_A and p_0 are coincident when F_0 is a zero force. Because of $CC[p_0\ p_4] \cap (\backslash[p_4] \cup [p_0])$, p_B and p_0 are coincident when F_3 is a zero force.

Because of $\backslash[p_1]$, p_A is on the opposite side of p_1 with respect to p_0 when F_0 is parallel with the axis of F_1+F_2. Because of $\backslash[p_4]$, p_B is on the opposite side of p_4 with respect to p_0 when F_3 is parallel with the axis of F_1+F_2.

There may be more than one position available for both p_A and p_B. The choice for one or another does not have any effect on the final result.

- The angle $\sphericalangle[F_0\ F_3]$ may be acute or obtuse, meaning that it would be sometimes described by an intersection of two half-planes and sometimes by a union of two half-planes. This would cause an issue since different Boolean combinations would be required by different positions of points. In order to describe it always with the same Boolean combination, the angle $\sphericalangle[F_0\ F_3]$ is here divided as a sum of two angles that are always either acute or straight. This is done by constructing points p_C and p_D as follows:

$$p_C \in (VHP[p_1\ p_0] \cap VHP[p_0\ p_4]) \cap VSE[p_0\ p_1] \cap CC[p_0\ p_1] \cap (\backslash VSE[p_0\ p_A] \cup [p_A])$$
$$p_D \in (VHP[p_1\ p_0] \cap VHP[p_0\ p_4]) \cap VSE[p_0\ p_4] \cap CC[p_0\ p_4] \cap (\backslash VSE[p_0\ p_B] \cup [p_B])$$

The role of these constraints is as follows.

Because of $(VHP[p_1\ p_0] \cap VHP[p_0\ p_4])$, p_C and p_D always belong to the angle $\sphericalangle\{F_0\ F_3\}$.

Because of $VSE[p_0\ p_1]$, p_C is on the axis of F_0. Because of $VSE[p_0\ p_4]$, p_D is on the axis of F_3.

Because of $CC[p_0\ p_1]$, p_C and p_0 are coincident when F_0 is a zero force. Because of $CC[p_0\ p_4]$, p_C and p_D are coincident when F_3 is a zero force.

Because of $(\backslash VSE[p_0\ p_A] \cup [p_A])$, p_C is on the opposite side of p_1 with respect to p_0 when F_0 is parallel with the axis of F_1+F_2. Because of $(\backslash VSE[p_0\ p_B] \cup [p_B])$, p_D is on the opposite side of p_4 with respect to p_0 when F_3 is parallel with the axis of F_1+F_2.

As a result, the angle $\sphericalangle[F_0\ F_3]$ is always equivalent to $\sphericalangle[F_0\ p_0p_C]+\sphericalangle[p_0p_C\ F_3]$ and to $\sphericalangle[F_0\ p_0p_D]+\sphericalangle[p_0p_D\ F_3]$, whatever the positions of p_1 and p_4.

- As a consequence, $\circlearrowright\{F_0\ F_1\ F_3\}$ is verified — *i.e.* F_0, F_1 and F_3 are read clockwise in that order — if F_1 either belongs to $\sphericalangle[F_0\ p_0p_C]$ or to $\sphericalangle[p_0p_C\ F_3]$. If F_1 exerts a pull, this means that:

$$p_7 \in (HP[p_6\ p_1\ p_0] \cap HP[p_6\ p_0\ p_C]) \cup (HP[p_6\ p_C\ p_0] \cap HP[p_6\ p_0\ p_4])$$

If F_1 exerts a push, this means that:

$$p_7 \in (HP[p_6\ p_0\ p_1] \cap HP[p_6\ p_C\ p_0]) \cup (HP[p_6\ p_0\ p_C] \cap HP[p_6\ p_4\ p_0])$$

- Also, $\circlearrowright[F_0\ F_2\ F_3]$ is verified — *i.e.* F_0, F_2 and F_3 are read clockwise in that order — if F_2 either belongs to $\sphericalangle[F_0\ p_0p_D]$ or to $\sphericalangle[p_0p_D\ F_3]$. If F_2 exerts a pull, this means that:

$$p_7 \in (HP[p_8\ p_0\ p_1] \cap HP[p_8\ p_D\ p_0]) \cup (HP[p_8\ p_0\ p_D] \cap HP[p_8\ p_4\ p_0])$$

If F_2 exerts a push, this means that:

$$p_7 \in (HP[p_8\ p_1\ p_0] \cap HP[p_8\ p_0\ p_D]) \cup (HP[p_8\ p_D\ p_0] \cap HP[p_8\ p_0\ p_4])$$

- Finally, in order to verify $\circlearrowright[F_1\ F_2]$ — *i.e.* F_1 is read clockwise before F_2 —, two cases must be distinguished: (a) the types of application of F_1 and F_2 are equivalent — *i.e.* F_1 and F_2 both exert either a push or a pull —; or (b) the types of application of F_1 and F_2 are different — *i.e.* either F_1 exerts a pull while F_2 exerts a push or vice versa.

(a) If F_1 and F_2 both have the same type of application, they have the geometric property to be always on opposite sides of the axis F_1+F_2 in the form diagram (figure 287 and figure 288).

figure 287
Both F_1 and F_2 exert a pull.

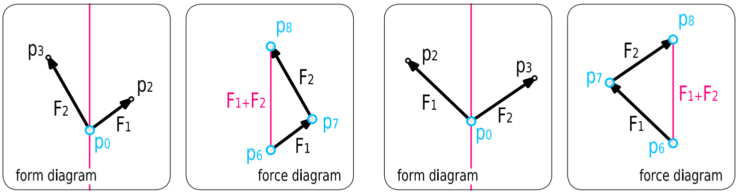
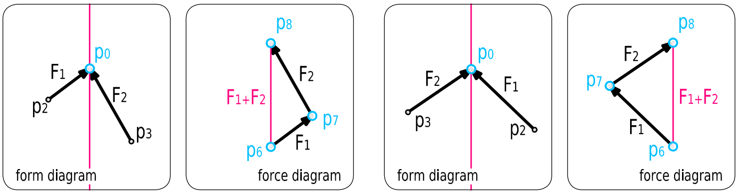


figure 288
Both F_1 and F_2 exert a push.



Therefore, if the angle $\sphericalangle[F_0\ F_3]$ is divided in three acute angles $\sphericalangle[F_0\ p_0p_B]$ + $\sphericalangle[p_0p_B\ p_0p_A]$ + $\sphericalangle[p_0p_A\ F_3]$, F_1 is clockwise before F_2 only if:

$$((F_1 \in \sphericalangle[F_0\ p_0p_B]) \wedge (F_2 \in \sphericalangle[p_0p_B\ p_0p_A]))$$
$$\vee ((F_1 \in \sphericalangle[p_0p_B\ p_0p_A]) \wedge (F_2 \in \sphericalangle[p_0p_A\ F_3]))$$

And this is equivalent to the following check:

$$(F_1 \notin \sphericalangle]p_0p_A\ F_3]) \wedge (F_2 \notin \sphericalangle[F_0\ p_0p_B[)$$

where F_1 may be parallel to p_0p_A but not to F_3, and F_2 may be parallel to p_0p_B but not to F_0. As a result, if F_1 and F_2 exert a pull, point p_7 must satisfy the following constraint:

$$p_7 \in \backslash(\backslash HP[p_6\ p_0\ p_A] \cap HP[p_6\ p_0\ p_4]) \cap \backslash(HP[p_8\ p_0\ p_1] \cap \backslash HP[p_8\ p_0\ p_B])$$

$$\longleftrightarrow p_7 \in (HP[p_6\ p_0\ p_A] \cup \backslash HP[p_6\ p_0\ p_4]) \cap (\backslash HP[p_8\ p_0\ p_1] \cup HP[p_8\ p_0\ p_B])$$

And if F_1 and F_2 exert a push, point p_7 must satisfy the following constraint:

$$p_7 \in \backslash(\backslash HP[p_6\ p_A\ p_0] \cap HP[p_6\ p_4\ p_0]) \cap \backslash(HP[p_8\ p_1\ p_0] \cap \backslash HP[p_8\ p_B\ p_0])$$

$$\longleftrightarrow p_7 \in (HP[p_6\ p_A\ p_0] \cup \backslash HP[p_6\ p_4\ p_0]) \cap (\backslash HP[p_8\ p_1\ p_0] \cup HP[p_8\ p_B\ p_0])$$

(b) If, on the other hand, F_1 and F_2 both have a different type of application, they have the geometric property to be always on the same side of the axis F_1+F_2 in the form diagram (figure 289 and figure 290).

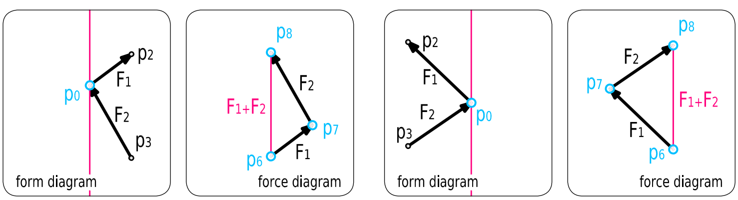


figure 289
F_1 exerts a pull and F_2 a push.

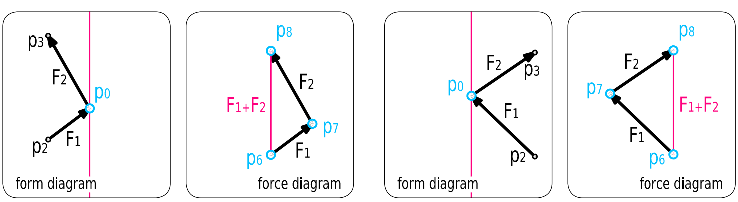


figure 290
F_1 exerts a push and F_2 a pull.

Consequently, if F_1 exerts a pull and F_2 exerts a push, F_1 is always clockwise before F_2 if:

- $p_7 \in VHP[p_8\, p_6]$ except in the case where the cycles $\circlearrowright[p_0 p_A\, F_1\, F_3[\wedge \circlearrowright]F_0\, F_2\, p_0 p_B]$ exist (figure 291) — *i.e.* except when $p_7 \in (Hp[p_6\, p_A\, p_0] \cap \backslash Hp[p_6\, p_4\, p_0]) \cap (\backslash Hp[p_8\, p_0\, p_1] \cap Hp[p_8\, p_0\, p_B])$

or · $p_7 \in VHP[p_6\, p_8]$ and the cycles $\circlearrowright[p_0 p_A\, F_2\, F_3] \wedge \circlearrowright[F_0\, F_1\, p_0 p_B]$ exist (figure 292) — *i.e.* $p_7 \in (Hp[p_8\, p_A\, p_0] \cap Hp[p_8\, p_0\, p_4]) \cap (Hp[p_6\, p_1\, p_0] \cap Hp[p_6\, p_0\, p_B])$.

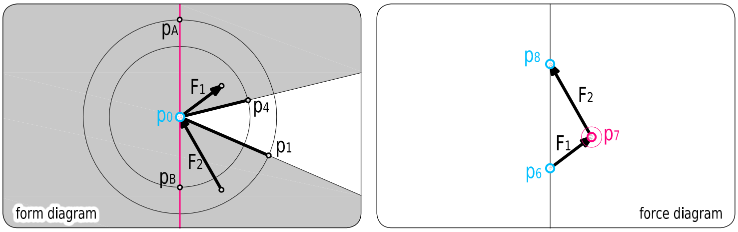


figure 291
Impossible case when F_1 pulls and F_2 pushes.

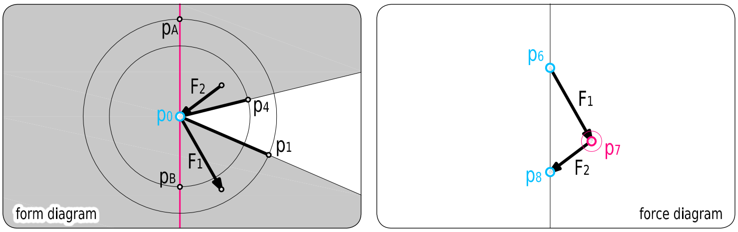


figure 292
Additional case when F_1 pulls and F_2 pushes.

Putting it all together, if F_1 exerts a pull and F_2 exerts a push, F_1 is always clockwise before F_2 if:

$$p_7 \in \big(VHP[p_8\, p_6] \cap \backslash(Hp[p_6\, p_A\, p_0] \cap \backslash Hp[p_6\, p_4\, p_0] \cap \backslash Hp[p_8\, p_0\, p_1] \cap Hp[p_8\, p_0\, p_B]) \big) \cup \big(VHP[p_6\, p_8] \cap (Hp[p_6\, p_1\, p_0] \cap Hp[p_6\, p_0\, p_B] \cap Hp[p_8\, p_A\, p_0] \cap Hp[p_8\, p_0\, p_4]) \big)$$

$$\longleftrightarrow p_7 \in \big(VHP[p_8\, p_6] \cap (\backslash Hp[p_6\, p_A\, p_0] \cup Hp[p_6\, p_4\, p_0] \cup Hp[p_8\, p_0\, p_1] \cup \backslash Hp[p_8\, p_0\, p_B]) \big) \cup \big(VHP[p_6\, p_8] \cap (Hp[p_6\, p_1\, p_0] \cap Hp[p_6\, p_0\, p_B] \cap Hp[p_8\, p_A\, p_0] \cap Hp[p_8\, p_0\, p_4]) \big)$$

Similarly, if F_1 exerts a push and F_2 exerts a pull, F_1 is always clockwise before F_2 if:

$$p_7 \in \big(VHP[p_6\, p_8] \cap (\backslash Hp[p_6\, p_0\, p_A] \cup Hp[p_6\, p_0\, p_4] \cup Hp[p_8\, p_1\, p_0] \cup \backslash Hp[p_8\, p_B\, p_0]) \big) \cup \big(VHP[p_8\, p_6] \cap (Hp[p_6\, p_0\, p_1] \cap Hp[p_6\, p_B\, p_0] \cap Hp[p_8\, p_0\, p_A] \cap Hp[p_8\, p_4\, p_0]) \big)$$

The two particular cases are illustrated in figure 293 and figure 294.

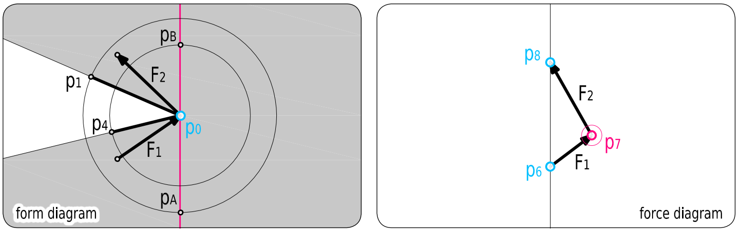


figure 293
Impossible case when F_1 pushes and F_2 pulls.

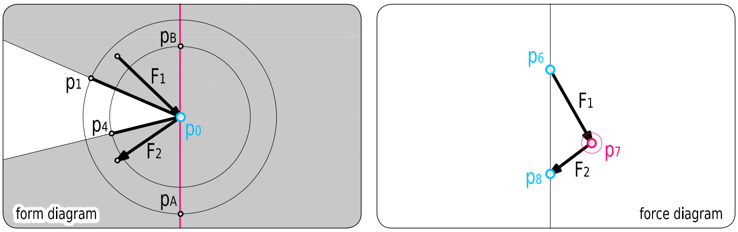


figure 294
Additional case when F_1 pushes and F_2 pulls.

- In order for the main condition $\circlearrowright[F_0\ F_1\ F_2\ F_3]$ to remain true regardless of whether F_1 and F_2 exert a pull or a push, there are four sub-conditions:

$$\circlearrowright[F_0\ F_1\ F_3] \wedge \circlearrowright[F_0\ F_2\ F_3] \wedge \circlearrowright[F_1\ F_2]$$

$$\begin{aligned}
\longleftrightarrow\ & (\circlearrowright[F_0\ F_1\ F_3] \wedge \circlearrowright[F_0\ F_2\ F_3] \wedge \circlearrowright[F_1\ F_2]\ \textit{when both } F_1 \textit{ and } F_2 \textit{ exert a pull})\\
\vee\ & (\circlearrowright[F_0\ F_1\ F_3] \wedge \circlearrowright[F_0\ F_2\ F_3] \wedge \circlearrowright[F_1\ F_2]\ \textit{when both } F_1 \textit{ and } F_2 \textit{ exert a push})\\
\vee\ & (\circlearrowright[F_0\ F_1\ F_3] \wedge \circlearrowright[F_0\ F_2\ F_3] \wedge \circlearrowright[F_1\ F_2]\ \textit{when } F_1 \textit{ exerts a pull and } F_2 \textit{ exerts a push})\\
\vee\ & (\circlearrowright[F_0\ F_1\ F_3] \wedge \circlearrowright[F_0\ F_2\ F_3] \wedge \circlearrowright[F_1\ F_2]\ \textit{when } F_1 \textit{ exerts a push and } F_2 \textit{ exerts a pull})
\end{aligned}$$

This is equivalent to apply a constraint of the following type onto p_7:

$$\begin{aligned}
\longleftrightarrow p_7 \in\ & (C_{1,pull} \cap C_{2,pull} \cap C_{pull\text{-}pull})\\
\cup\ & (C_{1,push} \cap C_{2,push} \cap C_{push\text{-}push})\\
\cup\ & (C_{1,pull} \cap C_{2,push} \cap C_{pull\text{-}push})\\
\cup\ & (C_{1,push} \cap C_{2,pull} \cap C_{push\text{-}pull})
\end{aligned}$$

- The recapitulation of all the constraints to be applied on p_7 is consequently the following one. If F_0, F_1, F_2, F_3 are four subsequent forces in the force diagram such that:

$$\begin{aligned}
F_0 &= \text{Force}[p_0\ p_1\ p_5\ p_6]\\
F_1 &= \text{Force}[p_0\ p_2\ p_6\ p_7]\\
F_2 &= \text{Force}[p_0\ p_3\ p_7\ p_8]\\
F_3 &= \text{Force}[p_0\ p_4\ p_8\ p_9]
\end{aligned}$$

than, point p_7 must belong to the following constraint in order to guarantee Ax.27 — *i.e.* in order to ensure that F_0, F_1, F_2 and F_3 are read clockwise in that order in the form diagram —:

$$
\begin{aligned}
\mathrm{ReadingCycleDom}[p_7] = {} & (\, C_{1,\mathrm{pull}} \cap C_{2,\mathrm{pull}} \cap C_{\mathrm{pull\text{-}pull}} \,) \\
\cup\ & (\, C_{1,\mathrm{push}} \cap C_{2,\mathrm{push}} \cap C_{\mathrm{push\text{-}push}} \,) \\
\cup\ & (\, C_{1,\mathrm{pull}} \cap C_{2,\mathrm{push}} \cap C_{\mathrm{pull\text{-}push}} \,) \\
\cup\ & (\, C_{1,\mathrm{push}} \cap C_{2,\mathrm{pull}} \cap C_{\mathrm{push\text{-}pull}} \,)
\end{aligned}
$$

where:

$$
\begin{aligned}
& p_A \in \mathrm{SE}[p_0\ p_6\ p_8] && \cap\ \mathrm{VHP}[p_0\ p_4] \cap \mathrm{CC}[p_0\ p_1] \cap (\backslash[p_1] \cup [p_0]) \\
& p_B \in (\mathrm{SE}[p_0\ p_6\ p_8] \cap \mathrm{VSE}[p_0\ p_A]) && \cap\ \mathrm{VHP}[p_1\ p_0] \cap \mathrm{CC}[p_0\ p_4] \cap (\backslash[p_4] \cup [p_0]) \\
& p_C \in (\mathrm{VHP}[p_1\ p_0] \cap \mathrm{VHP}[p_0\ p_4]) && \cap\ \mathrm{VSE}[p_0\ p_1] \cap \mathrm{CC}[p_0\ p_1] \cap (\backslash\mathrm{VSE}[p_0\ p_A] \cup [p_A]) \\
& p_D \in (\mathrm{VHP}[p_1\ p_0] \cap \mathrm{VHP}[p_0\ p_4]) && \cap\ \mathrm{VSE}[p_0\ p_4] \cap \mathrm{CC}[p_0\ p_4] \cap (\backslash\mathrm{VSE}[p_0\ p_B] \cup [p_B])
\end{aligned}
$$

$$
\begin{aligned}
C_{1,\mathrm{pull}} &= (\mathrm{HP}[p_6\ p_1\ p_0] \cap \mathrm{HP}[p_6\ p_0\ p_C]) \cup (\mathrm{HP}[p_6\ p_C\ p_0] \cap \mathrm{HP}[p_6\ p_0\ p_4]) \\
C_{1,\mathrm{push}} &= (\mathrm{HP}[p_6\ p_0\ p_1] \cap \mathrm{HP}[p_6\ p_C\ p_0]) \cup (\mathrm{HP}[p_6\ p_0\ p_C] \cap \mathrm{HP}[p_6\ p_4\ p_0])
\end{aligned}
$$

$$
\begin{aligned}
C_{2,\mathrm{pull}} &= (\mathrm{HP}[p_8\ p_0\ p_1] \cap \mathrm{HP}[p_8\ p_D\ p_0]) \cup (\mathrm{HP}[p_8\ p_0\ p_D] \cap \mathrm{HP}[p_8\ p_4\ p_0]) \\
C_{2,\mathrm{push}} &= (\mathrm{HP}[p_8\ p_1\ p_0] \cap \mathrm{HP}[p_8\ p_0\ p_D]) \cup (\mathrm{HP}[p_8\ p_D\ p_0] \cap \mathrm{HP}[p_8\ p_0\ p_4])
\end{aligned}
$$

$$
\begin{aligned}
C_{\mathrm{pull,pull}} &= (\mathrm{HP}[p_6\ p_0\ p_A] \cup \backslash\mathrm{HP}[p_6\ p_0\ p_4]) \cap (\backslash\mathrm{HP}[p_8\ p_0\ p_1] \cup \mathrm{HP}[p_8\ p_0\ p_B]) \\
C_{\mathrm{push,push}} &= (\mathrm{HP}[p_6\ p_A\ p_0] \cup \backslash\mathrm{HP}[p_6\ p_4\ p_0]) \cap (\backslash\mathrm{HP}[p_8\ p_1\ p_0] \cup \mathrm{HP}[p_8\ p_B\ p_0])
\end{aligned}
$$

$$
\begin{aligned}
C_{\mathrm{pull,push}} = {} & (\ \mathrm{VHP}[p_8\ p_6] \\
& \quad \cap\ (\backslash\mathrm{Hp}[p_6\ p_A\ p_0] \cup \mathrm{Hp}[p_6\ p_4\ p_0] \cup \mathrm{Hp}[p_8\ p_0\ p_1] \cup \backslash\mathrm{Hp}[p_8\ p_0\ p_B])\ ) \\
\cup\ & (\ \mathrm{VHP}[p_6\ p_8] \\
& \quad \cap\ (\mathrm{Hp}[p_6\ p_1\ p_0] \cap \mathrm{Hp}[p_6\ p_0\ p_B] \cap \mathrm{Hp}[p_8\ p_A\ p_0] \cap \mathrm{Hp}[p_8\ p_0\ p_4])\ )
\end{aligned}
$$

$$
\begin{aligned}
C_{\mathrm{push,pull}} = {} & (\ \mathrm{VHP}[p_6\ p_8] \\
& \quad \cap\ (\backslash\mathrm{Hp}[p_6\ p_0\ p_A] \cup \mathrm{Hp}[p_6\ p_0\ p_4] \cup \mathrm{Hp}[p_8\ p_1\ p_0] \cup \backslash\mathrm{Hp}[p_8\ p_B\ p_0])\ ) \\
\cup\ & (\ \mathrm{VHP}[p_8\ p_6] \\
& \quad \cap\ (\mathrm{Hp}[p_6\ p_0\ p_1] \cap \mathrm{Hp}[p_6\ p_B\ p_0] \cap \mathrm{Hp}[p_8\ p_0\ p_A] \cap \mathrm{Hp}[p_8\ p_4\ p_0])\ )
\end{aligned}
$$

Synthesized diagrams of p_7's resulting domains are shown figure 295 for various positions of p_1, p_4, p_6 and p_8. Regions in the force diagram are divided in the four following sub-regions:

- I F_1 pulls and F_2 pulls
- II F_1 pushes and F_2 pushes
- III F_1 pulls and F_2 pushes
- IV F_1 pushes and F_2 pulls

In certain positions, F_1 or F_2 is allowed to push or pull. The choice for one type of application or another depends on the position of p_2, respectively p_3.

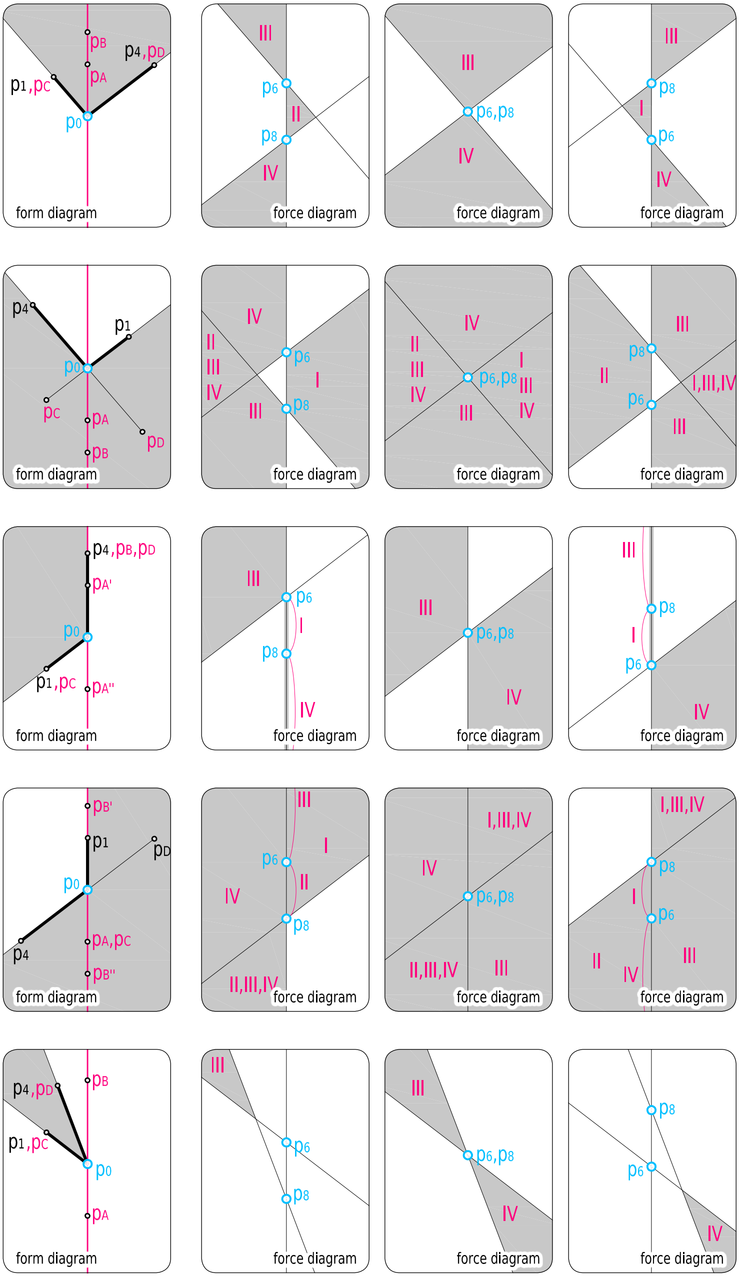
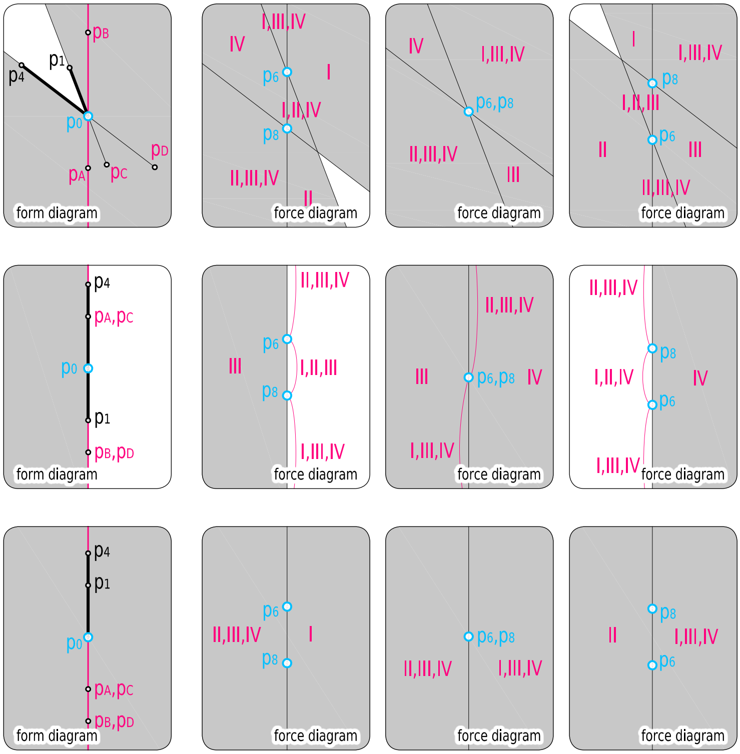


figure 295
domain of p_7 for various positions of p_1, p_4, p_6 and p_8, such that F_1 and F_2 are read clockwise in the right order between F_0 and F_4.



constraints for pushing forces or pulling forces · Similar constructions can be used to compel a force to be pushing or pulling. For instance, if the force F_0 in figure 296 must remain pushing, the point p_0 (instead of point p_7) must be constrained so that either {F_4 pushes and F_0 pushes} or {F_4 pulls and F_0 pushes} and the point p_1 (instead of point p_7) must be constrained so that either {F_4 pushes and F_0 pushes} or {F_4 pulls and F_0 pushes}. In other words:

$$p_0 \in \quad (C_{4,push} \cap C_{0,push} \cap C_{push\text{-}push}) \cup (C_{4,pull} \cap C_{0,push} \cap C_{pull\text{-}push})$$

$$p_1 \in \quad (C_{0,push} \cap C_{1,push} \cap C_{push\text{-}push}) \cup (C_{0,push} \cap C_{1,pull} \cap C_{push\text{-}pull})$$

These constraints are identical to those constructed in the previous paragraph except that all the references to the points are shifted so that p_7 is replaced by p_0 and p_1 successively. The intersection of these constraints with the reading cycle domains are shown in figure 296 and figure 297.

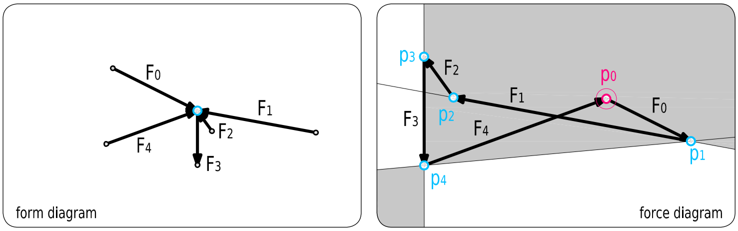


figure 296
the domain that p_0 must hold in order for F_0 to be pushing.

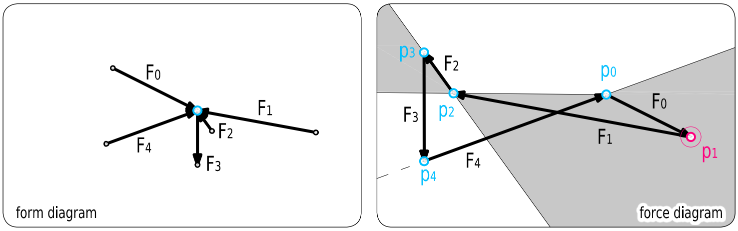


figure 297
the domain that p_1 must hold in order for F_0 to be pushing.

constraints for compression rods or traction rods · A rod always in compression or in traction is simply obtained by compelling the two forces defining it to be either pushing or pulling. For instance, figure 298 shows the domain that p_1 must hold in order for the rod R_0 to remain in compression.

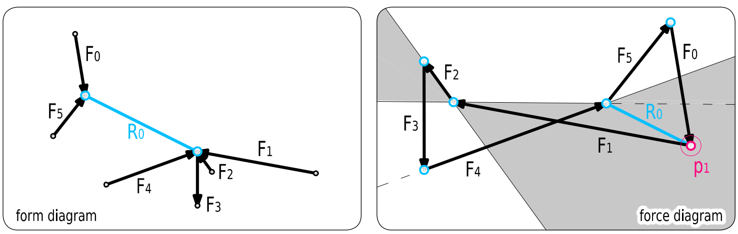
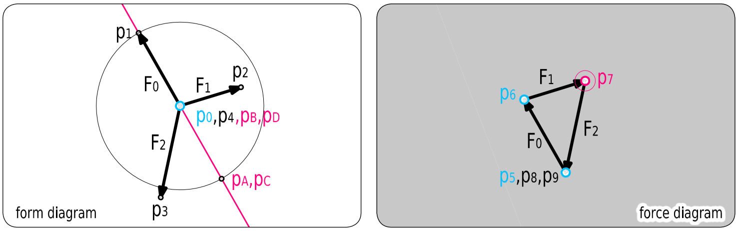


figure 298
the domain that p_1 must hold in order for R_0 to remain in compression.

particular behaviours of the reading cycle domain · The ReadingCycleDom mentioned above remains valid for any number of forces applied on p_0 as long as zero forces are not taken into account, meaning that the ReadingCycleDom must be recalculated when a force becomes a zero force.

However, the construction of the domain remains valid when only two forces are applied on p_0 for the following reason. If F_1 and F_2 are the only two forces applied on p_0, it means that F_0 and F_3 are zero forces and hence, that p_5, p_6, p_8, p_9 are all coincident and that p_0, p_1, p_4 are also coincident. As a result, p_A, p_0, p_B, p_C and p_D are coincident, meaning that the constraints $C_{1,pull}$, $C_{1,push}$, $C_{2,pull}$, $C_{2,push}$, $C_{pull,pull}$, $C_{push,push}$, $C_{push,pull}$ and $C_{pull,push}$ are all equal to the entire plane Ω. The ReadingCycleDomain of p_7 is consequently equal to the entire plane Ω, which is correct.

figure 299
the domain that p_7 must hold when only three forces are applied on p_0.



The same is true when only three forces are applied on p_0. If, for instance, F_3 is a zero force, points p_5, p_8 and p_9 are coincident and p_0 and p_4 are also coincident (figure 299). As a consequence, p_A, p_B, p_C and p_D are constrained as follows:

$$p_A \in SE[p_0\ p_6\ p_8] \cap CC[p_0\ p_1] \cap (\backslash[p_1] \cup [p_0])$$
$$p_B \in [p_0]$$
$$p_C \in VSE[p_0\ p_1] \cap CC[p_0\ p_1] \cap (\backslash VSE[p_0\ p_A] \cup [p_A]$$
$$p_D \in [p_0]$$

$$C_{1,pull} = HP[p_6\ p_1\ p_0] \cup HP[p_6\ p_C\ p_0]$$
$$C_{1,push} = HP[p_6\ p_0\ p_1] \cup HP[p_6\ p_0\ p_C]$$
$$C_{2,pull} = \Omega$$
$$C_{2,push} = \Omega$$

$$C_{pull,pull} = HP[p_6\ p_0\ p_A]$$
$$C_{push,push} = HP[p_6\ p_A\ p_0]$$
$$C_{pull,push} = VHP[p_5\ p_6] \cup (VHP[p_6\ p_5] \cap HP[p_6\ p_1\ p_0] \cap HP[p_6\ p_A\ p_0] = VHP[p_5\ p_6]$$
$$C_{push,pull} = VHP[p_5\ p_6] \cup (VHP[p_6\ p_5] \cap HP[p_6\ p_0\ p_1] \cap HP[p_6\ p_0\ p_A] = VHP[p_5\ p_6]$$

As a result p_7 can be anywhere in the plane, which is correct:

$$ReadingCycleDom[p_7] = HP[p_6\ p_0\ p_A] \cup HP[p_6\ p_A\ p_0] \cup VHP[p_5\ p_6] \cup VHP[p_5\ p_6] = \Omega$$

when a point belongs to multiple force polygons · When a point p_7 belongs to multiple force polygons in the force diagram, the global ReadingCycleDom$[p_7]$ is the intersection of all the local ReadingCycleDom$[p_7]$ associated with each force polygon. This is mostly due to the fact that, if there is no application of geometric constraint other than those defining forces and rods, a point in the force diagram modifies the orientation and magnitude of only two forces for each point of application in the space diagram (figure 300). When other geometric constraints are applied on the points in the force diagram, they would take Ax.27 into account by propagation of each concerned reading cycle domain — see sub-section 19 (“constraint propagations”, page 201).

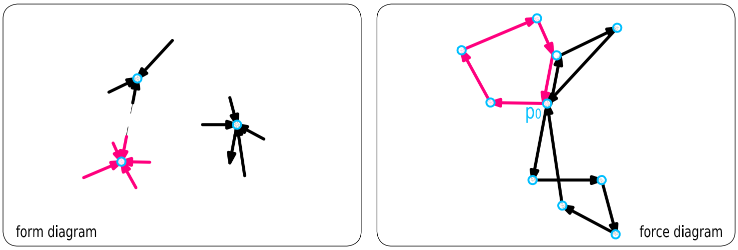


figure 300
a point p_0 that belong to multiple force polygons.

allowing the modification of reading cycle domains · To compel each point of the force diagram inside its ReadingCycleDom results in preventing any modification of reading cycle. However, changing the reading cycles of forces may be useful for the user. Two approaches are available to change the reading cycles of forces.

(1) The first approach prevents any point from being outside its ReadingCycleDom. If the user wants to modify the current reading cycle of forces, he has to modify the equilibrium explicitly: he has to (a) move points in the force diagram in order to cancel the force whose order in the cycle must be changed and (b) make it reappear at the desired position. This approach is detrimental to the fluidity with which users handle strut-and-tie networks. The following approach is better in this respect.

(2) The second approach allows each point to be outside its ReadingCycleDom. As soon as one point is dragged outside its ReadingCycleDom, the network is automatically modified so that the new reading cycle matches the new position. f the point that is dragged is in the form diagram, the change only concerns the force whose type of application is defined by the point. If the point that is dragged is in the force diagram, the change concerns the two forces that it links.

In both cases, the change of order in the force polygon requires the coincidence of two points and the duplication of another. An example is shown in figure 301 and figure 303 : p_2 and p_3 become coincident and p_5 is duplicated into p_5 and p_7.

figure 301
the force polygon prior to the change of reading cycle.

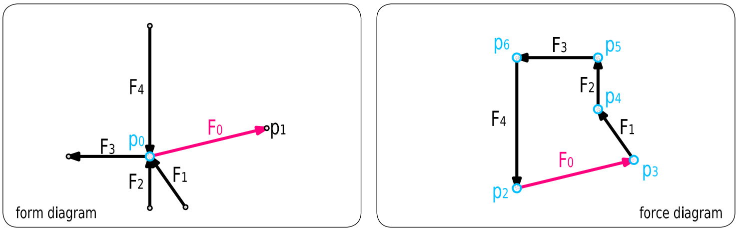


figure 302
the force polygon "during" the change of reading cycle.

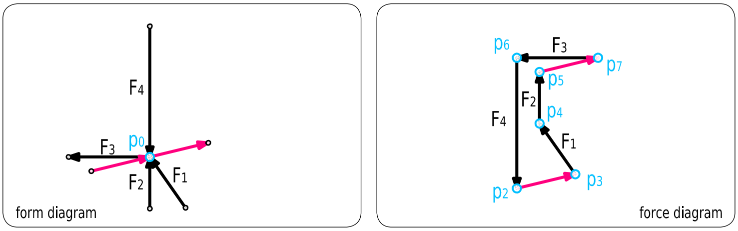
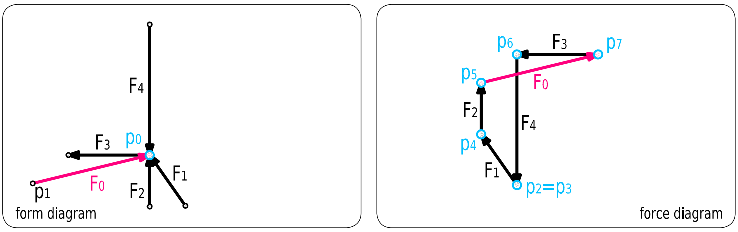


figure 303
the force polygon after the change of reading cycle.



It may be that other constraints were applied explicitly onto p_2 and p_3 before the change of order. A choice will therefore have to be made as to whether these constraints must be deleted, kept or copied. Since there is no reason to delete them or to apply them all (with parameters p_5 and p_7 in place of p_2 and p_3), the most natural choice seems to be the following one:

- if the constraint is always verified because of the coincidence p_2=p_3, if it is always identical to another because of the coincidence p_2=p_3 or if p_2 and p_3 are two parameters of the constraint, the constraint is applied after changing p_2 by p_5 and p_3 by p_7
- if not, the constraint is kept with the original parameters p_2 and p_3.

For instance, if a constraint DiscInside[p_B p_2 p_3] was applied onto a point p_A prior to the change of order in the force polygon, it would be replaced by a constraint DiscInside[p_B p_5 p_7] applied onto p_A.

22 facilitating the crossing of rods

It has been shown in the paragraph entitled "why a uniform reading cycle imposes the absence of almost any intersection of rods in the space diagram" (page 124) that each crossing of two rods and each crossing of a rod with a half-line of force in the form diagram should be replaced by a new point after dividing the concerned rods and duplicating the concerned force.

Since moving a point (in the form diagram or in the force diagram) can change the orientation of rods and forces in the form diagram, it may happen that this move requires the division of a rod (or its cancellation) or the duplication of a force (or its cancellation). This sub-section describes a systematic method to update rods and forces when a point moves and changes their orientation and position.

the topological domain · First, the topological domain is defined such that any move of a point p_0 inside its TopologicalDomain[p_0] does not create new crossing of rods and forces and does not cancel other crossings of rods and forces in the form diagram. Accordingly, if a point in the form diagram or in the force diagram is dragged and causes the move of other points in the form diagram, it is known that, if each of these points is not moved outside its corresponding topological domain, no update of crossing of rods and forces must be performed. As a result, it is sufficient to construct the topological domain in the form diagram only.

If a point p_0 in the form diagram links n rods $\{R_0, R_1, ..., R_{n-1}\}$, its TopologicalDomain is the intersection of n sub-domains, each of these sub-domains being equal to the reading cycle domain of the point that defines the type of application of the force that is at the other extreme of the rod R_i ($i \in [0,n-1]$) — see sub-section 21 ("constraints for a uniform reading cycle of forces", page 243). For instance, the topological domain of p_0 in figure 304 is equal to the intersection of the three reading cycle domains that are shown in figure 305.

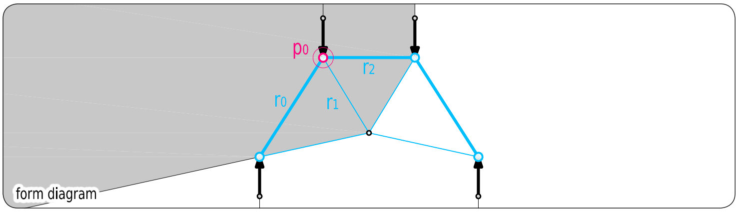


figure 304
in grey, the topological domain of p_0.

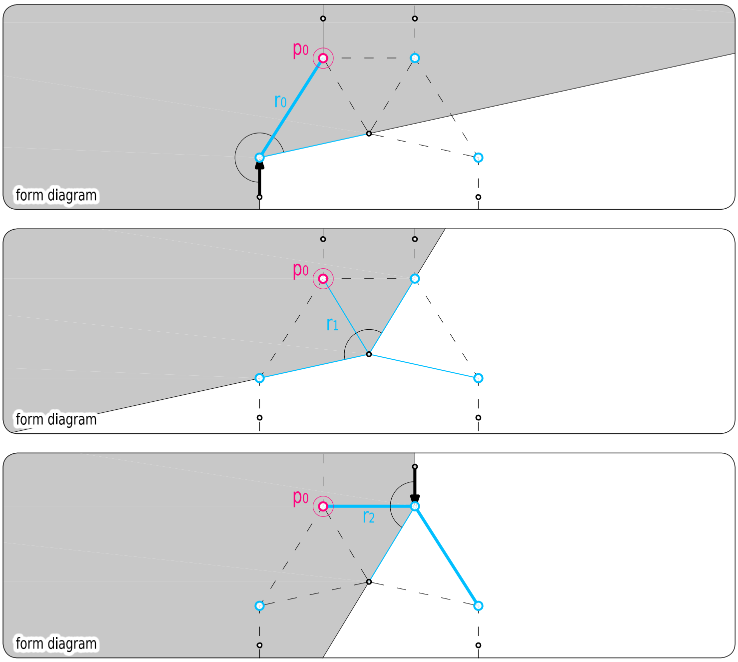


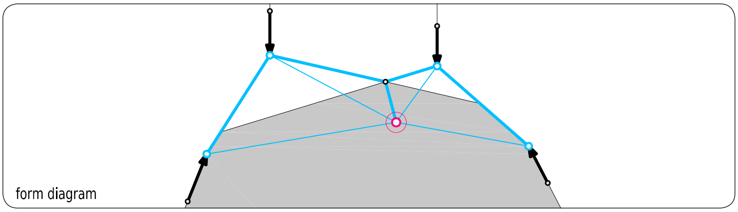
figure 305
in grey, the three topological sub-domains of p_0; each sub-domain corresponds to the reading cycle domain of p_0 around an opposite point that is connected to p_0 by a rod or a half-line of force.

Another example of topological domain is shown in figure 306.

Because the points at the other extreme of the rods may be dependent of the point p_0 whose topological domain is sought, the construction of the topological domain of p_0 may include interdependency. This means that the displayed topological domain is not equivalent to the actual set of allowed positions. Hopefully this disadvantage is acceptable for two reasons.

Firstly, most of these interdependencies are superfluous — *i.e.* the constraints that cause an interdependency belong to the same geometric relationships — and can be avoided using permutations of constraints — see the paragraph entitled “automatic deletion of superfluous interdependencies” (page 200).

figure 306
another example of topological domain.



Secondly, the topological domain is not aimed to be displayed or intersected with other domains. Its unique role is to alert when a point is dragged outside its topological domain. As long as a point p_0 belongs to its TopologicalDomain[p_0], it does not matter whether the boundaries of the TopologicalDomain[p_0] move simultaneously along with p_0 or not.

change of the reading cycle of rods · When a point p_0 is dragged outside its topological domain onto a position p*, the following operations must be performed:

- transform each rod pointing on p_0 into a couple of opposite forces and temporarily cancel the constraints that compel each couple of forces to be parallel and of equal magnitude
- move p_0 and the forces applied on it onto the position p*
- rotate each force that was part of a rod and, when appropriate, update the force polygon in order to satisfy each reading cycle of forces, using the recommendations of the paragraph entitled "allowing the modification of reading cycle domains" (page 256)
- for each couple of forces: move points in the force diagram so that, if two forces of a same couple do not cross other forces, they can be transformed into a rod;
- for each intersection of forces, add a point at the intersection and apply new forces on it. These forces have the same magnitude as the intersecting forces. New and old opposite forces form new couples of opposite forces
- transform each couple of forces into a rod

A simple application of this process is illustrated on the following figures.

Since no input from the user is required in this sequence, it can be performed automatically as soon as a point p_0 is dragged onto a position p* beyond its topological domain. This sequence updates the form diagram in such a way that no constraint of the input domain is cancelled or added.

If the update of the reading cycles is not allowed because of some constraints of the input domain, the move of p_0 onto p^* must be prevented.

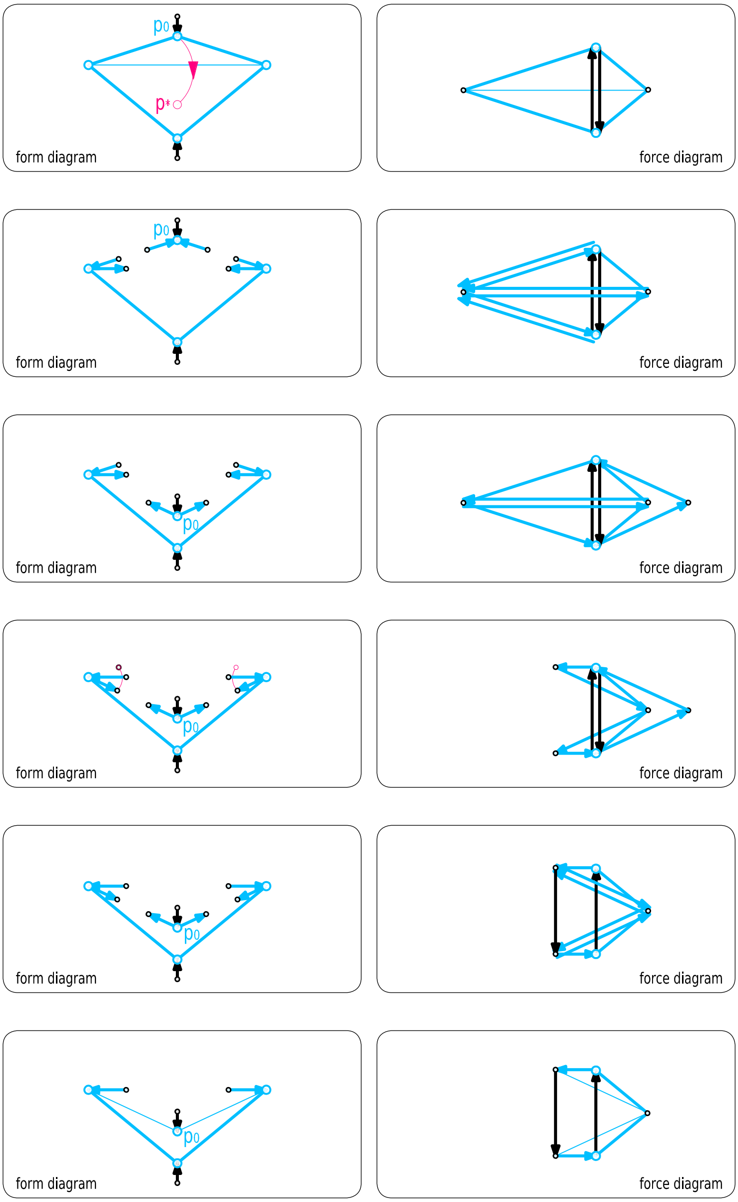
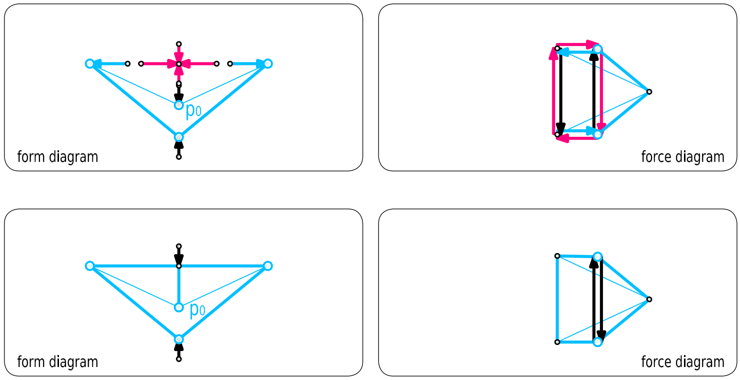


figure 307
point p_0 is dragged onto p^*; description of eight successive steps that can be automated in order to update each reading cycle of forces and each crossing of rods.


p0
form diagram
force diagram
p0
form diagram
force diagram

PRODUCTION RULES FOR COMPUTER-AIDED GRAPHIC STATICS

This section develops specifications for the dynamic construction and modification of any strut-and-tie networks in static equilibrium within two reciprocal diagrams.

Sub-section 23 ("objects and native operations", page 267) defines the minimum set of operations required allow the user to build or modify a strut-and-tie network. As summarized in sub-section 24 ("higher-order procedures", page 273), these native operations can be assembled in a procedural manner in order to execute more complex geometric and structural routines.

Sub-section 25 ("functional flow", page 281) recapitulates the functional flow of the proposed computer-aided environment and briefly describes some user-interfaces.

23 objects and native operations

imperative coding · The goal of this sub-section is to identify a minimum set of operations capable of transforming any strut-and-tie network in equilibrium into another one. If algorithms exist to process these basic operations, they can be assembled in sequence in order to define and hence perform more complex operations. Some of these more complex operations will be presented in sub-section 24 ("higher-order procedures", page 273).

The entire sequence of operations forms the "*construction plan*". A construction plan is therefore a sufficient description of the strut-and-tie network and its geometric description.

Before presenting the minimum set of native operations, data types are listed in the following paragraph.

data types · Besides the set of data types natively supported in usual programming languages — *e.g.* numbers, Booleans and arrays —, the following data types should exist:

- Point — it is defined, at a minimum, by the following parameters:
  - the diagram to which it belongs
  - its current position in that diagram
  - the global intersection of constraints that are applied on it by the user
- Constraint — it is defined, at a minimum, by
  - its type: HalfPlane, DiscInside, DiscOutside, Union, Intersection or Inversion
  - if it is a fundamental constraint: the three points that define the region of the constraint
  - if it is not: the set of constraints to be unite, intersect or invert
- Force — it is defined, at a minimum, by

- the four points defining it: the point of application in the form diagram, the point defining its type of application in the form diagram and two points defining its magnitude in the force diagram

- Rod — it is defined, at a minimum, by
  - the two opposite forces that are equivalent to it

Moreover, a form diagram and a force diagram should be identified. The reading cycle of forces in the form diagram must be set either clockwise or anticlockwise.

minimum set of native operations · In order to ease the implementation of the native operations, their function is limited as far as possible and the objects to be processed must comply with strict requirements. Unlike the higher-order procedures that will be describe in the next sub-section, they are not aimed to be intuitive for the practitioner.

Besides the common native operations supported in programming languages — *e.g.* assignments, arithmetic, comparisons, definitions and execution of subroutines, loops and static conditional statements (not to be confused with the dynamic conditional statements, page 233) — the following native operations should exist:

- CreatePointInTheFormDiagram[]
  - creates a new point that belongs to the form diagram; no constraint is applied on the point.
  - returns the point

- CreatePointInTheForceDiagram[]
  - creates a new point that belongs to the force diagram; no constraint is applied on the point
  - returns the point.

- DeletePoint[p_A]
  - removes the point p_A from the diagram
  - p_A must be a point on which no constraint and no force are applied.

- MovePoint[p_A coord]
  - moves the point p_A as close as possible to the position defined by the coordinates coord — see the paragraph entitled “allowing dynamic displacements of points” (page 143).

- MergePoint[p_A p_B]
  - replaces every occurrence of p_B by p_A
  - points p_A and p_B must be coincident
  - if p_B belongs to the form diagram, no force and no rod should be applied onto it.
- Halfplane[p_A p_B p_C]
  - creates a HalfPlane[p_A p_B p_C] constraint
  - returns the constraint.
- DiscInside[p_A p_B p_C]
  - creates a DiscInside[p_A p_B p_C] constraint
  - returns the constraint.
- DiscOutside[p_A p_B p_C]
  - creates a DiscOutside[p_A p_B p_C] constraint
  - returns the constraint.
- UnitCompass[p_A p_B p_C]
  - creates a DiscOutside[p_A p_B p_C] constraint
  - returns the constraint.
- Intersection[C_A C_B]
  - creates a constraint equivalent to the intersection of the constraint C_A with the constraint C_B
  - returns the intersection constraint.
- Union[C_A C_B]
  - creates a constraint equivalent to the union of the constraint C_A with the constraint C_B
  - returns the union constraint.
- Inversion[C_A]
  - creates a constraint equivalent to the inversion of the constraint C_A
  - returns the inversion constraint.
- DeleteConstraint[C_A]
  - deletes the constraint C_A
  - C_A can be any type of constraint
  - C_A should not be applied on any point and should not be part of any intersection, union or inversion constraint

- ApplyConstraint[p_A C_A]
  - intersects the constraint C_A with all the constraints already applied on point p_A;
  - if necessary, p_A is moved in order to belong to the region of C_A
  - if the application of C_A empties the domain of p_A, the application is not performed.

- CancelConstraint[p_A C_A]
  - removes the constraint C_A from all the constraints applied on point p_A;
  - C_A should be applied directly (not as part of a sub-intersection or union of constraint).

- SwitchDependencies[p_A p_B C_A]
  - switch the dependencies between p_A and p_B throughout the constraint C_A
  - the constraint C_A must be applied on p_A and p_B must be a direct parameter defining C_A
  - returns the newly created symmetrical constraint

- CreateZeroForce[p_A p_B]
  - creates a zero force that is applied on p_A in the form diagram and that is defined by p_B in the force diagram
  - p_A must belong to the form diagram and p_B to the force diagram
  - returns the force.

- DeleteZeroForce[F_A]
  - removes the zero force F_A
  - F_A must be a zero force — *i.e.* the two points defining its magnitude in the force diagram must coincide

- ResolveForce[F_A p_A p_B p_C]
  - resolves the force F_A into two forces; they are applied on the same point of application as F_A
  - p_A and p_B become the points defining the type of application of the two forces in the form diagram and are constrained as such — see the paragraph entitled “constraining forces” (page 153).
  - p_C is the point linking the two forces in the force diagram
  - the force F_A no longer exists after its resolution
  - returns the two forces in their order of presentation in the force diagram.

- SwapForceCycle[F_A F_B]
  - swap the order in which forces F_A and F_B are read in the form diagram and in the corresponding force polygon
  - F_A and F_B must be applied on the same point in the form diagram and they must be consecutive in the force diagram
  - F_A must be read before F_B in the form diagram and in the corresponding force polygon
  - no constraint should already be applied on any point that define F_A and F_B

- CreateRod[F_A F_B]
  - replaces the two forces F_A and F_B by a rod
  - the forces F_A and F_B should not be already part of a rod
  - the points defining F_A and F_B must be constrained in such a way that F_A and F_B are compatible for any position of points: F_A and F_B must always be parallel, of equal magnitude and opposite, and their point of application in the form diagram must be aligned with the forces
  - returns the rod.

- CancelRod[R_A]
  - replaces the rod R_A by the two forces that defines it

In addition to these, there should be operations:

- to get informations — *e.g.* to get the set of constraints applied on a certain point or to get the parameter points defining a certain fundamental constraint
- to perform checks — *e.g.* to check if a certain constraint is applied on a point or if a certain point is a child of another certain point
- to select objects — *e.g.* to select a set of points, strut-and-tie sub-network or force networks with the mouse
- to alter the appearance of objects — *e.g.* to hide or show a certain point.

24 higher-order procedures

developing intuitive and efficient routines · High-order procedures are meant to automate sequences of native operations. The amount of operations to be performed by these procedures is not limited and they are developed in order to be as intuitive and efficient for the user as possible.

Routines can serve purely geometric purposes — *e.g.* the computations detailed in sub-section 17 (“examples of graphical computations”, page 165) — vectorial purposes — *e.g.* the computation of the resultant of a set of forces — or structural purposes — *e.g.* the modification of the cross sections of a beam in order to minimise inner bending moments. Some routines are broad and others are specific to the structural model type being use.

Key general procedures to develop are those directly extending the scope of the native operations:

- to apply a Compass or a Straightedge constraint
- to apply a constraint on several points simultaneously
- to delete a constraint although it is applied on point
- to switch constraint dependencies between points that are not direct relatives
- to merge two points on which forces are applied
- to move points so that two forces are sufficiently compatible to be replaced by a rod
- to swap forces that are not direct neighbours in the force polygon
- …

Other procedures to develop are those implementing classical graphic statics methods:

- the construction of a funicular polyline passing through one, two or three given points

- the identification of the order in which intersecting forces shall be taken into considerations in order to produce a compression-only or a tension-only funicular polyline
- the computation of the centre of gravity of n given forces
- the computation of the resultant of n given forces
- …

Some procedures to develop are especially relevant for constraint-based graphic statics, they help the construction of the diagrams:

- the displacement of points so that two strut-and-tie sub-networks can be merged together
- the application of geometric boundaries to points and rods — *i.e.* to compel a rod to remain inside a given area, see next paragraph
- the imposition of conditions on inner stresses — *i.e.* to compel a set of rods to remain in compression, see the paragraphs entitled "constraints for pushing forces or pulling forces" (page 253) and "constraints for compression rods or traction rods" (page 254)
- the application of affine transformations — *e.g.* translation, scaling, rotation, shear mapping — to a given strut-and-tie network (see Huerta·2010 for practical applications)
- the creation of a symmetric copy of a strut-and-tie network
- the displacement of a point so that a rod in a strut-and-tie network passes through a given point
- …

The following paragraphs illustrate how to create two of these routines.

example: adding a constraint to prevent two line segments from intersecting each other · Many practical applications are strut-and-tie networks that are asked to remain a inside given area (figure 20, page 41 for instance). These applications greatly benefits from constraint-based graphic statics since these boundaries are automatically propagated on every domain of solutions associated to the strut-and-tie network.

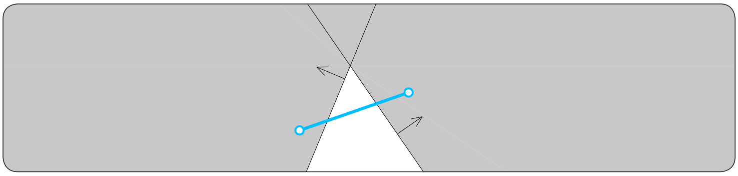


figure 308
a rod that is not totally inside a given region.

Though it is easy to constrain a point inside a given region by direct application of constraints, to constrain an entire rod inside a given region is less straightforward. For instance, figure 308 shows a rod that is not entirely inside a region although its extreme points are constrained inside that region. This issue can here be settled by applying constraints on both extreme points in order to prevent each rod from crossing each line segment that form the boundary of the given region.

Concretely, two line segments p_0p_1 and p_2p_3 (one of which can be a rod) never cross if the point p_0 is constrained as follows (figure 309):

p_0 ∈ VeeringHalfPlane[p_1 p_A] ∪ VeeringHalfPlane[p_B p_1] ∪ VeeringHalfPlane[p_B p_A]
where: p_M ∈ MidPoint[p_2 p_3]
p_A ∈ ([p_2] ∪ [p_3]) ∩ VeeringHalfPlane[p_1 p_M]
p_B ∈ ([p_2] ∪ [p_3]) ∩ VeeringHalfPlane[p_M p_1]
∩ (\VeeringStraightedge[p_1 p_A] ∪ [p_A])

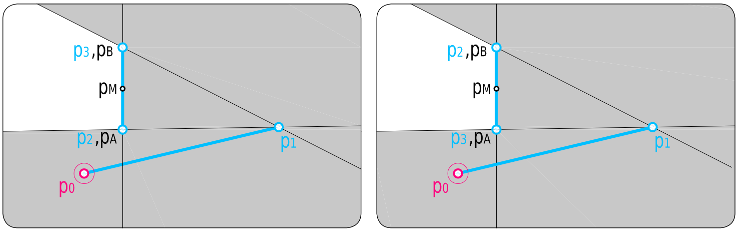


figure 309 the domain that p_0 must hold in order to prevent the intersection of the two line segments; two examples.

If p_1, p_2 and p_3 are aligned — regardless of whether p_1 is between p_2 and p_3 or not —, p_A and p_B would then be coincident and p_0 would belong to the entire plane, as expected. The same is true when p_2 and p_3 are coincident.

Similar constraints are automatically applied on points p_1, p_2 and p_3 by symmetry.

As a result, rods will never cross boundaries if the following procedure is performed for each potentially crossing pairs of line segments:

- NoLineSegmentsCross[p_0 p_1 p_2 p_3]
  - · applies a constraint on p_0 so that the line segments p_0p_1 and p_2p_3 never cross
  - · the four points p_0, p_1, p_2 and p_3 must belong to the same diagram
  - · returns the constraint that has been applied on p_0

NoLineSegmentsCross[p_0 p_1 p_2 p_3] = {

```
    if ¬(    (GetDiagram[p1] = GetDiagram[p2])
           ∧ (GetDiagram[p2] = GetDiagram[p3])
           ∧ (GetDiagram[p3] = GetDiagram[p4]) )
        {throw error "Points p1, p2, p3 and p4 do not belong to the same diagram"}

    else {
        current_diagram := GetDiagram[p0]

        pM := CreatePointInDiagram[current_diagram]
        ApplyConstraint[pM MidPoint[p2 p3] ]

        pA := CreatePointInDiagram[current_diagram]
        pB := CreatePointInDiagram[current_diagram]

        cA := Union[ Position[p2] Position[p3] ]
        cB := Intersection[ VeeringHalfPlane[p1 pM] cA ]
        cC := Intersection[ VeeringHalfPlane[pM p1] cA ]
        ApplyConstraint[pA cB]
        ApplyConstraint[pB cC]

        cD := Union[   VeeringHalfPlane[p1 pA]
                       VeeringHalfPlane[pB p1]
                       VeeringHalfPlane[pB pA] ]
        ApplyConstraint[p0 cD]

        return cD
    }
}
```

example: creating a funicular polyline passing through one given point · The following procedure creates a simple funicular polyline that supports any number of forces and whose first rod has to pass through a given point. Since a load is generally free of moving along a given line of action, the point of application of a load in the form diagram is usually already constrained on a straightedge. For instance, the point p_1 in figure 310 belong to a Straightedge[p_4 p_2 p_3] constraint. As a consequence, the creation of a funicular polyline is equivalent to constraining each point of force application so that its position is fixed on the corresponding line of action.

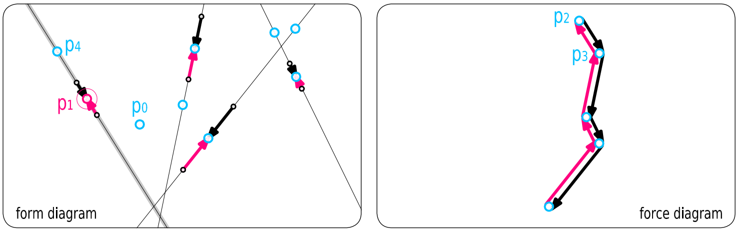


figure 310 initial situation before the execution of the CreateCatenary procedure; the domain of p_1 is highlighted.

The procedure uses the GetParam[object i] native operation that returns the point that is the i^th^ parameter (starting from 0) of the given object — where the object is a constraint, a force or a rod.

The procedure can be defined as follows:

- CreateSimplyConnectedNetwork[p_0 forces]
  - creates a funicular polyline that supports the forces listed in the array Forces and whose first rod passes through p_0
  - the forces are supported in the order in which they are referenced in the array Forces
  - the forces are assumed to be already consecutive in the force diagram
  - returns an array containing the rods that have been created

```
CreateCatenary[p0 forces] = {
    numberOfForces := ArrayLength[forces]

    — check whether the forces are consecutive in the force diagram or not:
    i := 0
    while i<numberOfForces-1 {
        if ¬(GetParam[forces[i] 2] = GetParam[forces[i+1] 3])
            {throw error "Forces "+i+" and "+(i+1)+" are not consecutive
                                                         in the force diagram"}
        i := i+1
    }

    — creation of the pole:
    pole := CreatePointInForceDiagram[]

    — arrays that will contains the new rays and the new rods:
    rays:= CreateArray[numberOfForces*2]
    rods:= CreateArray[numberOfForces-1]
    i := 0
    while i<numberOfForces {
        — resolution of the forces in order to form the rays (figure 311):
        tempRays := ResolveForce[forces[i] CreatePointInFormDiagram[]
                                            CreatePointInFormDiagram[] pole]
        rays[i*2] := tempRays[0]
        rays[i*2+1] := tempRays[1]

        — alignment of the points of application (figure 312):
        if (i=0) ∨ (i=1) {
            se := CreateStraightedge[p0 pA GetParam[rays[0] 2] ]
            ApplyConstraint[ GetParam[rays[i*2] 0] se]
        }
        else {
```

```
            se := CreateStraightedge[ GetParam[rays[(i-1)*2] 0] pA
                                                          GetParam[rays[(i-1)*2] 2] ]
            ApplyConstraint[ GetParam[rays[i*2] 0] se]
        }

        — creation of the rods (figure 313):
        if i>0 {
            rods[i-1] := CreateRod[rays[(i-1)*2] rays[(i*2)+1] ]
        }

        i := i+1
    }
    return rods
}
```

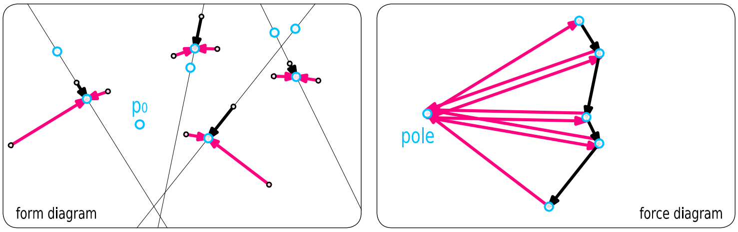


figure 311
creation of the pole and resolution of forces.

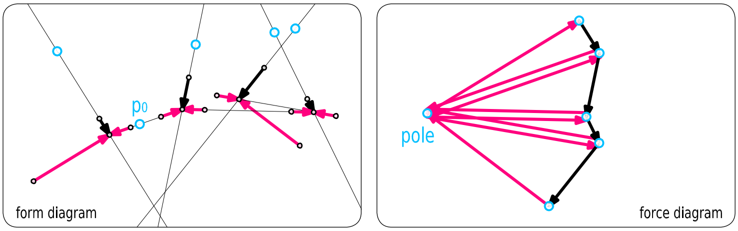


figure 312
alignment of the points of application.

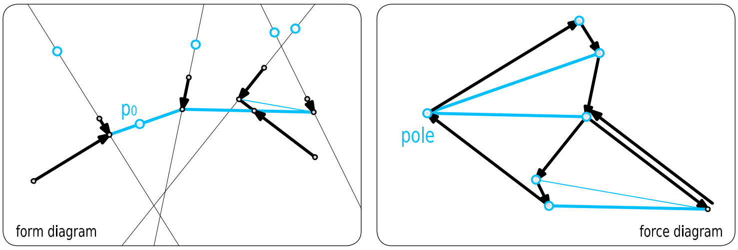


figure 313
creation of the rods.

In the example in figure 313, the force diagram has been automatically rearranged after the execution of the operation ResolveForce[forces[3]] because the newly resolved forces crossed the rod that has been created by the operation CreateRod[rays[2] rays[5]].

The CreateSimplyConnectedNetwork procedure should be flanked by similar others, *e.g.* to let the user select the rod through which the funicular polyline has to pass or to create funicular polylines that pass through two or three points.

25 functional flow

the symbolic solver and the numerical solver · Because the structural designer modifies his or her design step by step, the resolution of the geometric constraints is performed in a similar sequential manner — *i.e.* each new operation builds on the previous results. A rough functional flow diagram of the approach is presented in the following figure: boxes are computed data and arrows are algorithms or user's commands.

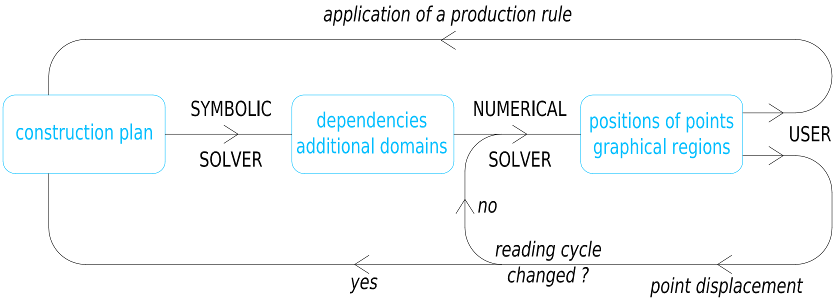


figure 314 functional flow diagram.

The construction plan holds a declarative list of all the operations applied by the designer to the strut-and-tie network. This plan is first analysed by a symbolic solver. It performs all the operations that can be made regardless of the actual positions of points. For instance, it identifies each constraint dependency and produces a symbolic description of each input and strict domain (page 151), each force domain (page 153), each reading cycle domain (page 243), each topological domain (page 259) and all the propagation domains that can be handled symbolically (page 201).

This first solver offers the advantage of not being executed after most point displacements.

The constraints created by this symbolic solver do not have the same service life as the constraints created by the user. The former become inefficient when there is a modification of the constraints applied on one of their parent points or one of the points on which their are applied. The latter remain efficient as long as the user does not delete them explicitly.

The update of these positions and hence of the actual shape of the domains of solutions is then undertaken by a numerical solver whose job is mainly to compute orthogonal projections on lines or circles — see sub-section 15 ("graphical regions and dynamic compliance with geometric relationships", page 137). The eventual numerical propagation domains should be performed by this solver too.

The geometric construction is displayed once the numerical solver has found the solution — *i.e.* once each point is inside a non-empty region. The user can then modify this result by moving points or applying new operations on the strut-and-tie network.

locale computations · If the operation modifies the construction plan the solving process must be rerun. Thanks to the fact that the positions of points are the only variables, their dependencies is fully expressed with directed graphs of dependencies — see sub-section 16 ("constraint (inter) dependencies", page 155). This means that the minimum set of points concerned by a given operation can be easily identified and that a local treatment of the data is sufficient for processing modifications in both the symbolic and numerical solvers.

automatic routines ·If a point is moved outside its reading cycle domain or outside its topological domain, routines can be executed automatically in order to update the reading cycle of forces and hence the force polygons — see the paragraph entitled "allowing the modification of reading cycle domains" (page 256) and the sub-section 22 ("facilitating the crossing of rods", page 259). These automatic routines apply operations that modify the construction plan and must therefore be computed by the symbolic solver again.

user-interfaces · Three types of user-interfaces can already be identified. The first type is the one that has been used in the previous sub-section. It consists of successive written declarations assembled into a script.

The second type is the one common to contemporary cad drawing: points are represented and dragged in a plane — here in a form diagram and in a force diagram — and operations on the construction plan are executed directly after they have been selected on button panels or entered in a command prompt.

The third type displays constraints and points as boxes in an interactive directed graph of dependencies. Connecting these boxes together allows the creation and application of geometric constraints, in a similar way to grasshopper components (Payne/Issa·2009 and Khabazi·2010).

The first type seems to be better suited to the definition of new higher-order procedures while the second and the third types appear more adequate to jointly build strut-and-tie networks interactively.

DISCUSSION

This final section discusses the results of the tool that has been developed throughout the previous sections. First sub-section 26 ("applications", page 287) produces an overview of the large set of expected applications and exemplifies some of them. Future research approaches are then proposed in sub-section 27 ("future research", page 305). A general conclusion concerning this thesis is finally drawn in sub-section 28 ("conclusions", page 315).

26 applications

The tool is expected to aid many different approaches to structural design and analysis. The following paragraphs present a general overview of all the intended applications. Examples of practical applications are subsequently explained.

who ? · Generally speaking, constraint-based graphic statics are intended to assist structural designers, be they structural or civil engineers, architects, industrial designers, preservation engineers, students and others.

when? · As outlined in sub-section 04 ("proposal: a tool to accompany the construction of static equilibriums", page 29), constraint-based graphic statics seem adequate for use in the following contexts:

- initial design explorations of structural shapes, especially when new typologies are sought (to suit new materials or new ways of production)
- cross-professional design team meetings, since drawings — and, for instance, graphical depictions of shapes and forces — constitute a common medium for architects, engineers and clients; it is not only suited to passive exchanges of information between remote design teams, but also to direct dialogues thanks to the generalisation of new portable computing devices such as touchpads
- amendments of structural geometries following advanced structural analysis, since strut-and-tie models are appropriate for efficiently modifying structural behaviours (Zastavni·2010)
- preservation assessments when the stability of existing structures is better understood in a graphical environment — *i.e.* for structures with a plastic behaviour (such as reinforced concrete) in which identified defects can be considered as new geometric requirements (Fivet·2012) and for masonry structures where stability is essentially a geometric matter (Huerta·2006b and Loits·2010).

- the teaching of architecture students and civil engineering students, since graphic statics have always been a fruitful pedagogic tool (Allen·2009):

“[Using graphic statics]*, we should have a course of Engineering Mechanics, so invigorating to the mind, that our students, having undergone its discipline, would feel themselves men, well prepared for work, capable of appreciating the conditions, and reasoning upon the data, of large class of practical questions to which they might require to address themselves*”*.* (Chalmers·1881, preface page vii)

what ? · Strut-and-tie networks are very well suited for use as abstractions of the stability of a very wide range of structures, as long as their utilisation can be described in the plane (which does not impede the design of most spatial structures):

- statically determinate and indeterminate reticular structures, including pre/post-stressed structures and self-stressed structures — *e.g.* roof structures, frames, bridges, tensegrities, reciprocal frames
- mechanisms, linkages (Herrmann·1892, Kempe·1877 and McCarthy/...·2011) — *e.g.* movable bridges
- thrust lines within compression-only structures — *e.g.* masonry arches
- load paths within materials showing a plastic behaviour (Ochsendorf·2005) — *e.g.* reinforced concrete bridges
- Euler-Bernoulli beams subjected to bending — *e.g.* hyperstatic beams, pre-stressed beams — and columns, including graphical computations of deformations, second-order effects and studies using *Mohr*'s circle for controlling moments of inertia (Pirard·1950, page 157) and for controlling basic deflections (Heyman·2008b, page 28)
- discontinuous stress-fields within plastic materials — *e.g.* reinforced concrete shear-walls — including studies using *Mohr*'s circle for controlling stress states (Fivet·2012) — see sub-section 27 (“future research”, page 305) for more details
- stabilising slopes, retaining walls and foundations — *i.e.* soil mechanics (Terzaghi·1966).

These applications therefore consider almost any building material — *e.g.* steel, wood, concrete, glass, ceramics, polymers, including their combinations into composites, *e.g.* reinforced wood (Trautz/Koj·2009a and Trautz/Koj·2009b). Moreover, although the original subject of this research deals solely with architectural structures, constraint-based graphic statics also appears convenient for the design of naval structures, biomechanics, robotics, furniture and industrial design.

how ? · The following paragraphs exemplify some applications. The first example uses constraint-based graphic statics to reconstruct the reticular shed built in Chiasso by Robert Maillart in 1924. The second example shows how geometrical constraints can be used to control bending moments explicitly. The third example explains how static indeterminacy in reticular structures can be managed graphically. The final example assesses the stability of a masonry arch.

(1) constructing a reticular structure · In order to illustrate how a reticular structure can be shaped with constraint-based graphic statics, this paragraph recreates the Chiasso sheds designed by Robert Maillart in 1924.

The initial settings considered here are those identified in Zastavni·2008a (page 290 and following) and Zastavni·2008b: the symmetrical peaked roof consists of a 10 centimetres wide concrete slab; its slope and its position is mainly given by the size of trains and the openings and slabs of the adjoining building (figure 315); the axial distance between vertical members is determined at 260 centimetres (half the distance between columns of the building adjacent to the shed); and the buildable volume is mainly given by storage requirements and the geometry of the adjoining building.

The following reconstruction is quite different from the one described in Zastavni·2008a and Zastavni·2008b and from Maillart's original design process. The main reason is that the diagrams will be here in static equilibrium at each single step. The goal here is not to find a way to “close” the force polygon — *i.e.* to ensure that the final shape of the shed is in equilibrium. The goal here is to modify an initial force polygon already closed — *i.e.* to alter a shape already in equilibrium — until:

- the only remaining forces are the loads applied on the structure or the reactions at the supports
- the stresses in each element are less than their maximum strengths
- the overall structural system is sufficiently stable.

First, the buildable volume is defined as a Boolean combination of HalfPlane constraints that are dependent on a set of free points, a Straightedge constraint symbolizes the roof (figure 316). A first load case is then deduced from the

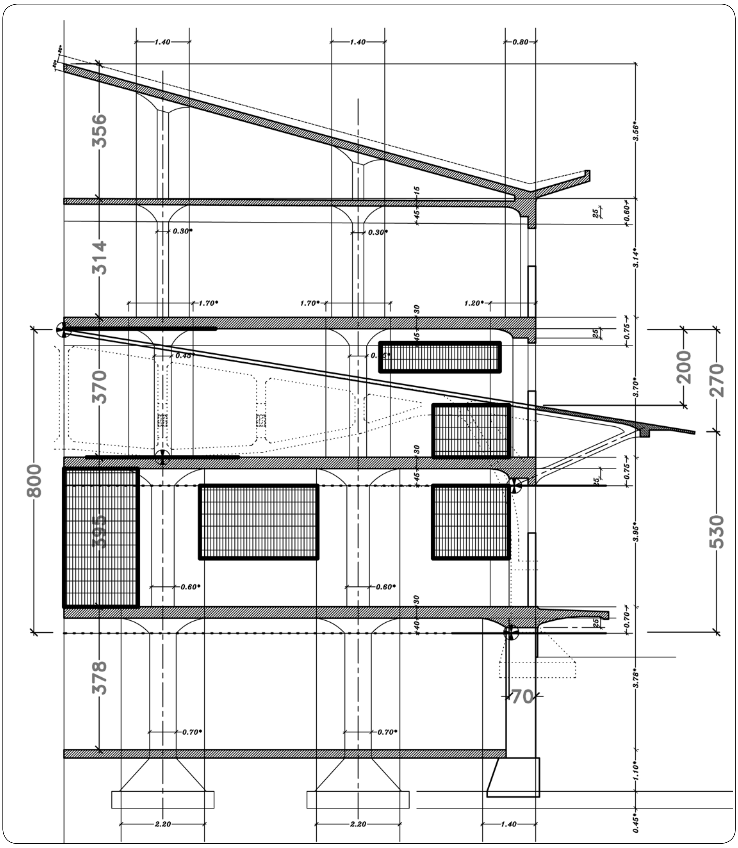


figure 315
influence of the adjoining building on the geometry of the shed, from Zastavni·2008a.

initial properties of the roof and is discretized with equilibrated pairs of forces (figure 317). These pairs of forces are constrained on the axes given on figure 316.

This system is already in equilibrium. The role of the designer is now to focus on what operations should be applied to transform these initial internal forces into rods.

For instance, he can apply new equilibrated pairs of forces on each point of the roof. These new forces being of equal magnitudes and parallel to the axis of the roof, they can be transformed into rods in which new axial stresses can flow (figure 318). In order to get rid of forces F_A, F_B and F_C, the designer creates a new simply connected strut-and-tie network, *i.e.* a funicular polyline, that takes up equal but opposite forces (figure 319). This polyline can then be merged with the original network while transforming corresponding forces into rods (figure 320).

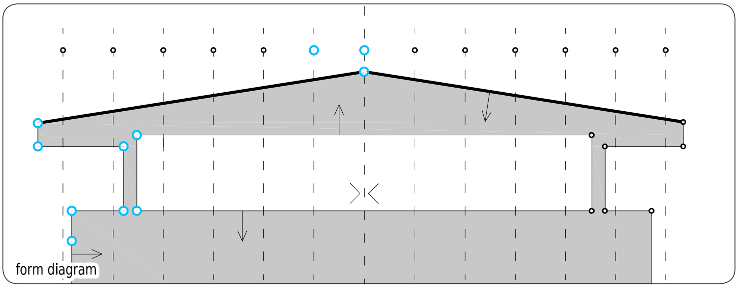


figure 316
reconstruction of Maillart's Chiasso sheds (snapshot 0); initial buildable volume.

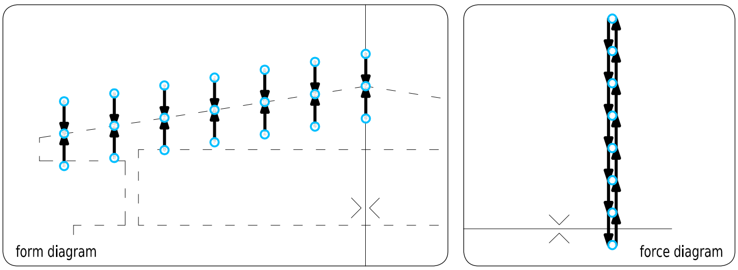


figure 317
reconstruction of Maillart's Chiasso sheds (snapshot 1); initial set of loads.

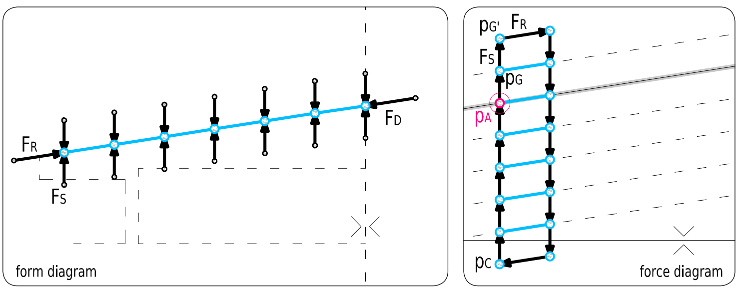


figure 318
reconstruction of Maillart's Chiasso sheds (snapshot 2); the linear domain of p_A is highlighted.

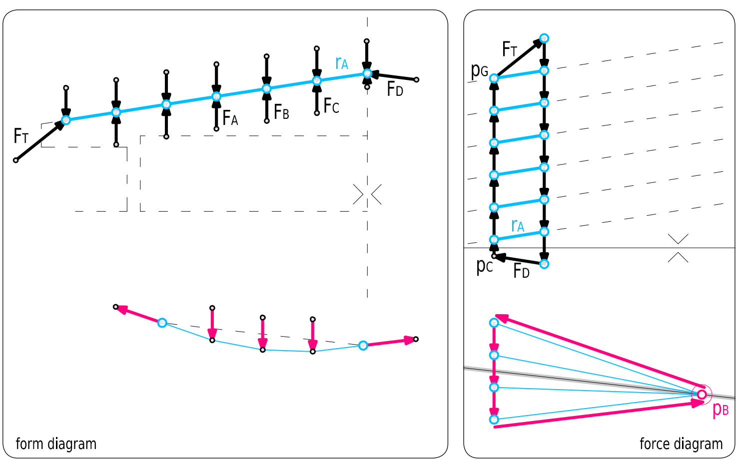


figure 319
reconstruction of Maillart's Chiasso sheds (snapshot 3); the linear domain of p_B is highlighted.

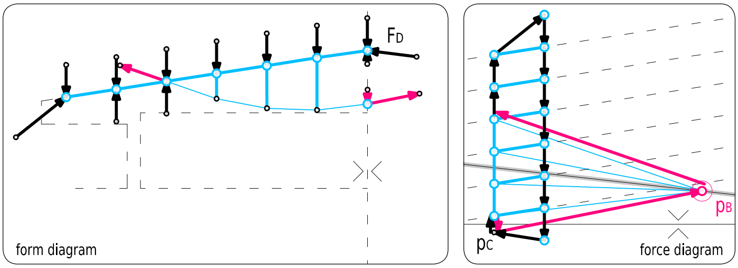


figure 320
reconstruction of Maillart's Chiasso sheds (snapshot 4); the linear domain of p_B is highlighted.

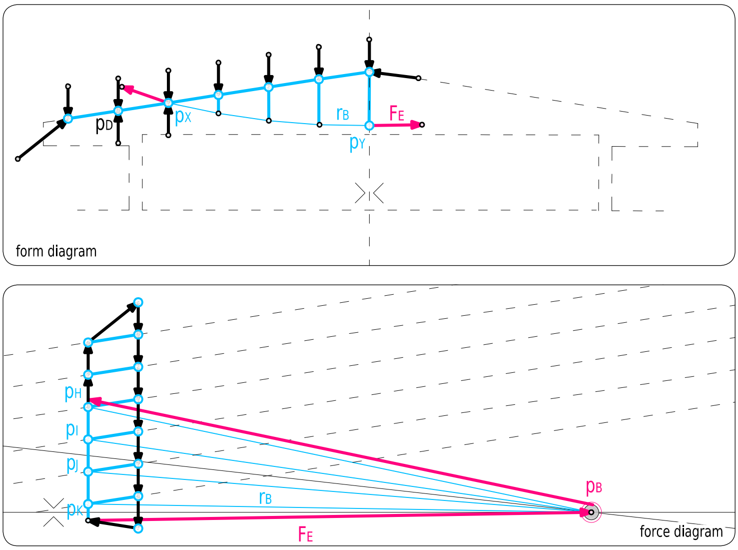


figure 321
reconstruction of Maillart's Chiasso sheds (snapshot 5); the linear domain of p_B is highlighted.

In figure 319, p_C has been constrained so that the force F_D becomes symmetrical to the rod r_A. On the opposite side, point $p_{G'}$ has been moved onto p_G in order to add the forces F_R and F_S and to work with F_T instead. Forces F_D and F_T have currently no real structural meaning so far. Their purpose is to provide "grips" on which other strut-and-tie sub-networks can be attached. For instance, F_D will be used to link the current strut-and-tie network with the symmetrical network.

For the same reason, the pole p_B is dragged along its domain in figure 321 until the force F_E and the rod r_B become perfectly symmetrical in both form and force diagrams. This position coincides with the axis of symmetry in the force diagram. Point p_B is consequently constrained at the intersection of this line of symmetry with the line of the poles of the funicular polyline. That ensures that the forces applied onto the position p_Y (in the form diagram) are symmetrical and in static equilibrium.

figure 322
reconstruction of Maillart's Chiasso sheds (snapshot 6); the linear domain of p_H, p_I, p_J and p_K are highlighted.

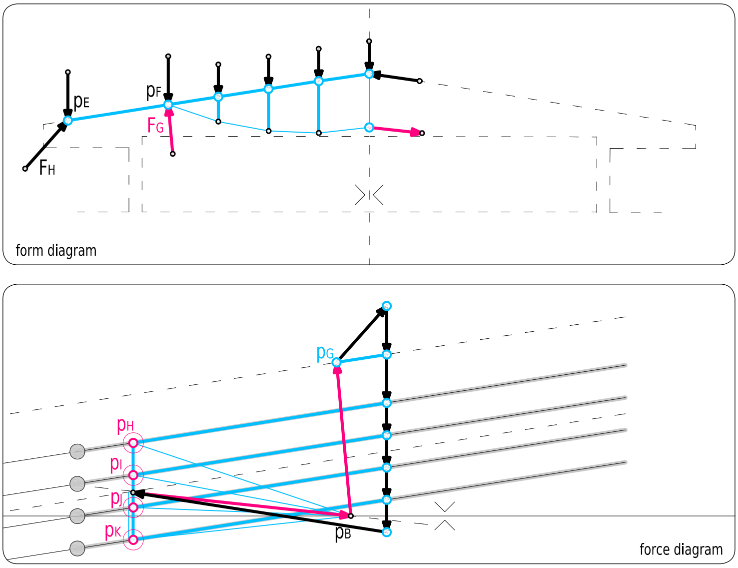
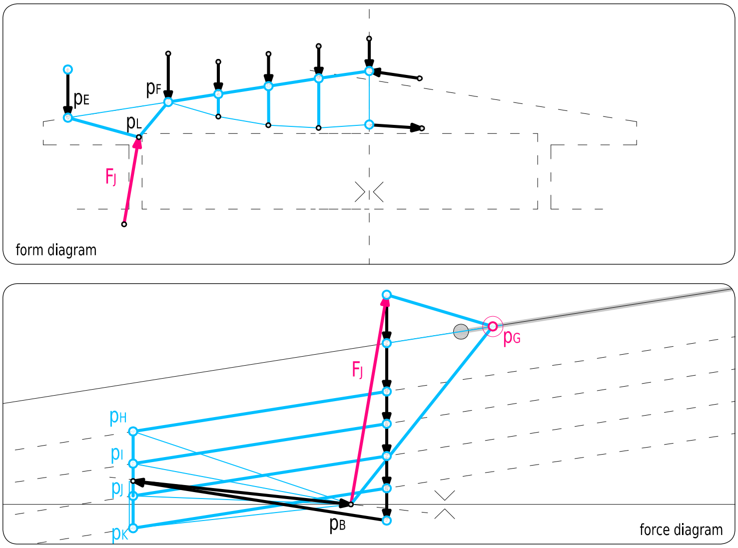


figure 323
reconstruction of Maillart's Chiasso sheds (snapshot 7); the linear domain of p_G is highlighted.



From figure 321 to figure 322, points p_H, p_I, p_J and p_K are moved along their domain in order to minimize the stresses in the funicular polyline: if they move, the geometry of the funicular line updates and the domain of p_B updates since the funicular polyline has to pass through points p_X and p_Y in the form diagram (figure 321).

A new point (p_L) is added in figure 323. This point is equilibrated with three forces that are constrained in such a way that the new force F_J is in equilibrium with the forces F_G and F_H. As soon as the new rods are formed, point p_G is automatically moved in order to stay inside its new propagation domain — this propagation domain ensures that p_L remains inside the buildable volume defined in figure 316.

The presence of this point p_L explains why both forces applied on p_D in figure 321 have been divided into two equals parts and moved on p_E and p_F in figure 322. Indeed, only three rods are connected by p_L thanks to this division and there is absolutely no node connecting more than three rods in the entire structure.

Finally, a new point p_M and two equilibrated forces are added in figure 324. In order to ensure that the new rod stays inside the available area for the column, the geometry of the funicular polyline has to be altered. The dependencies between p_B and the set $\{p_H, p_I, p_J, p_K\}$ are consequently switched to facilitate the control of the funicular polyline. As a result, p_B can be dragged again. Its domain, *i.e.* a tiny horizontal line segment in figure 324, ensures that every rod remains inside the buildable volume.

The shaping process can now be stopped since all the remaining forces are either applied loads or reaction forces. All the movable points — *e.g.* p_G, p_B and p_L — and their domains of solutions synthesize the remaining degrees of freedoms with which this structural shape can be deformed. For instance, an

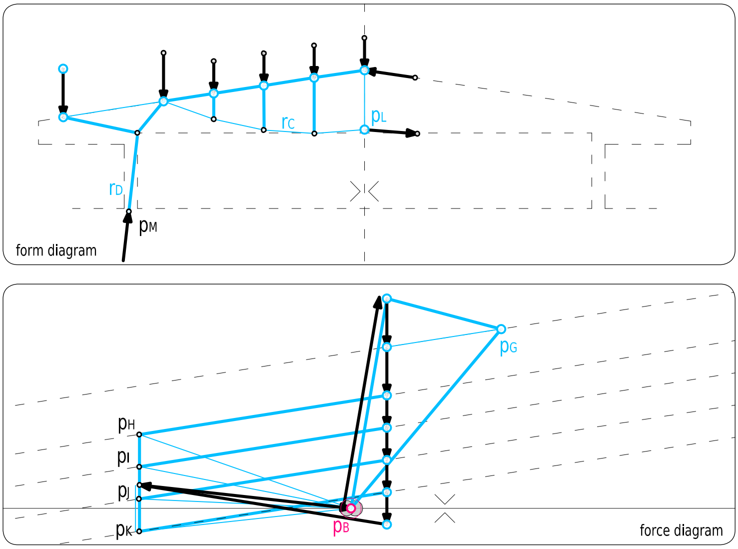


figure 324 reconstruction of Maillart's Chiasso sheds (snapshot 8); the linear domain of p_B is highlighted.

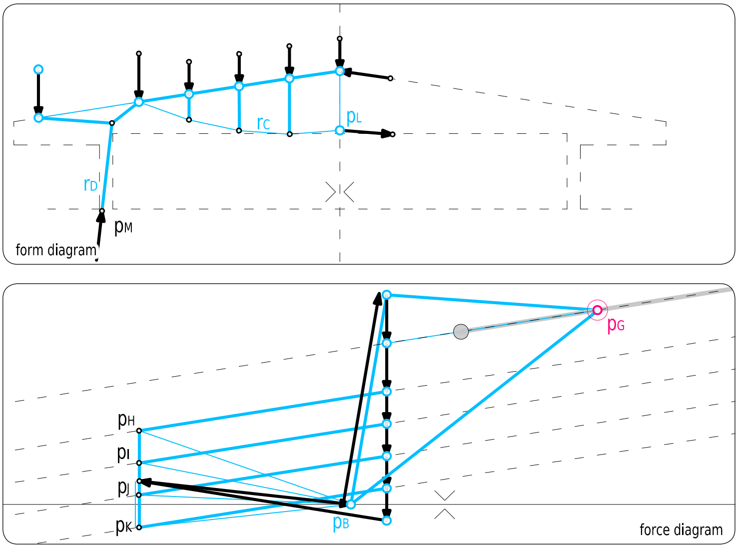


figure 325 reconstruction of Maillart's Chiasso sheds (snapshot 9); alternative shape obtained by moving p_G, its linear domain is highlighted.

alternative structural shape can be obtained by moving the point p_G further (figure 325). Thanks to the nature of the reciprocal diagrams, the impact of this alternative can be studied simultaneously in two ways: (1) according to spatial considerations — *e.g.* the increase of free space for storage in the form diagram — and (2) according to mechanical considerations — *e.g.* the material needed to resist the increasing magnitudes of forces described in the force diagram.

If desired, this strut-and-tie network can also be used as a basis to obtain a new strut-and-tie typology by dividing and recombine rods. The design approach that has been given here is essentially a bottom-up process: small parts are first built and the structure is obtained by assembling sub-parts together. The opposite, top-down, process could have been followed as well: a global rough network, *e.g.* a simply connected strut-and-tie network, would have been constructed initially and then modified by successive steps.

Sometimes rods, *e.g.* r_C, directly define the orientation and width of structural members and can be sized directly according to the magnitudes represented in the force diagram. Sometimes rods define the eccentricity of the axial and transverse forces acting on structural members. For instance, the eccentricity of r_D defines the bending moments occurring inside the column (figure 326). Also, the applied loads can be further discretized in order to de-

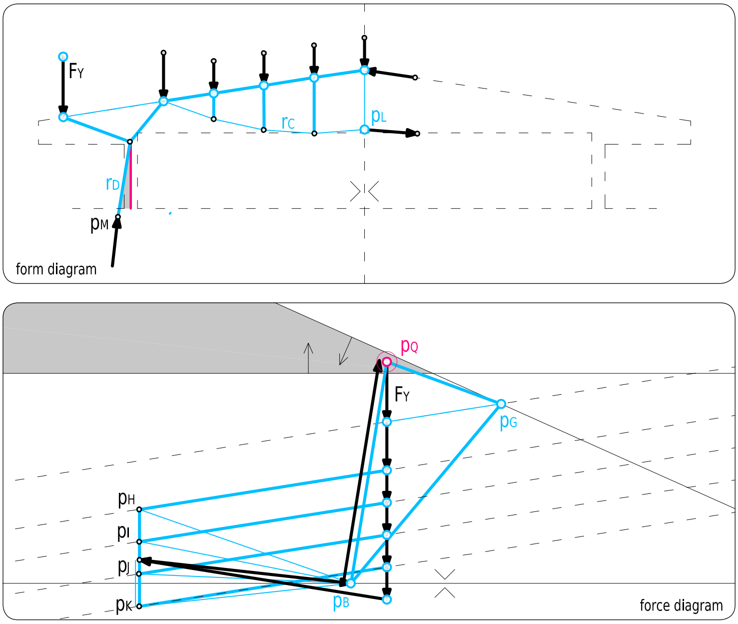


figure 326
reconstruction of Maillart's Chiasso sheds (snapshot 10); study of the impact of another load case by moving p_Q, its domain is highlighted.

scribe the bending moments in the roof. The visual and numerical description of the bending moments is given by other constraint-based routines that are at the disposal of the user.

The study of the structure under other load cases can also be performed on the same diagrams. The basic example of figure 326, shows the variation of the bending moments in the column when an additional load is hung on the roof edge. The magnitude of this additional load is actually defined by the position of p_Q in the force diagram — F_Y being the resultant of the initial load and the additional one. Moving p_Q modifies the load case and directly displays its impact on the stresses in the column. Moreover, the domain of p_Q already bounds the admissible additional load on the roof edge. If the structure has to sustain a bigger load on the roof edge, other positions of points in the form diagram or in the force diagram have to be changed by the user. More complex load cases can be obtained by moving multiple points at the same time.

(2) controlling bending moments with geometrical constraints · This paragraph highlights how bending moments can be controlled by purely geometric means with constraint-based graphic statics and what benefits it can bring. Given two loads, a clamped beam with uniform inertia throughout (figure 327) is to be sized using graphical methods — this example is from Muttoni/…·1997 page 7 point 1.2.2 and from Zastavni·2008a, appendix 4.

Thanks to the force diagram, the bending moments are given by a simply connected strut-and-tie network (*i.e.* a funicular polyline) whose moving pole is p* (figure 329). This pole is constrained in such a way that the funicular polyline passes through p_1 and a point p_2.

Using the lower-bound theorem of plastic theory (page 19), this beam can be designed for a desired magnitude of bending moments. Given the bending moment distribution of figure 329, it is understood that two plastic hinges must develop in order for a mechanism to exist and the beam to collapse (figure 328). The first plastic hinge is situated on p_0 and the other on p_5 or p_6 (figure 330).

figure 327
initial settings for the clamped beam.

figure 328
two possible collapse mechanisms.

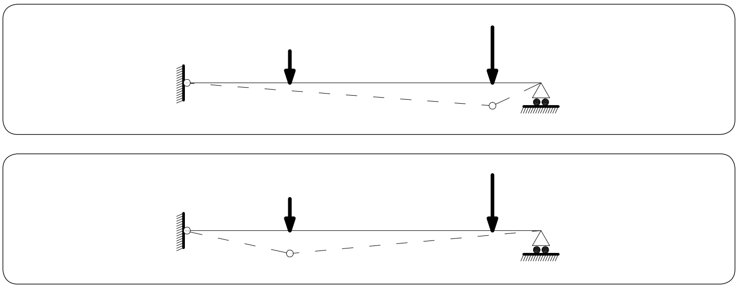
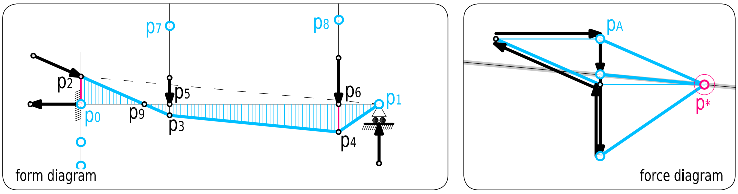


figure 329
bending moment distribution.



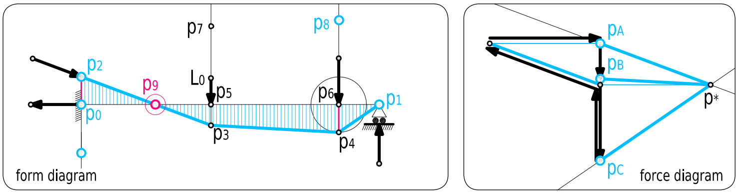


figure 330
bending moment distribution.

The smallest bending moments for which this beam has to be designed are consequently those occurring when the moment on p_0 is equivalent to the maximum between the moments on p_5 and p_6 — *i.e.* when the distance p_0p_2 is equal to the maximum between the distances p_3p_5 and p_4p_6. These values are here simply obtained by applying a MaxDistanceCompass[p_0 p_3 p_5 p_6 p_4] constraint on p_2 — see the paragraph entitled "the ProximityCondition constraint" (page 236). This compass is centred in p_0 and its radius is equal to the greatest distance between p_3p_5 and p_4p_6.

Because the pole p^* is dependent on p_2 and because p_3 and p_4 are dependent on the position of p^*, interdependency occurs between p^* and p_2. Concretely, when the position of p_2 is updated such that it belongs to the compass, the line of the poles on which p^* is constrained rotates. This updates the position of p^*, which in turn updates the altitude of p_3 and p_4, and subsequently the position of p_2 (due to the compass). These automatic updates eventually converge to a solution every time p^* is dragged.

As a result, a point of zero bending moment is identified on p_9 and the funicular polyline provides the smallest bending moment distribution for which this beam has to be designed. The required section of the beam can subsequently be obtained.

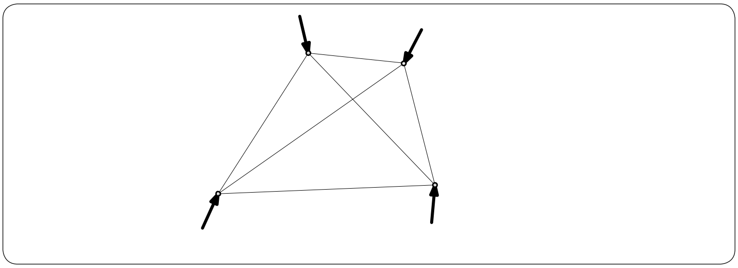
The above process is clearly faster and more interactive than any equivalent algebraic resolution.

For another example, figure 330 shows the same beam in which the point of application (p_7) of the load L_0 is not a given value but an unknown factor and in which the position of the zero bending moment p_9 is a free parameter. If point p_3 is constrained as follows, it defines the position of L_0 for which bending moments on p_0 and p_6 are equal:

$p_3 \in$ Straightedge[p_4 p^* p_B] ∩ Straightedge[p_2 p_2 p_9]
where: $p^* \in$ Straightedge[p_A p_2 p_9] ∩ Straightedge[p_C p_1 p_4]
$p_4 \in$ Straightedge[p_8 p_A p_B] ∩ Compass[p_6 p_0 p_2] ∩ HalfPlane[p_0 p_0 p_1]
$p_6 \in$ Straightedge[p_8 p_A p_B] ∩ Straightedge[p_0 p_0 p_1]
$p_7 \in$ Straightedge[p_3 p_B p_C]

(3) designing with indeterminacy · Structural indeterminacy can be defined as the ability of a strut-and-tie network to present different inner equilibria under constant external loads, meaning that multiple force diagrams can describe the same form diagram. This is encountered when at least one point can be moved in the force diagram without changing the orientation of any rod or force. The frame of figure 331 describes a simple structure of this kind.

figure 331
an indeterminate frame.



The final step of its construction is shown in figure 332. It presents the following geometric properties:

$p_4 \in$ Straightedge[p_0 p_0 p_2] ∩ Straightedge[p_1 p_1 p_3]
$p_8 \in$ Straightedge[p_{12} p_2 p_3]
$p_9 \in$ Straightedge[p_5 p_0 p_3]
$p_{10} \in$ Straightedge[p_9 p_0 p_4] ∩ Straightedge[p_6 p_0 p_1]
$p_{11} \in$ Straightedge[p_{10} p_1 p_4] ∩ Straightedge[p_7 p_1 p_2]
$p_{12} \in$ Straightedge[p_{11} p_2 p_4] ∩ Straightedge[p_9 p_3 p_4]

The point p_4 is added and constrained by the software itself in order to prevent the crossing of the rods — see sub-section 22 ("facilitating the crossing of rods", page 259). The resulting four rods act exactly as if they were two crossing rods. Indeed, the network is constrained such that both rods in a pair have always equal magnitude, equal orientation and are aligned in the form diagram.

Moving p_9 on its own domain — *i.e.* on a line — changes magnitudes without modifying any orientation (figure 332, figure 333, figure 334 and figure 335). It follows that the domain of p_9 is a comprehensive graphical representation of the indeterminacy of the frame. In most cases, the degree of indeterminacy of a structure is equal to the number of rectilinear domains inside which points can be moved without modifying any orientation of force or rod — a two-dimensional domain is said equivalent to two linear domains.

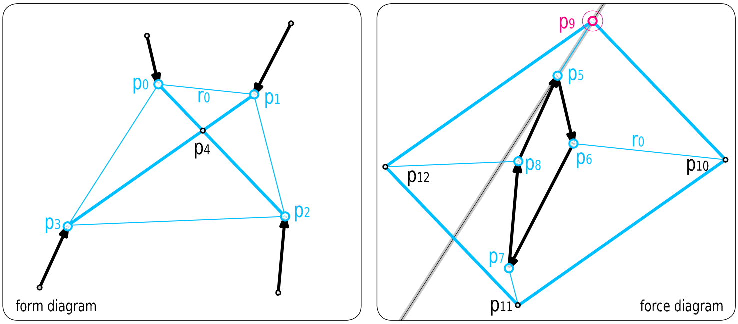


figure 332
the domain of p_9 is a representation of the indeterminacy of this frame.

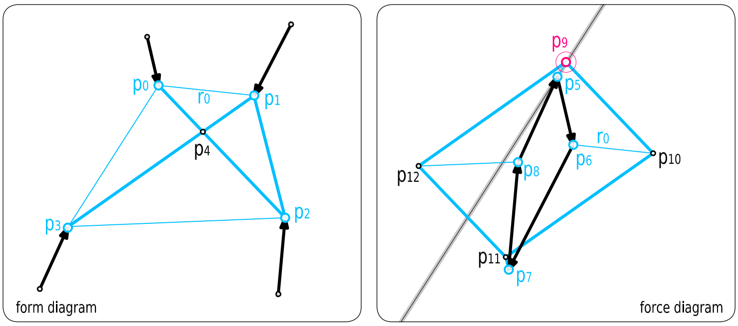


figure 333
p_9 moves and cause another force distribution.

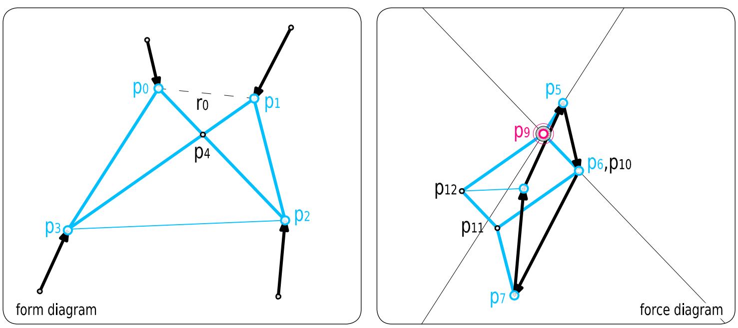


figure 334
another possible force distribution in which rod R_0 is superfluous.

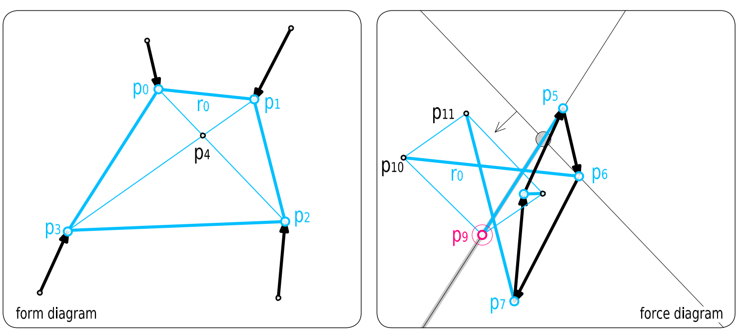


figure 335
another possible force distribution in which rod R_0 is in compression.

The compactness of the force diagram provides a direct insight about the efficiency of the chosen force distribution.

Moreover, the current position of p_9 consequently defines whether rods are in traction or in compression. For instance, the rod R_0 is in traction in figure 332 when p_9 belongs to HalfPlane[p_6 p_0 p_2], is unnecessary in figure 334 when p_9 belongs to Straightedge[p_6 p_0 p_2] and is in compression in figure 335 when p_9 belongs to HalfPlane[p_6 p_2 p_0]. Cross-sections can then be deduced from the force diagram so that the inertias impose the desired stress distribution.

The graphical handling of structural indeterminacy allows unprecedented control over the structure being shaped. For instance, it is asked which orientation of the load F_0 would ensure rod R_0 remains compressed without changing stresses inside the ties of figure 335. The first step to solving this issue is to fix the inner tension forces by switching the dependencies between p_6 and p_8 so that p_8 becomes completely free of constraint and p_6 becomes constrained on Straightedge[p_{10} p_0 p_1] (figure 336). Dependencies can then be further switched between p_6 and p_{13} so that p_{13} becomes free of constraint and point p_6 must then be further constrained on Straightedge[p_5 p_0 p_{13}] so that the orientation of F_0 is given by point p_{13} in the form diagram. The resulting propagation domain of p_{13} ultimately describes all the orientations that the load F_0 can have in order to ensure that the rod R_0 remains compressed.

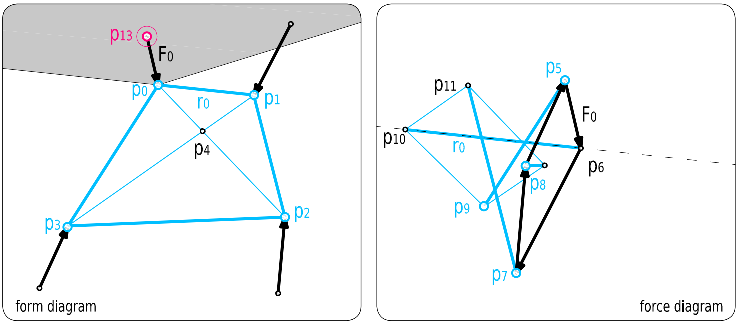


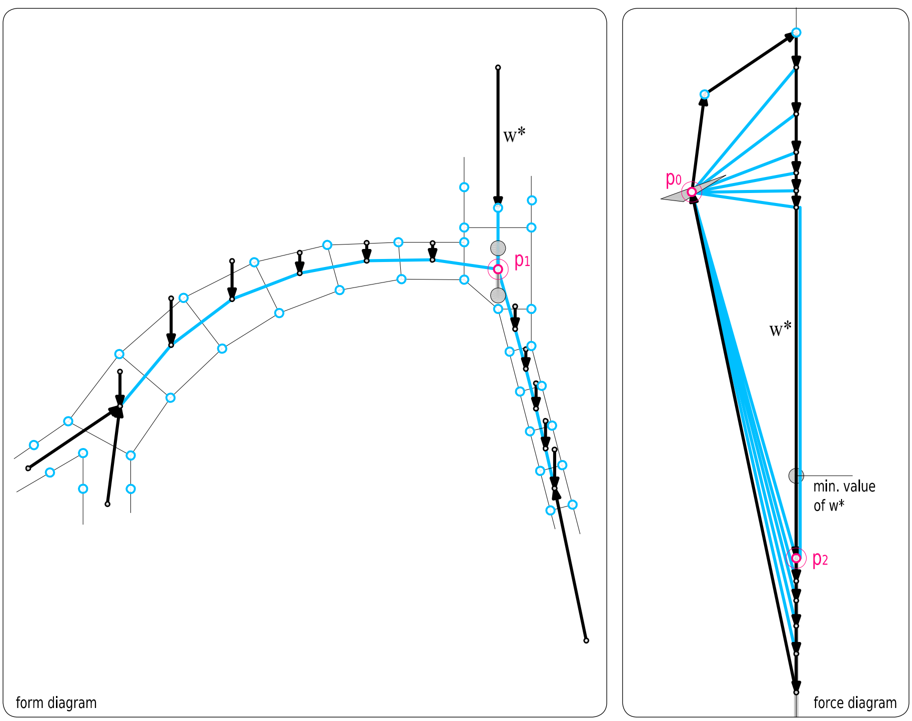
figure 336
the domain of p_{13} reflects all the orientations the load F_0 can have so the R_0 remains compressed.

(4) assessing the stability of a masonry complex · The final example is analytical rather than design-oriented: the stability of a given geometry of compressed voussoirs has to be checked. This means that a thrust line balanced with the dead loads of the voussoirs must exist inside the arch geometry.

The set of possible thrust lines inside the arch of figure 337 can be fully characterised with graphical domains of solutions. The first stage is to build a Boolean combination of HalfPlane constraints that corresponds to the geometry of the arch and to apply it to each node of the generated thrust line. The generality of this description method allows local specificities — *e.g.* irregularities detected in situ — to be taken into account without additional device.

The propagation of this set of constraints onto p_0 provides the near-triangular region in which p_0. This domain reflects the entire set of valid geometries that the thrust line can have. Moreover, the edges of the near-triangular region provide the extreme positions of p_0 for which rays are of minimum and maximum magnitudes.

figure 337
A masonry arch and its inner thrust line; highlighted regions represent the domain of stability of this arch.



The domain of p_1 — a linear segment —, also reflects the set of possible thrust lines, this time in the form diagram.

Since point p_2 is also a parameter of the thrust line, its domain is affected by the propagation of the constraints that define the geometry of the arch. As a result, its domain covers all the domain of stability of the arch too. Its domain therefore synthesises all the possible values of the weight w^*, and in particular its minimum value, for which the stability of the arch is guaranteed.

Furthermore if the forces corresponding to the actual dead load of the voussoirs are calculated with graphical constraints based on the geometry of these voussoirs and if the position of each load is constrained to be in line with the centre of gravity of the voussoir, the process then becomes design-oriented: moving a point that defines a voussoir would lead to the update of its dead load, to the update of the position of the pole p_0 and eventually to the update of the thrust line. That means that an interactive optimisation of the geometry of the voussoirs can be carried out in a fully graphical way.

27 future research

The framework, as presented here, still exposes significant limitations that prevent its full operational use. However, there is no reason to rule out the possibility of filling in and extending the existing framework in a manner that respects the objectives outlined in sub-section 03 ("answers: exemplary practices", page 17) and sub-section 04 ("proposal: a tool to accompany the construction of static equilibriums", page 29). Moreover, the full graphical approach taken in this research has already provided results that suggest fruitful new advances in the design and understanding of structural equilibrium. The paragraphs below discuss these perspectives of research.

defining additional algorithms for constraint propagation · It has been shown in sub-section 19 ("constraint propagations", page 201) that additional algorithms are still required to ensure the complete propagation of constraints, and hence the precise description of every domain of solutions. Although this current lack of completeness does not jeopardise the dynamic handling of strut-and-tie networks — *e.g.* empty domains can be avoided without that device — and the other techniques developed in this thesis, it forces the user to be careful in interpreting domains. Developments in this field should therefore be a priority.

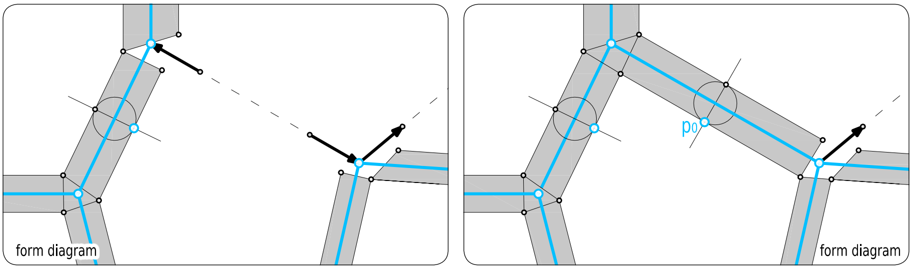
assessing the tool and enhancing its usability · As the environment is intended to be extremely intuitive and easy to use, in-situ practical assessments are mandatory in order to measure and improve the actual speed of software processing, the relevance and extent of the tool's capabilities and its ergonomics. These assessments should lead to a definition of an appropriate graphical user interface — GUI — a rewriting of the inner algorithms and the definition of new simplifying routines — *i.e.* new and more intuitive geometric constraints. Prior to the coding of these routines, a specific study of their geometric properties must be undertaken in order to define them robustly — *i.e.* to ensure that any change of position of a point will not jeopardise the sought behaviour of the constraint.

On the other hand, it would be beneficial to identify a number of best practices that may guide the designer towards a productive process. These are all the more welcome as the user is the sole master of each design choice and the only person responsible for the proper course of the graphical operations.

developing specific fields of applications · Another part of the research to be continued is the development of libraries of routines aimed at constructing specific abstraction models: thrust lines and stereotomy for masonry design and analysis (Heyman·1995); discontinuous stress fields in reinforced concrete (Kostic·2009, Muttoni/…·2011) or in reinforced wood (Trautz/Koj·2009a, Trautz/Koj·2009b), graphical methods for soil mechanics (Chalmers·1881, Terzaghi·1966) *etc.* Again, the definition of these routines must be preceded by a complete study of the geometric properties — concerning both the shape and the forces — on which they are to be built.

By way of illustration, the library dedicated to the design of discontinuous stress fields can be regarded as a plug-in that replaces the set of equilibrium operations by adding an additional layer of abstraction. For example, the operation transforming two well-suited forces into a new rod is superseded by an operation transforming two well-suited forces into a new rectilinear stress field (figure 338). The latter operation executes the former but also applies new geometric constraints that guarantee the specific properties of discontinuous stress fields — *e.g.* stress fields cannot be superimposed, their width (in the form diagram) can vary but must be greater than the one allowed by the strength of the material (regarding the actual resultant stress and the thickness of the beam) and nodal stress fields must present a geometry capable of uniform stress distribution.

figure 338 construction of a rectilinear stress field (right) from two opposite forces (left): dragging p_0 will change the width of this stress field and automatically update the geometry of the nodes.



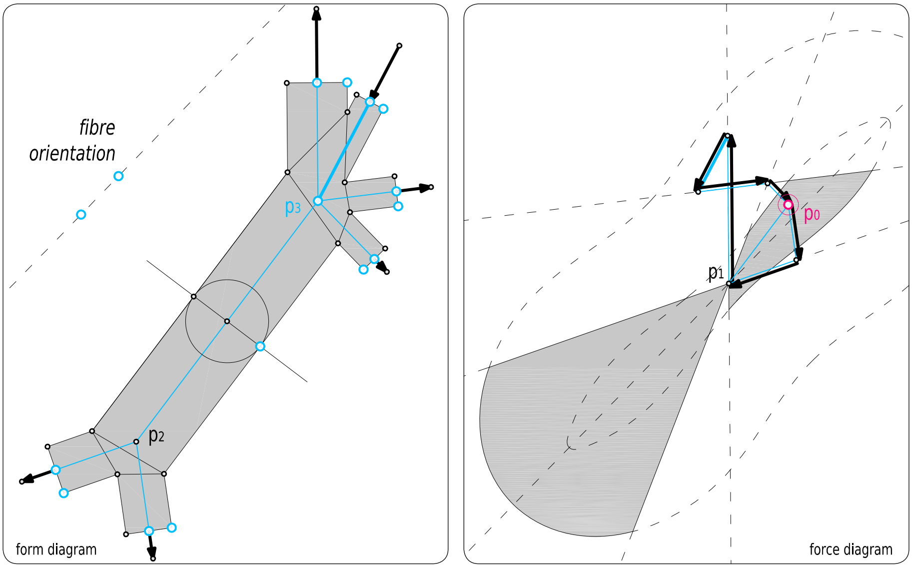


figure 339
a rectilinear stress field: dragging p_0 will cause the rectilinear stress field to be rotated around p_3 or to switch from traction to compression; as long as p_0 remains within its graphical domain of solution (the shaded area in the force diagram), the strength of this stress field is ensured to be below the plastic limit of the wood; the asymmetrical curvy shape of this domain of solution is due to the anisotropic nature of wood.

The details of this library have already been partially developed by the author (Fivet·2012). It has been reported that, assuming some particular conditions guaranteeing its plastic behaviour, discontinuous stress fields inside wood-joints can be modelled in full with geometric constraints related to the form diagram and the force diagram. One interesting feature of this development is the ability to represent graphically the range of force magnitudes and orientations that ensure a certain stress field remains below the plastic limit of this anisotropic material (figure 339). Another feature is the ability to control and handle the state of stress acting on a particular cut plane by way of a *Mohr*'s circle constructed in the force diagram (figure 340).

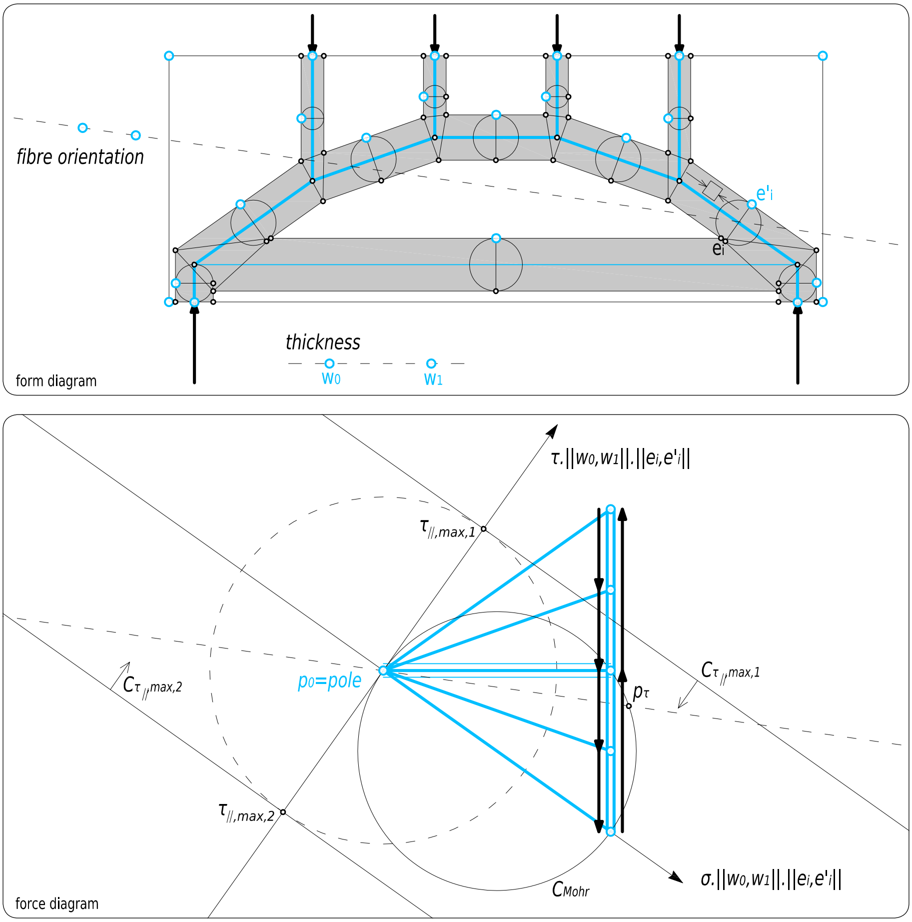


figure 340
study of the state of the stress acting on a particular position and cut plane: the *Mohr*'s circle is constructed in the force diagram in full; τ represents the tangential component of the constraint and σ the normal component; for example, point $p_τ$ reflects the state of the stress acting on the plane parallel to the fibre orientation; as a result, constraining $p_τ$ further will affect the domain of solutions of point e_i and hence, the width of the corresponding rectilinear stress field.

enhancing the geometric understanding of structures · This thesis has explored how (and the extent to which) any static equilibrium may be conditioned by purely geometric properties — *e.g.* geometric conditions for a rod to remain in traction or compression and the geometric description of variables of indeterminacy. Indeed, it seems that constraint-based graphic statics is rather conducive to an entirely geometry-based reconstruction of the classical theory of structures. Given the unique nature of such a reconstruction — *i.e.* at the crossroads of graph theory and (graphical) set theory — it can be hoped that research in this direction will result in the discovery of original theorems of mechanics of structures. Not only would such results benefit from the computational simplification offered by the force diagram and the graphical domains of freedoms, but also from the visual expressiveness they give the designer.

Two specific fields of study can already be tackled:

(1) the study of the close correlation that exists between the topology of the graphical domains of freedom and the rigidity of the structure. This study would lead to the definition of new geometrical criteria for structural robustness as well as to their dynamic control for design purposes.

Structural robustness is defined as "*the property of systems that enables them to survive unforeseen or unusual circumstances*" (Knoll/Vogel·2009) — *e.g.* excessive loads or the collapse of part of the structure. Robustness can be obtained from various strategies, including sizing for a strength above the minimum theoretically required and the design of multiple load paths. In this case, the former and the latter are dealt with by graphical domains of freedom and their behaviour after the elimination of rods.

(2) the study of the ability to execute algorithmic structural optimisations through the application of geometrical constraints. Structural optimisation is defined as "*the subject of making an assemblage of materials sustain loads in the best way*" (Christensen/...·2009) and commonly refers to specific computerised algorithms. It is achieved by automatically varying the topology of the structure, its geometries and the stiffness of its members automatically. The use of reciprocal diagrams for discrete topology optimisation has already unveiled very interesting computational benefits in Micheletti·2008, Block·2009, Beghini·2013 and Baker/...·2013. Constraint-based graphic statics offers a fairly original theoretical basis on which the power of optimisation methods

is yet to be discovered. This is mostly due to the geometric simplifications offered by the force diagram and the transcription of complex equational problems into symbolic geometric shapes.

extending the environment to the third dimension · Although great masters of the past have built spatial structures for millennia with the aid of working drawings which were only plane, the environment presented in this thesis would be of narrow interest if its extension to the third dimension was inconceivable or simply not sought. Indeed, the interactive construction of spatial static equilibrium would certainly promote the emergence of new structural typologies and the mastering of problems that, today, are difficult to solve.

This extension can be envisioned in three ways:

(1) the *projective* approach: the form diagram is defined in a three-dimensional space but the force diagram is not; the force diagram is dynamically adjusted based on the current axonometric projection of the form diagram; geometric constraints are planar and applied either on the planar force diagram or on particular axonometric projections of the form diagram

(2) the *full 3D* approach: the form diagram, the force diagram and geometric constraints are all defined and handled in a three-dimensional space

(3) the *composite* approach: the form diagram, the force diagram and geometric constraints are defined in a three-dimensional space, but represented and handled on axonometric projections.

The first approach is apparently the only one that has received extensive development in the literature so far — *e.g.* Daubresse·1904, Henneberg·1911, Mayor·1926, Foulon·1969 and, to a certain extent, Block/…·2007. It actually takes advantage of the following properties: *"If forces in space are in equilibrium, their projections on any plane are also in equilibrium.* […] *For a system of forces in space to be in equilibrium, it is both necessary and sufficient for the orthogonal projections on three rectangular planes to be in equilibrium."* (Daubresse·1904, page 43, free translation). Another beneficial consequence of orthogonal projections is the conservation of parallelism between reciprocal rods.

This approach is the fastest way to implement tri-dimensional constraint-based graphic statics since there are no conceptual differences between the planar constraint-based reciprocal diagrams and the parallel projections.

Preventing rods and forces from being intersected into the projected form diagram — see sub-section 21 ("constraints for a uniform reading cycle of forces", page 243) — is one of the few additional mechanisms that must be carried out. However, being unable to apply spatial geometric constraints seriously limits the benefits of this approach.

The second approach is probably the most natural, but requires new theoretical research regarding the properties of tridimensional graphic statics based on premises that are completely different from those already developed in the literature. One basis may be the paper of Rankine (Rankine·1864) in which magnitudes are represented by volumes of polyhedral frames (Akbarzadeh/...·2013). Another would represent magnitudes by rods in a three-dimensional diagram. Both diagrams will be correct according to new 3D rules — *e.g.* rules governing the reading cycle of forces in space — but the 2D representations of these diagrams may not be reciprocal in the classical point of view. The merits of this approach cannot be evaluated without an adapted user interface.

The third approach is halfway between the first and the second approach. It is distinguished from the first approach by allowing the application of spatial geometric constraints and distinguished from the second approach by securing the reciprocity — *i.e.* parallelism and reading cycles — of the displayed 2D diagrams. In other words, the third approach allows the control of pure 3D treatments through a reciprocal 2D display. These preliminary remarks motivate the development of this last approach.

The extension of the $\mathsf{Proximity}_{2D}$ and $\mathsf{Laterality}_{2D}$ relationships to the third dimension is direct:

- four points p_0, p_1, p_2 and p_3 satisfy the $\mathsf{Proximity}_{3D}[p_0\ p_1\ p_2\ p_3]$ relationship only if the distance from p_0 to p_1 is less than or equal to the distance from p_2 to p_3;
- five points p_0, p_1, p_2, p_3 and p_4 satisfy the $\mathsf{Laterality}_{3D}[p_0\ p_1\ p_2\ p_3\ p_4]$ relationship only if (a) p_2, p_3 and p_4 are collinear or (b) point p_0 is on the left of or in line with any observer positioned on p_1, standing up according to the direction from p_2 to p_3 — *i.e.* feet on p_2 and head on p_3 — and looking towards the direction from p_2 to p_4. The $\mathsf{Laterality}_{3D}$ relationship can also be defined using the right-hand rule shown in figure 341.

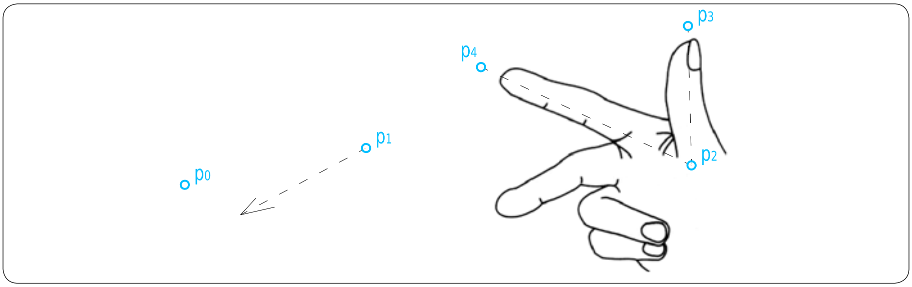


figure 341
mnemonic description of the Laterality$_{3D}$ relationship.

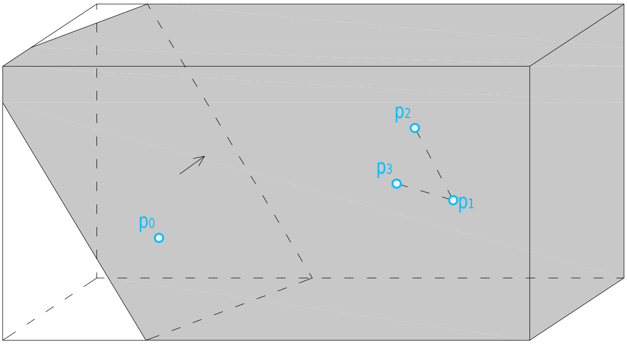


figure 342
representation of a HalfSpace [$p_0p_1p_2p_3$] fundamental constraint.

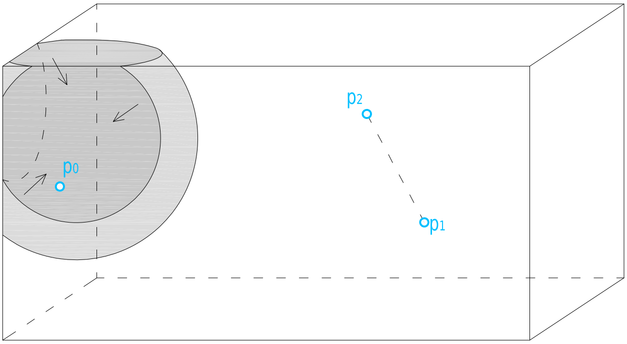


figure 343
representation of a SphereInside [$p_0p_1p_2$] fundamental constraint.

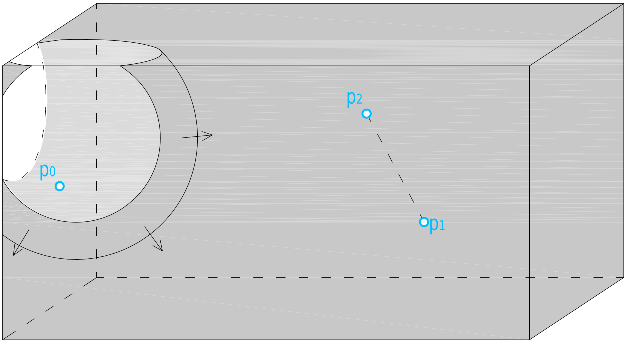


figure 344
representation of a SphereOutside [$p_0p_1p_2$] fundamental constraint

The three corresponding fundamental constraints HalfSpace[p_0 p_1 p_2 p_3], SphereInside[p_0 p_1 p_2] and SphereOutside[p_0 p_1 p_2] are illustrated in figure 342, figure 343 and figure 344 respectively.

Planar fundamental constraints can easily be deduced from these three tri-dimensional constraints by intersecting the latter with a plane defined as the intersection HalfSpace[p_0 p_1 p_2 p_3] ∩ HalfSpace[p_0 p_1 p_3 p_2].

As a beneficial consequence of the affine/metric distinction, these three constraints are sufficient for describing any geometric problem in space — in other words, the extension to the third dimension does not require the creation of new fundamental constraints. However, two new non-fundamental constraints mixing affine and metric considerations can be added to this set for practical reasons: they are expected to be used abundantly but they cannot be defined without interdependency. These two constraints — ConeInside and ConeOutside — are defined as follows:

- the ConeInside[p_0 p_1 p_2 p_3 p_4 p_5] constraint corresponds to the closed solid of revolution obtained by rotating a line passing through p_3 and parallel to the orientation p_4p_5 around the straight line passing through p_0 and parallel to the orientation p_1p_2. This solid is of a different geometric nature depending on whether these two lines are skew (figure 345), secant (figure 346) or parallel (figure 347).
- the ConeOutside[p_0 p_1 p_2 p_3 p_4 p_5] constraint corresponds to inversion of the open solid of revolution obtained in an analogous manner (figure 348).

These basic considerations highlight the direct analogy that exists between the planar constraint-based geometric environment described in this thesis and its tri-dimensional equivalent. Most mechanisms of planar constraint-based graphic statics — including the resolution of graphical inequalities through symbolic propagation of constraints, the management of interdependent constraints, the switching of constraint dependencies and the creation of dynamic conditional statements — have the same computational complexity in both 2D and 3D environments. The main subjects that still require special attention are (1) the adaptation of the propagation methods, (2) the identification of the 3D geometric constraints capable of ensuring the reciprocity of spatial diagrams of graphic statics — if it does exist — and (3) the dynamic graphical user interface allowing the intuitive control of these diagrams.

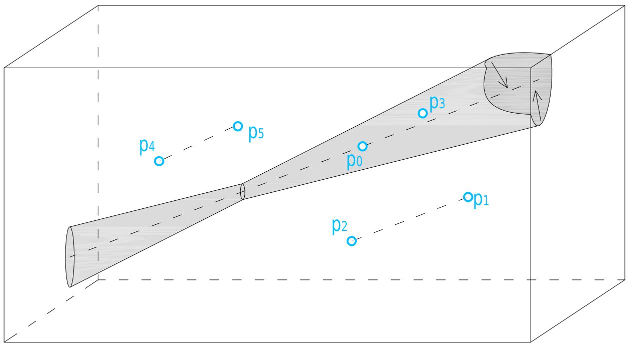


figure 345
representation of a ConeInside[p_0 p_1 p_2 p_3 p_4 p_5] constraint generated by two skew lines;

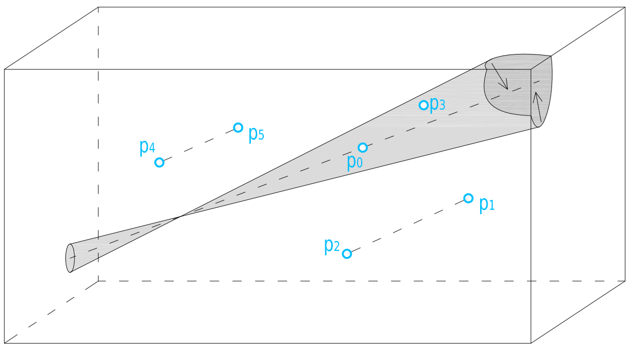


figure 346
representation of a ConeInside[p_0 p_1 p_2 p_3 p_4 p_5] constraint generated by two secant lines;

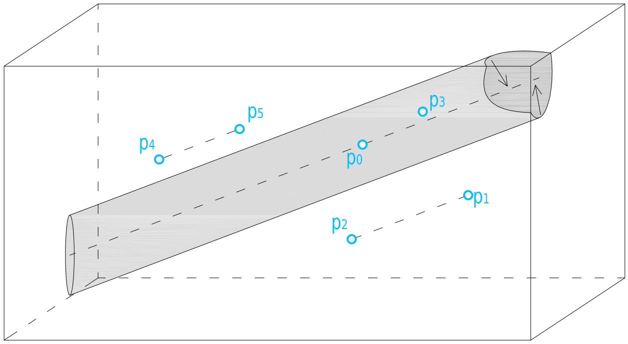


figure 347
representation of a ConeInside[p_0 p_1 p_2 p_3 p_4 p_5] constraint generated by two parallel lines;

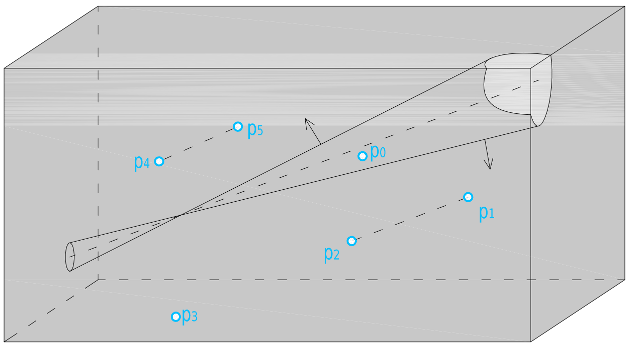


figure 348
representation of a ConeOutside[p_0 p_1 p_2 p_3 p_4 p_5] constraint generated by two secant lines;

28 conclusions

recapitulation · This thesis has defined rules and techniques for a computer-aided tool aimed at interactively assisting the definition of structural equilibriums.

The first section (“introduction”, page 1) contrasted the lack of appropriate tools for the initial shaping of structures with their significance for meeting consistency and efficiency requirements. It then briefly identified the expected capabilities of a tool that would be a useful complement to the existing set of structural design tools. In response to that, the main features of the proposal were finally presented and contextualized.

The second section (“geometric axiomatisation of graphic statics”, page 55) built an axiomatic set of geometric rules to define strut-and-tie networks in static equilibrium while the third section (“dynamic handling of geometric constraints”, page 135) presented original techniques to fulfil these rules when parameter positions vary dynamically. The fourth section (“production rules for computer-aided graphic statics”, page 265) subsequently set out the means by which the designer can produce plane constraint-based graphic statics using this environment.

Finally, the last section (“discussion”, page 285) argued the effectiveness of the proposed environment through various design applications and opened up new perspectives of research.

original contributions · The contributions made by this thesis are mainly theoretical. Two concepts have been outlined initially. They can both be regarded as an extension of classical graphic statics: the first links a graphical region with the admissible positions of each point that controls the geometry of graphic statics diagrams; the second prevents each temporary constructed diagram from being incomplete, *i.e.* not in static equilibrium. While the former lay behind the search for automated techniques for constructing graphical regions of solutions, the latter did this for equilibrium operations on diagrams. They consequently led to seek out a constraint-based graphic statics framework through various original statements, including:

- a renewed definition of the force diagram and its reciprocal rules with the form diagram
- a fully geometric axiomatisation of graphic statics and a new geometric grammar to describe it
- computer-aided rules capable of interactively assisting the construction and modification of reciprocal diagrams
- a dynamic geometry environment able to handle multiple solutions to a problem at the same time, switch the parametric hierarchy on demand, compute complete interdependency — *e.g.* allowing the geometric construction of algebraically curved constraints — and execute dynamic conditional statements geometrically
- some symbolic techniques of constraint propagation to ensure consistency between certain graphical domains of solutions.

As a result and from a broader standpoint, this thesis highlights a little more the significance of geometric reasoning in structural design: whether a structure is in static equilibrium is only a matter of geometry; whether a rod is in tension or compression is only a matter of geometry; whether a thrust line in equilibrium remains inside a given shape is only a matter of geometry — constraint-based graphic statics is an appropriate instrument for finding, expressing and handling all these geometric rules synthetically.

"The designs of an Engineer are geometric conceptions, his structures are geometric forms, within which forces statically combined act along geometric lines, so that it is natural that he strive to follow a train of geometric thought." (Chalmers·1881, preface page viii)

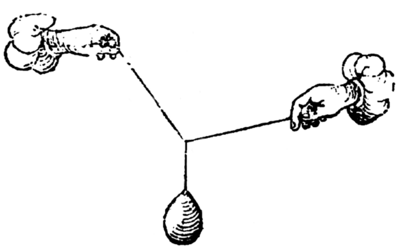


figure 349
three forces in equilibrium (Stevin/...·1634, page 505);

reservations · This praise of geometry also highlights the fact that the tool only deals with static equilibriums and with what can somehow be expressed through geometry. As such, the tool has specific possibilities and should be used alongside the many tools available. It is designed to support structural designers before and after analysis tools. Sometimes, in the best cases, it can make analysis and optimisation tools superfluous but it is not supposed to replace them.

However, numerous purposes can be achieved with this tool, using only geometry and static equilibrium. Sub-section 26 ("applications", page 287) has established a preliminary list and has detailed some examples. Most of them may seem unexpected and currently require more research. For instance, Mohr's circles can be used with constraint-based graphic statics in order to study the state of stresses of a discontinuous stress field inside anisotropic material. Also, permutation of hierarchy of constraint-based graphic statics can be used to control beam deflections directly.

Applications of graphic statics are many and form a treasure which cannot be perceived by a beginner. The daily use of graphic statics is rare nowadays. Computerized interfaces of graphic statics have no future if their advantages are not advertised to current and future practitioners.

This is all the more the case as graphic statics compel designers to be properly responsible for the outcome of the process. The same is true for constraint-based graphic statics. The proposed tool cannot be viewed as a software that provides an high-performance output on the basis of a series of inputs chosen by the user. The proposed tool is rather a drafting table equipped with various implements. The user has to know how to use these implements in order to succeed, even if they are very intuitive.

expected outcomes · Although further developments and field assessments are still needed, substantial benefits can reasonably be anticipated. What constraint-based graphic statics would add most is better control of the structural typology being shaped.

This control would allow the designer to give more coherence to the structural shape in light of conditions that may and may not be objectivised and which might arise only during the course of the design process, and/or that are just a product of the designer's unpredictable sensitivity and creativity — as long as those conditions are somehow connected to the geometrical shaping of the structure, directly or indirectly. For example, these conditions might concern its integration with the spatial context, its architectural quality, its robustness, its durability, the economy of material, the process of building, its aesthetics, its functional uses etc. They are intended to improve the architectural, economic and ecological qualities of the structure and its surroundings.

This is a new kind of control. Indeed, constraint-based graphic statics encourages the emergence of new design approaches that are highly interactive, pre-cognitive and chronology-free:

- *highly interactive* because (1) equilibrium and force magnitudes are visually expressed through the force diagram, (2) forces and structural geometries are bound together in a dynamic and homogeneous environment and (3) routines and parameterised dragging give the user ongoing ease for modifying and constraining the equilibrium being shaped
- *pre-cognitive* because (1) the user continuously knows that the structure is in equilibrium before it is even checked and (2) every graphical region of solutions marks out the range of design possibilities before they are even explored
- *chronology-free* because (1) equilibrium operations are bijective and hence allow the user to alternate between bottom-up approaches — *e.g.* the assembly of pre-existing smaller structural parts — and top-down approaches — *e.g.* the refinement of existing structural parts — at any time, (2) switching constraint dependencies frees the user from the traditional parametric hierarchy, (3) any stored set of operations applied to construct a custom static equilibrium can instantaneously be rerun with new initial conditions and (4) the design of the rod geometries, their inner stresses, their sections and inertias, their static indeterminacy, their boundary conditions and the material strength can be performed independently and simultaneously.

In other words, constraint-based graphic statics offers more control over the structural shape and its inner stresses along with more freedom of composition. This has the potential of supporting new, more efficient and more appropriate structural design methods as well as encouraging the user to achieve new, more efficient and more appropriate structural typologies.

REFERENCES

◈ engineering practices

○ geometry
◎ mechanical geometry
◍ computational geometry

■ graphic statics
◪ other tools, methods and theories

◬ computer programming

A

Bruno Abdank-Abakanowicz
Les intégraphes, la courbe intégrale et ses applications, étude sur un nouveau système d'intégrateurs mécaniques. ◎ Abakanowicz·1886
Gauthier-Villars · Paris · 1886

Harold Abelson / Gerald Jay Sussman
Structure and Interpretation of Computer Programs. △ Abelson/Sussman·1996
second edition · MIT Press · Cambridge, Massachusetts · 1996

Bill Addis
The Art of the Strutural Engineer. ◈ Addis·1994
Artemis · London · isbn 1874056412 · first published in 1994

Bill Addis
Creativity and Innovation The structural engineer's contribution to design. ◈ Addis·2001
Architectural Press · Oxford · isbn 0750642106 · 2001

Bill Addis
Building: 3000 Years of Design Engineering and Construction. ◈ Addis·2007
Phaidon · London NewYork · isbn 978-0714841465 · first published in 2007 · reprinted in 2008

Bill Addis
The Relationship between Geometry and Statics in Structural Design. ◪ Addis·2010
Geometry and Proportion in Structural Design, essays in Ricardo Aroca's Honour · edited by Pepa Cassinello, Santiago Huerta, José Miguel de Prada Poole & Ricardo Sánchez Lampreave · Universidad Politecnica de Madrid · Escuela Tecnica Superior de Arquitectura · Madrid · 2010

Masoud Akbarzadeh / Tom Van Mele / Philippe Block
Equilibrium of Spatial Structures using 3-d Reciprocal Diagrams. ■ Akbarzadeh/...·2013
Proceedings of IASS Symposium · Beyond the limits of man · Wroclaw University of Technology · Poland · 2013

Edward Allen / Waclaw Zalewski
Form and Forces: Designing Efficient, Expressive Structures. ■ Allen·2009
Wiley & Sons · isbn 978-0470174654 · 2009

Stanford Anderson
Eladio Dieste: Innovation in Structural Art. ◈ Anderson·2004
Princeton Architectural Press · 2004

Archimedes of Syracuse / François Peyrard
Œuvres d'Archimède, traduites littéralement avec un commentaire. ○ Archimedes/...·1808
second edition · Fr. Buisson Libraire-Editeur · Paris · 1808.

I.I. Artobolevskii
Mechanisms for the generation of plane curves. ◎ Artobolevski·1964
translated by R.D. Wills · translation edited by W. Johnson · Pergamon Press · 1964

Autodesk
Robot Structural Analysis. ◪ Autodesk/...·2008
software · www.autodesk.com/robot · available from 2008

B

William F. Baker / Lauren L. Beghini / Arkadiusz Mazurek / Juan Carrion / Alessandro Beghini
Baker/...·2013 ◪ *Maxwell's reciprocal diagrams and discrete Michell frames.*
Struct Multidisc Optim · Springer · Published online : 16 March 2013

Cecil Balmond
Balmond·2007 ◈ *Informal.*
Prestel · 2007

John H. Barr
Barr·1899 ◎ *Kinematics of Machinery,*
A brief treatise on constrained motions of machine elements.
first Edition · John Wiley and Son · New York · 1899

Roman Barták
Barták·1999 ◍ *Constraint Programming: In Pursuit of the Holy Grail.*
Proceedings of the Week of Doctoral Students (WDS99) · part IV · MatFyzPress · Prague · pages 555-564 · June, 1999

Maria G. Bartolini Bussi / Annalisa Martinez / Marcello Pergola / Carla Zanoli / Marco Turrini / Daniela Nasi / Andrea Ricchetti / Stefano Malagoli / Maurizio Vacca / Paola Dinelli / Stefano Greco
Bartolini/...·1999 ◎ *Museo Universitario di Storia Naturale e della Strumentazione Scientifica.*
http://museo.unimo.it/theatrum/inizio.htm · Università degli studi di Modena e Reggio Emilia · visited on July 12, 2012 · 1999

Maria G. Bartolini Bussi / Michela Maschietto
Bartolini/...·2006 ◎ *Macchine matematiche: dalla storia alla scuola.*
Convergenze · a cura di F. Arzarello, L.Giacardi, B. Lazzari · Springer · isbn: 978-8847004023 · 2006

Christoph Baumberger
Baumberger·2012 ◈ *Structural Concepts and Spatial Design:*
On the Relationship Between Architect and Engineer.
in Cooperation: the Engineer and the Architect · page 57 · edited by Aita Flury · Birkhäuser · Basel · 2012

J. Bauschinger
Bauschinger·1871 ■ *Elemente der Graphischen Statik.*
R. Oldenbourg · München · 1871

Lauren Lynne Beghini
Beghini·2013 ◪ *Structural optimization using Maxwell's reciprocal diagrams and graphic statics.*
in Phd Thesis 'Building Science through topology optimization' · directed by Prof. Glaucio H. Paulino · University of Illinois at Urbana-Champaign · 2013

Corinne Bélier / Barry Bergdoll / Marc Le Coeur
Bélier/...·2012 ◈ *Labrouste (1801-1875), architecte*
La structure mise en lumière.
Éditions Nicolas Chaudun · Cité de l'architecture & du patrimoine · isbn 978-2350391380 · 2012

Eduardo Benvenuto
Benvenuto·1985 ◪ *The Parallelogram of Forces.*
Meccanica · issue 20 · pages 99-109 · 1985

Christian Bessiere
Bessiere·2006 ◍ *Constraint Propagation.*
Technical Report LIRMM 06020 · University of Montpellier · 2006

BERNHARD BETTIG / JAMI SHAH
Solution Selectors:
A User-Oriented Answer to the Multiple Solution Problem in Constraint Solving. ◍ Bettig/Shah·2003
Journal of Mechanical Design · volume 125 · pages 443-451 · September, 2003

BERNHARD BETTIG / CHIRSTOPH M. HOFFMANN
Geometric Constraint Solving in CAD. ◍ Bettig/...·2011
JCISE 021001 · 9 pages · March 9, 2011

MAX BILL
Robert Maillart, Ponts et Constructions. ◈ Bill·1955
Girsberger · 1955

GEORGE DAVID BIRKHOFF
A Set of Postulates for Plane Geometry, Based on Scale and Protractor. ○ Birkhoff·1932
The Annals of Mathematics · second series · volume 33 · number 2 · pages 329-345 · April, 1932

GEORGE DAVID BIRKHOFF / RALPH BEATLEY
Basic Geometry. ○ Birkhoff/...·1959
third edition · Scott, Foresman & Company · USA · 1959

PHILIPPE BLOCK / JOHN OCHSENDORF
Thrust Network Analysis: A New Methodology for Three-Dimensional Equilibrium. ◪ Block/...·2007
Journal of the International Association for shell and spatial structures · issue 48 · pages 167-173 · 2007

PHILIPPE BLOCK
Thrust Network Analysis, Exploring Three-dimensional Equilibrium. ◪ Block·2009
PhD Thesis directed by Prof. John Ochsendorf · Massachusetts Institute of Technology · Cambridge · 2009

HARRY BLUM
A Transformation for extracting new descriptors of shape. ◍ Blum·1967
Models for the Perception of Speech and Visual Form · edited by W. Wathen-Dunn · pages 362–380 · M.I.T. Press · Cambridge · Massachussetts · USA · 1967

J.A. BONDY / U.S.R. MURTY
Graph Theory. ◪ Bondy/Murty·2008
Graduate Texts in Mathematics · edited by S. Axler & K.A. Ribet
Springer · isbn 978-1-84628-969-9 · 2008

WILLIAM BOUMA / IOANNIS FUDOS / CHRISTOPH HOFFMANN / JIAZHEN CAI / ROBERT PAIGE
A Geometric Constraint Solver. ◍ Bouma/Fudos/...·1995
in "Computer-aided Design" · volume 27 · number 6 · pages 487-501 · 1995

ROBERT HENRY BOW
Economics of construction in relation to framed structures. ■ Bow·1873
E. & F. N. Spon · London NewYork Edinburgh · 1873

ROBERT HENRY BOW
A treatise on bracing
with its application to bridges and other structures of wood or iron. ■ Bow·1874
D. Van Nostrand · New-York · 1874

H. BROCARD
Note sur un compas trisecteur proposé par M Laisant. ◎ Brocard·1875
Bulletin de la Société Mathématique de France · volume 3 · pages 47-48 · 1875

C

William D. Jr. Callister
Callister·2007 ◪ *Materials Science and Engineering – An Introduction.*
Seventh Edition · John Wiley & Sons · isbn 978-0471736967 · 2007

François Cardarelli
Cardarelli·2008 ◪ *Materials Handbook - A Concise Desktop Reference.*
Springer · isbn 978-1846286681 · 2008

James B. Chalmers
Chalmers·1881 ■ *Graphical Determination of Forces in Engineering Structures.*
Macmillan & Co · London · 1881

T.M. Charlton
Charlton·2002 ◈ *A History of the Theory of Structures in the Nineteenth Century.*
Cambridge University Press · isbn 978-0521524827 · 2002

Peter W. Christensen / Anders Klarbring
Christensen/...·2009 ◪ *An Introduction to Structural Optimization.*
Solids Mechanics and its Applications · volume 153 · edited by G.M.L. Gladwell · Springer · 2009

Andy Clark / David J. Chalmers
Clark/Chalmers·1998 ◪ *The Extended Mind.*
Analysis · volume 58 · pages 10-23 · 1998

George R. Collins
Collins·1963 ◈ *Antonio Gaudi: Structure and Form.*
Perspecta · volume 8 · MIT Press · pages 63-90 · 1963

Jürg Conzett / Judit Solt
Conzett/Solt·2008 ◈ *Soigner le savoir-faire.*
Tracés · volume 134 · 2008

Jürg Conzett / Gianfranco Bronzini / Patrick Gartmann / Michel Carlana / Andrea Iorio / Francesco Dal Co
Conzett/...·2011 ◈ *Jürg Conzett, Gianfranco Bronzini, Patrick Gartmann : forms of structures.*
Milano · Electa · 2011

Barthélémy-Édouard Cousinery
Cousinery·1839 ■ *Le calcul par le trait, ses éléments et ses applications, à la mesure des lignes, des surfaces et des cubes, à l'interpolation graphique et à la détermination, sur l'épure, de l'épaisseur des murs de soutènement et des murs de culée des voûtes.*
Carilian-Goeury & Vr Dalmont · Paris · 1839

Henry Crapo
Crapo·1979 ◪ *Structural Rigidity.*
Structural Topology · issue 19 · page 26 · 1979

Luigi Cremona
Cremona·1868 ■ *Corso di statica grafica.*
Regio Istituto Tecnico Superiore · Milano · 1868-1869

Luigi Cremona
Cremona·1872 ■ *Le Figure Reciproche Nella Statica Grafica.*
edited by R.A.d. Lincei · Opera Matematiche di Luigi Cremona · Hoepli · Milano · pages 336-364 · 1872

Luigi Cremona
Les figures réciproques en statique graphique. ■ Cremona·1885
Gauthier-Villars · Paris · 1885

Carl Culmann
Die graphische Statik. ■ Culmann·1866
Zürich · Meyer und Zeller · 1866

Carl Culmann
Traité de Statique Graphique. ■ Culmann·1880
Dunod · Paris · 1880

Carl Culmann / Karl Wilhelm Ritter
Anwendungen der Graphischen Statik. ■ Culmann/Ritter·1888
4 tomes · Meyer & Zeller · Zürich · 1888-1900

D

Paul Daubresse
Notes du cours de Statique Graphique professé aux écoles spéciales de l'Université catholique de Louvain. ■ Daubresse·1904
H. Ghysebrechts · Louvain · 1904-1905

Carl Dea
JavaFX 2.0: Introduction by Example. △ Dea·2011
APress · isbn 978-1430242574 · 2011

Joshua D. Deaton / Ramana V. Grandhi
A survey of structural and multidisciplinary continuum topology optimization: post 2000. ◪ Deaton/...·2013
Struct Multidisc Optim · published online: 04 july 2013

Mark de Berg / Otfried Cheong / Marc van Kreveld / Mark Overmars
Computational Geometry - Algorithms and Applications. ◍ deBerg/...·2008
third edition · Springer-Verlag · Berlin Heidelberg · isbn: 978-3540779735 · 2008

Eladio Dieste / Antonio Jiménez Torrecillas / Magdalena Torres Hidalgo / Nicolás Ramírez Moreno
Eladio Dieste, 1943-1996. ◈ Dieste/...·2001
Sevilla · Junta de Andalucía · Consejería de Obras Públicas y Transportes · 2001

Maurice Dohmen
A survey of constraint satisfaction techniques for geometric modeling. ◍ Dohmen·1995
Computers & Graphics · volume 19 · issue 6 · pages 831-845 · 1995

Philip Drew
Frei Otto - Form and Structure. ◈ Drew·1976
Crosby Lockwood Staples · London · isbn 0258970537 · 1976

A. Jay Du Bois
The elements of graphical statics and their application to framed structures. ■ DuBois·1875
John Wiley & son · New-York · 1875

Pierre Duhem
Les origines de la statique. ◪ Duhem·1905
Hermann · Paris · 1905-1906

E

Samuel Earnshaw
Earnshaw·1858 ◪ *A treatise on Statics*
containing the theory of equilibrium of forces and numerous examples.
fourth edition · London · Deighton, Bell and Co. · 1858

Henry T. Eddy
Eddy·1878 ■ *On the Two General Reciprocal Methods in Graphical Statics.*
American Journal of Mathematics, pure and applied · issue 1 · pages 322-335 & plates IV-V · 1878

Eric T. Eeckhoff
Eeckhoff·1999 ○ *Contractibility of Regular Polygons.*
Iowa State University · MATH 599 · Creative Component · 1999

Euclid / Johan Ludvig Heiberg / Richard Fitzpatrick
Euclid·2008 ○ *Euclid's Elements of Geometry.*
The Greek text of "Euclidis Elementa, edidit et Latine interpretatus est I.L. Heiberg, in aedibus B.G. Teubneri, 1883–1885" · edited, and provided with a modern English translation by Richard Fitzpatrick · isbn 978-0615179841 · 2008

F

Colin Faber / Félix Candela
Faber/Candela·1963 ◈ *Candela : The Shell Builder.*
London · The architectural Press · 1963

Seibert Fairman / Chester S. Cutshall
Fairman/...·1932 ■ *Graphic Statics.*
McGraw-Hill Book Company · New York and London · 1932

Antonio Favaro,
Favaro·1879 ■ *Leçons de statique graphique.*
traduites de l'italien par Paul Terrier · Gauthier-Villars · Paris · 1879

Corentin Fivet / Denis Zastavni
Fivet/Zastavni·2012 ◈ *The Salginatobel bridge design process by Robert Maillart (1929).*
Journal of the International Association for shell and spatial structures · issue 53 · pages 39-48 · 2012

Corentin Fivet
Fivet·2012 ■ *Recherche des modalités permettant l'étude des assemblages en bois par manipulation graphique de champs de contraintes discontinus.*
certification thesis "Le Bois dans la construction" · UCLouvain · Belgium · May 2012

Aita Flury / C. Baumberger / E. Boesch / M. Boesch / R. Boltshauser / J. Conzett / C. Galmarini / A. Hagmann / P. Kahlfeldt / A. Krischanitz / M. Monotti / A. Muttoni / C. Penzel / M. Peter / M. Pogacnik / S. Polonyi / U.B. Roth / M. Schlaich / R. Salvi / H. Schnetzer / J. Schwartz / J. Solt / Y. Weinand / A. Wiegelmann / C. Wieser
Flury/...·2012 ◈ *Cooperation: The Engineer and the Architect.*
edited by Aïta Flury · Birkhäuser · Basel · 2012

Émile Foulon
Foulon·1969 ■ *Les polygones funiculaires gauches et leurs applications au calcul des constructions à trois dimensions.*
Université de Liège · Faculté des sciences appliquées ULg · 1969

Marc Freixas / Robert Joan-Arinyo / Antoni Soto-Riera
A Constraint-Based Dynamic Geometry System. ◍ Freixas/...·2008
Journal of Computer-Aided Design · volume 42 · issue 2 · pages 151-161 · February 2010

François Frey
Analyse des structures et milieux continus - Mécanique des Structures. ◪ Frey·2000
Traité de Génie Civil de l'Ecole polytechnique fédérale de Lausanne · volume 2 · presses polytechniques et universitaires romandes · deuxième édition · isbn: 978-2880744342 · 2000

François Frey / Jaroslav Jirousek
Analyse des structures et milieux continus - Mécanique des Solides. ◪ Frey/Jirousek·2001
Traité de Génie Civil de l'Ecole polytechnique fédérale de Lausanne · volume 6 · presses polytechniques et universitaires romandes · isbn: 2880744636 · 2012

François Frey
Analyse des structures et milieux continus - Statique appliquée. ◪ Frey·2005
Traité de Génie Civil de l'Ecole polytechnique fédérale de Lausanne · volume 1 · presses polytechniques et universitaires romandes · deuxième édition · 2005

François Frey
Analyse des structures et milieux continus - Mécanique des Solides. ◪ Frey·2012
Traité de Génie Civil de l'Ecole polytechnique fédérale de Lausanne · volume 3 · presses polytechniques et universitaires romandes · deuxième édition · isbn: 978-2889150090 · 2012

Ioannis Fudos
Constraint Solving for Computer-Aided Design. ◍ Fudos·1995
doctoral thesis · directed by Christoph M. Hoffmann · Purdue University · August 1995

G

Maria E. Moreyra Garlock / David P.Billington
Felix Candela: Engineer, Builder, Structural Artist. ◈ Garlock/...·2008
Yale University Press · 2008

W. Gellert / H. Küstner / M. Hellwich / H. Kästner
Petite encyclopédie des mathématiques. ○ Gellert/...·1986
second french edition · translation directed by J.L. Lions · éditions K.Pagoulatos · Paris London Athens · 1986

Rolf Gerhardt / Karl-Eugen Kurrer / Gerhard Pichler
The methods of graphical statics and their relation to the structural form. ■ Gerhardt/...·2003
First International Congress on Construction History · edited by S. Huerta Fernandez & I. Juan de Herrera · SEdHC · ETSAM · Madrid · 2003

Sherif Ghali
Introduction to Geometric Computing. ◍ Ghali·2008
Springer-Verlag · London · isbn 978-1848001145 · 2008

Steven Givant / Paul Halmos
Introduction to Boolean Algebras. ○ Givant/Halmos·2009
Undergraduate Texts in Mathematics · Springer · 2009

Simon Greenwold / Edward Allen
Active Statics. ◪ Greenwold·2009
http://acg.media.mit.edu/people/simong/statics/data/ · visited in 2013

H

R. L. Hankinson
Hankinson·1921 ◪ *Investigation of Crushing Strength of Spruce at Varying Angles of Grain.*
Air Service Information Circular No.259 · U.S. Air Service · 1921

Ross Harmes / Dustin Diaz
Harmes/Diaz·2008 △ *Pro JavaScript Design Patterns.*
Apress · isbn 978-1430204954 · 2008

John M. Harris / Jeffry L. Hirst / Michael J. Mossinghoff
Harris/...·2008 ◪ *Combinatorics and Graph Theory.*
Undergraduate Texts in Mathematics · edited by S. Axler & K.A. Ribet
second edition · Springer · isbn 978-0-387-797710-6 · 2008

Robin Hartshorne
Hartshorne·2000 ○ *Geometry: Euclid and Beyond.*
Undergraduate texts in Mathematics · edited by S. Axler, F.W. Gehring &
K.A. Ribet · Springer · NewYork Berlin Heidelberg · isbn 0387986502 · 2000

W.S. Hemp
Hemp·1973 ◪ *Optimum Structures.*
Oxford University Press · isbn 978-0198561101 · 1973

Lebrecht Henneberg
Henneberg·1911 ■ *Die Graphische statik der starren systeme.*
B.G. Teubner · Leipzig Berlin · 1911

Gustav Herrmann
Herrmann·1892 ■ *The Graphical Statics of Mechanism.*
second edition · D. Van Nostrand Company · NewYork · 1892

David Hestenes / Garret Sobczyk
Hestenes/...·1987 ○ *Clifford Algebra to Geometric Calculus:*
A Unified Language for Mathematics and Physics.
Fundamental Theories of Physics · D.Reidel Publishing Company ·
Dordrecht Boston Lancaster Tokyo · isbn 9027716736 · 1987

Jacques Heyman
Heyman·1995 ◪ *The Stone Skeleton: Structural Engineering of Masonry Architecture.*
Cambridge University Press · 1995

Jacques Heyman
Heyman·1996 ◪ *Structural Analysis, A historical approach.*
Cambridge University Press · 1996

Jacques Heyman
Heyman·1999a ◪ *Navier's straitjacket.*
Architectural Science Review · volume 42 · issue 2 · pages 91-95 · 1999

Jacques Heyman
Heyman·1999b ◪ *The Science of Structural Engineering.*
Londres : Imperial College Press · 1999

Jacques Heyman
Heyman·2008a ◪ *Basic Structural Theory.*
Cambridge University Press · Cambridge · 2008.

Jacques Heyman
Heyman·2008b ◪ *Elements of stress Analysis.*
Cambridge University Press · Cambridge · 2008

R. B. Heywood
Photoelasticity for designers. ◪ Heywood·1969
Pergamon Press · Oxford London Edinburgh NewYork Toronto Sydney Paris Braunschweig · 1969

Marta Hidalgo / Robert Joan-Arinyo
Computing Parameter Ranges in Constructive Geometric Constraint Solving: Implementation and Correctness Proof. ◍ Hidalgo/...·2012
Journal of Computer-Aided Design · volume 44 · issue 7 · pages 709-72 · June 2012

David Hilbert
The Foundations of Geometry. ○ Hilbert·1902
translated by E.J.Townsend · The Open Court Publishing Company · Chicago · 1902

Rolland T. Hinkle
Kinematics of Machines. ◎ Hinkle·1953
Englewood Cliffs · N.J. Prentice-Hall, Inc. · 1953

C.M. Hoffmann / K.-J. Kim
Towards valid parametric CAD models. ◍ Hoffmann/Kim·2001
Journal of Computer-Aided Design · issue 33 · pages 81-90 · 2001

Christoph M. Hoffmann / Robert Joan-Arinyo
A Brief on Constraint Solving. ◍ Hoffmann/...·2005
Computer-Aided Design and Applications · volume 2 · number 5 · pages 655-663 · 2005

Andrée Corvol-Dessert / Patrick Hoffsummer / David Houbrechts / Georges-Noël Lambert / Catherine Lavier / Christine Locatelli / Jannie Mayer / Jean-Daiel Pariset / Alain Prévet / Jean-Louis Taupin / Yvonne Trenard
Les charpents du XI° au XIX° siècle, typologie et évolution en France du Nord et en Belgique. ◈ Hoffsummer/...·2002
Centre des monuments nationaux · Monum · éditions du patrimoine · Paris · isbn 2858223033 · 2002

Markus Hohenwarter / Michael Borcherds
Geogebra, free mathematics software for learning and teaching. ◍ Hohenwarter/...·2002
http://www.geogebra.org · visited on July 25, 2012 · first released in 2002

Judith Hohenwarter / Markus Hohenwarter
Introduction to Geogebra4 - Geogebra workshop handout. ◍ Hohenwarter/...·2012
available on http://www.geogebra.org/book/intro-en.pdf · April 12, 2012.

Alan Holgate
The art in Structural Design. ◈ Holgate·1986
Oxford University Press · 1986 · available on http://home.vicnet.net.au/~aholgate/structdes/taisd_anchor.html · last visited on june 2013

Alan Holgate
Structural Design as an Academic Discipline. ◈ Holgate·1991
first published in Structural Engineering Review · volume 3 · issue 3 · pages 147-156 · september 1991 · available on http://home.vicnet.net.au/~aholgate/structdes/papers/design_discipline.html · last visited on june 2013

Alan Holgate
The Art of Structural Engineering: The Work of Jorg Schlaich and His Team. ◈ Holgate·1997
Axel Menges · 1997

AUDUN HOLME
Holme·2010 ○ *Geometry, Our Cultural Heritage.*
second edition · Springer · isbn 978-3642144400 · 2010

CAY S. HORSTMANN & GARY CORNELL
Horstmann/...·2000 ◬ *Au cœur de Java 2: volume 2, Fonctions avancées.*
Collection Référence · CampusPress · isbn 2744008818 · 2000

CAY S. HORSTMANN & GARY CORNELL
Horstmann/...·2001 ◬ *Au cœur de Java 2: volume 1, Notions fondamentales.*
second edition · Collection Référence · CampusPress · isbn 2744011185 · 2001

SANTIAGO HUERTA FERNANDEZ
Huerta·2006a ◈ *Structural design in the work of Gaudí.*
Architectural Science Review · volume 49 · issue 4 · pages 324-339 · 2006

SANTIAGO HUERTA FERNANDEZ
Huerta·2006b ◪ *Galileo was Wrong : The Geometrical Design of Masonry Arches.*
Nexus Network Journal · volume 8 · issue 2 · pages 25-52 · 2006

SANTIAGO HUERTA FERNANDEZ
Huerta·2006c ◪ *Geometry and equilibrium: The gothic theory of structural design.*
Structural Engineer · 84 · pages 23-28 · 2006

SANTIAGO HUERTA FERNANDEZ
Huerta·2010 ◪ *Designing by Geometry: Rankine's Theorems of Transformation of Structures.*
Geometria and Proportion in Structural Design - Essays in Ricardo Aroca's Honour · edited by Pepa Cassinello, Santiago Huerta Fernandez, José Miguel de Prada Poole & Ricardo Sanchez Lampreave · Madrid · 2010

I

TULLIA IORI
Iori·2009 ◈ *Pier Luigi Nervi.*
Motta Architettura · Milan · 2009

J

L. JACOB / MAURICE D'OCAGNE
Jacob/dOcagne·1911 ◎ *Le calcul mécanique, appareils arithmétiques et algébriques, intégrateurs.*
encyclopédie scientifique publiée sous la direction du Dr Toulouse · Bibliothèque de Mathématiques Appliquées · Octave Doin & Fils · Paris · 1911

FLEEMING JENKIN
Jenkin·1869 ■ *On the practical application of reciprocal figures to the calculation of strains of frameworks.*
transactions of the Royal society of Edinburgh · volume XXV · pages 441-447 & plates XVII-XXI · 1869

ROBERT JOAN-ARINYO
Joan-Arinyo·2009 ◍ *Basics on Geometric Constraint Solving.*
EGC'09: XIII Encuentros de Geometria Computacional · Zaragoza · Spain · 2009

PHILIP JODIDIO
Jodidio·2007 ◈ *Santiago Calatrava.*
Taschen · isbn 978-3822848739 · 2007

K

ALFRED BRAY KEMPE
How to draw a straight line, a lecture on linkages. ◎ Kempe·1877
Nature Series · Macmillan & Co · London · 1877

ZUBIN KHABAZI
Generative algortihms using Grasshopper. ◪ Khabazi·2010
2010

F. KLEIN / F. TÄGERT
Famous Problems of Elementary Geometry: the duplication of the cube, the trisection of an angle, the quadrature of the circle. ○ Klein/Tägert·1897
an authorized translation of "Vorträge über ausgewählte fragen der elementargeometrie" · by Wooster Woodruff Beman & David Eugene Smith · Ginn and Company · Boston New-York Chicago London Atlanta Dallas Columbus SanFrancisco · 1897

FRANZ KNOLL / THOMAS VOGEL
Design for Robustness. ◪ Knoll/Vogel·2009
Structural Engineering Documents · number 11 · published by IABSE (International Association for Bridge and Structural Engineering) · 2009

DON KOBERG / JIM BAGNALL
The Universal Traveller, a Soft-Systems Guide to Creativity, Problem-Solving, and the Process of Reaching Goals. ◪ Koberg/Bagnall·1991
revised edition · New Horizons Edition · California · isbn 1560520450 · 1991

ADAM ADAMANDI KOCHANSKI
Observationes Cyclometricae ad facilitandam Praxin accomodatae. ○ Kochanski·1685
Acta Eruditorum · number 4 · pages 394-398 · 1685

ADAM ADAMANDI KOCHANSKI / HENRYK FUKS
Observationes Cyclometricae, latin text with annotated English translation. ○ Kochanski/Fuks·2011
arXiv:1106.1808v1 · 2011

AUGUST E. KOMENDANT
Dix-huit années avec Louis I Kahn. ◈ Komendant·2006
translated by Mathilde Bellaigue · preface by Louis Kahn · introduction by Bernard Marrey · Éditions du Linteau · Paris · 2006

ULRICH KORTENKAMP
Foundations of Dynamic Geometry. ◍ Kortenkamp·1999
PhD Thesis · directed by Prof. Jürgen Richter-Gebert & Prof. Günter M. Ziegler · ETH Zürich · 1999

ULRICH KORTENKAMP
Cinderella. ◪ Kortenkamp·2013
software · http://www.cinderella.de · visited in 2013

NEVEN KOSTIC
Topologie des champs de contraintes pour le dimensionnement des structures en béton armé. ◪ Kostic·2009
PhD thesis directed by Aurelio Muttoni · Ecole Polytechnique Fédérale de Lausanne · 2009

ELKE KARSNY
The Force Is in the Mind, The Making of Architecture. ◈ Krasny·2008
Architekturzentrum Wien · Birkhäuser · Basel Boston Berlin · isbn 978-3764389802 · 2008

L

Lorenz Lachauer / Hauke Jungjohann / Toni Kotnik
Lachauer/...·2011a ■ *Interactive parametric tools for structural design.*
IABSE-IASS symposium · "Taller, Longer, Lighter" · London · 2011

Lorenz Lachauer / Toni Kotnik
Lachauer/...·2011b ■ *Curved Bridge Design.*
Design Modeling Symposium · pages 145-152 · Springer · Berlin · 2011

Pierre Latteur
Latteur·1997 ◪ *Calculer une structure - de la théorie à l'exemple.*
3ème édition · Academia Bruylant · 1997

Pierre Latteur
Latteur·1998 ◪ *ISSD : An Interactive Software for Structural Design.*
http://www.issd.be · available since 1998

Jean-Marie Laborde / Max Marcadet
Laborde/...·2002 ◍ *Cabri Geometry II Plus.*
Cabrilog Company · http://www.cabri.com/ · visited on July 25, 2012 ·
first released in 2002

Jean-Marie Laborde / Max Marcadet
Laborde/...·2007 ◍ *Cabri II Plus — User Manual, Advanced & Reference.*
downloaded from http://www.cabri.com/download-cabri-2-plus.html ·
last visited on July 25, 2012 · 2007

Emile Michel Hyacinthe Lemoine
Lemoine·1888 ○ *De la mesure de la simplicité dans les constructions géométriques.*
Comptes rendus hebdomadaires des séances de l'académie des sciences ·
book 107 · pages 169-171 · July/December, 1888

Emile Michel Hyacinthe Lemoine
Lemoine·1902 ○ *Géométrographie ou art des constructions géométriques.*
Scientia - Physique/Mathématique · number 18 · Paris · february, 1902

A.S. Levens
Levens·1975 ◪ *Graphical methods in Research.*
R. E. Krieger · isbn 978-0882753164 · 1975

Maurice Lévy
Lévy·1874 ■ *La Statique graphique et ses applications aux constructions.*
Texte & Atlas · Gauthiers-Villars · Paris · 1874

Maurice Lévy
Lévy·1886 ■ *La Statique graphique et ses applications aux constructions.*
4 books · Gauthier-Villars · Paris · 1886-1888

Ferdinand von Lindemann
vonLindemann·1882 ○ *Ueber die Zahl p.*
Mathematische Annalen · volume 20 · number 2 · pages 213-225 · 1882

Jen Y. Liu
Liu·2001 ◪ *Strength Criteria For Orthotropic Materials.*
ICCE/8 : 8th Annual International Conference on Composite Engineering ·
Augustus 5-11, 2001

André Loits
Loits·2010 ◪ *Architecture civile (rénovation, restauration et technologie de l'architecture).*
lesson notes · 2010

JIM LOY
Geometry: Collapsible Compasses. ○ Loy·2012
website http://www.jimloy.com/geometry/collapse.html · visited on June 19th, 2012 · first published in 2000

JIM LOY
Trisection of an Angle. ○ Loy·2003
http://www.jimloy.com/geometry/trisect.htm · visited on July 5th, 2012 · edited in 1997 & 2003.

M

DAVID MAKINSON
Sets, Logic, and Maths for Computing. ◪ Makinson·2012
Undergraduate Topics in Computer Science · series edited by Ian Mackie · second edition · Springer · London Dordrecht Heidelberg NewYork · isbn 978-1447124993 · 2012

ELENA ANNE MARCHISOTTO / JAMES T. SMITH
The Legacy of Mario Pieri in Geometry and Arithmetic. ○ Marchisotto/...·2007
Birkhäuser · Boston Basel Berlin · isbn 978-0817632106 · 2007

PETER MARTI
Basic tools of reinforced concrete beam design. ◪ Marti·1985
ACI Journal · number 82-4 · January-February 1985

LORENZO MASCHERONI
La Geometria del Compasso. ○ Mascheroni·1797
Pavia · presso gli Eredi di Pietro Galeazzi · 1797

PASCAL MATHIS / SIMON E.B. THIERRY
A formalization of geometric constraint systems and their decomposition. ◍ Mathis/Thierry·2010
Formal Aspects of Computing · number 22 · pages 129-151 · 2010

JAMES CLERK MAXWELL
On Reciprocal Figures and Diagrams of Forces. ■ Maxwell·1864
Philosophical Magazine · volume 27 · number 4 · pages 250-261 · 1864

JAMES CLERK MAXWELL
On the application of the theory of reciprocal polar figures to the construction of diagrams of forces. ■ Maxwell·1867
The Engineer · issue 24 · 1867

JAMES CLERK MAXWELL
On Bow's method of drawing diagrams in graphical statics. ■ Maxwell·1876
Proceedings of the Cambridge Philosophical Society · issue 2 · pages 407-414 · 1876

JAMES CLERK MAXWELL / PETER MICHAEL HARMAN
The scientific letters and papers of James Clerk Maxwell. ◪ Maxwell/...·1995
edited by Peter Michael Harman · two volumes · Cambridge University Press · 1995

BENJAMIN MAYOR
Cours de Statique Graphique. ■ Mayor·1909
École polytechnique fédérale de Lausanne · 1906-1909

BENJAMIN MAYOR
Introduction à la statique graphique des systèmes de l'espace. ■ Mayor·1926
Librairie Payot & Cie · Lausanne Genève Neuchatel Vevey Montreux Berne · 1926

J. MICHAEL MCCARTHY / GIM SONG SOH
McCarthy/...·2011 ◪ *Geometric Design of Linkages.*
Interdisciplinary Applied Mathematics · edited by S.S. Antman, J.E. Marsden & L. Slrovich · Springer · isbn 978-1441978912 · 2011

JACK C. MCCORMAC / JAMES K. NELSON, JR.
McCormac/...·2003 ◪ *Structural Steel Design, LRFD Method.*
third edition · Pearson Education · New Jersey · isbn 0130479594 · 2003

JAMES A. MCHUGH
McHugh·1990 ◪ *Algorithmic Graph Theory.*
Prentice Hall · 013236152 · 1990

ANDREA MICHELETTI
Micheletti·2008 ◪ *On generalized reciprocal diagrams for self-stressed frameworks.*
International Journal of Space Structures · volume 23 · issue 3 · pages 153-166 · 2008

CHRISTIAN OTTO MOHR
Mohr·1886 ■ *Eine Aufgabe der graphischen Statik.*
Der Civilingenieur · pages 535-538 · 1886

GASPARD MONGE
Monge·1788 ◪ *Traité élémentaire de statique à l'usage des écoles de la marine.*
chez Baudouin · Paris · 1788

MOHSEN MOSTAFAVI / JÜRG CONZETT / BRUNO REICHLIN
Mostafavi/...·2003 ◈ *Structure as Space - Engineering and Architecture in the Works of Jürg Conzett.*
Architectural Association Publications · 2003.

AURELIO MUTTONI / JOSEPH SCHWARTZ / BRUNO THÜRLIMANN
Muttoni/...·1997 ◪ *Design of concrete structures with stress fields.*
Birkhäuser · 1997

AURELIO MUTTONI
Muttoni·2005 ◪ *L'art des structures, une introduction au fonctionnement des structures en architecture.*
Presses polytechniques et universitaires romandes · 2005

AURELIO MUTTONI / MIGUEL FERNANDEZ RUIZ / NEVEN KOSTIC
Muttoni/...·2011 ◪ *Champs de contraintes et méthode des bielles-et-tirants : Applications dans la conception et le dimensionnement des structures en béton armé.*
Syllabus · Laboratoire de construction en béton IBETON · Faculté de l'environnement naturel, architectural et construit ENAC · Ecole polytechnique fédérale de Lausanne EPFL · January 2011

N

JULIEN NARBOUX
Narboux·2007 ○ *Mechanical Theorem Proving in Tarski's Geometry.*
Conference on Automated Deduction in Geometry 2006 · edited by F. Botana & T. Recio · Springer-Verlag · Berlin Heidelberg · pages 139-156 · 2007

JULIUS NATTERER / JEAN-LUC SANDOZ / MARTIAL REY
Natterer/...·2000 ◪ *Construction en bois: Matériau, technologie et dimensionnement.*
Traités de Génie Civil de l'Ecole Polytechnique fédérale de Lausanne · volume 13 (W-0021) · isbn: 978-2880744008 · 2000

Pier Luigi Nervi
Savoir construire. ◈ Nervi·1997
2° edition · traduction de l'italien par Muriel Gallot · éditions du Linteau · Paris · isbn 2910342069 · 1997

Laurent Ney / Sigrid Adriaenssens / Stefan Devoldere / Iwan Strauven
Laurent Ney - Shaping Forces. ◈ Ney/...·2010a
Bozar Books · A+ Editions · 2010

Laurent Ney
Interview. ◈ Ney·2010b
Les cahiers de l'architecture · ISA St-Luc Tournai · 2010

Paul Nicholas
Approaches to Interdependency: early design exploration across architectural and engineering domains. ◈ Nicholas·2008
PhD Thesis · School of Architecture and Design · Design and Social Context Portfolio · RMIT University · 2008

Manfredi Nicoletti
Sergio Musmeci: Organicità di forme e forze nello spazio. ◈ Nicoletti·1999
Testo & Immagine · isbn 978-8886498647 · 1999

Arthur H. Nilson / David Darwin / Charles W. Dolan
Design of Concrete Structures. ◪ Nilson/...·2004
thirteenth edition · McGraw-Hill · isbn 0072483059 · 2004

C. B. Norris
Strength of Orthotropic Materials Subjected to Combined Stresses. ◪ Norris·1955
U.S. Forest Products Lab. Rep.1816 · FPL · Madison, WI · USA · 1955

O

John Ochsendorf
Practice before theory: the use of the lower bound theorem in structural design from 1850-1950. ◈ Ochsendorf·2005
Essays: the history of the theory of structures · pages 353-366 · 2005

R. Ogniewicz / M. Ilg
Voronoi Skeletons: Theory and Applications. ○ Ogniewicz/Ilg·1992
Conference on Computer Vision and Pattern Recognition · Champaign · Illinois · USA · 1992

Jacques Ozanam
L'usage du Compas de Proportion. ○ Ozanam·1691
Henry van Bulderen · La Haye · 1691

P

Fabrizio Palmisano / Amedeo Vitone / Claudia Vitone
A first approach to optimum design of cable-supported bridges using load path method. ◪ Palmisano/...·2008
Structural Engineering International · issue 4 · pages 412-420 · 2008.

Andrew Payne / Rajaa Issa
Grasshopper Primer. ◪ Payne/Issa·2009
second edition · 2009

Giuseppe (Ioseph) Peano
Peano·1889 ○ *Arithmetices principia, nova methodo exposita.*
Augustae Taurinorum Ediderunt Fratres Bocca · Roma Florentiae · pages 83–97 · 1889

Johann Friedrich Penther
Penther·1749 ○ *Praxis Geometriae.*
Berlegt von Johann Balthazar Probst · Augspurg · 1749

David Peyceré / Gilles Ragot / Guy Lambert / Joseph Abraham / Maurice Culot / Réjean Legault / Simon Texier
Peyceré/...·2000 ◈ *Les Frères Perret - L'œuvre complète.*
Editions Norma · isbn 978-2909283333 · 2000

Styliane Philippou
Philippou·2008 ◈ *Oscar Niemeyer, curves of irreverence.*
Yale University Press · NewHaven London · isbn 978-0300120387 · 2008

Antoine Picon
Picon·1988 ◈ *Architectes et ingénieurs au Siècle des lumières.*
Parenthèses · isbn 978-2863640494 · 1988

Antoine Picon / A. Chassagnoux / A.C. Webster / A. Föhl / A. Guiheux / A.G. Davenport / A.-L. Carré / A. Quénelle / A. Slaton / A.W. Skempton / B. Addis / B. Burkhardt / B. Forster / L.B. Lemoine / B.L. Hurst / B. Marrey / B.Shapiro Comte / B. Vaudeville / C.A. Jones / C. Journet / C. Maniaque / C.R. Blackwell / C. Schädlich / C. Simonnet / C. Wamsler / D. Bernstein / D. de Haan / D.P. Billington / D. Rastorfer / D. Rouillard / D. Sugden / D. Van Zanten / E. Benvenuto / E. Campagnac / E.C. English / E. Gallo /E. Kranakis / E.L. Kemp / E.N. DeLony / E. Perez Belda / F. Fromonot / F. Newby / F. Otto / F. Périé / F. Seitz / G. Auer / G. Chanvillard / G. Delhumeau / G. Fenske / G. K. Dreicer / G. Morel Journel / G. Ribeill / G.R. Larson / H. Vérin / J.A. Fernández Ordóñez / J. Delannoy / J.-F. Blassel / J.J. Arenas de Pablo / J. Kerisel / J.-M. Delarue / J. Natterer / J. Ramón Navarro Vera / J. Schlaich / K. Bowie / K. Chatzis / K. Ishii / L. Bruneel / M. Casciato / M. Chrimes / M. Cotte / M.E. Pawley / M. Fordham / M. Kutterer / M.K. Deming / M. Levy / M. Melaragno / M.N. Bussell / M.S. Buenaventura / M. Talamona / M. Virlogeux / N. Montel / N. Nogue / N. Okabe / O. Cinqualbre / P.C. Papademetriou / P. McCleary / P. Pinon / P. Potié / R.A. Paxton / R. Graefe / R. Guidot / R.J. Mackay Sutherland / R. Legault / R. Motro / R.M. Vogel / R.N. Dent / R. Rouyer / R. Thorne / S.E. Wermiel / S. GaspériniCoiffet / S.G. Fedorov / S.W. Ksiazek / T. Day / T.F. Peters / T. Ruddock / V.L. Roberts / V. PiconLefebvre / W.C. Brown / W.I. Liddell / W. Meighörner / Z.S. Makowski
Picon/...·1997 ◈ *L'art de l'ingénieur, constructeur, entrepreneur, inventeur.*
sous la direction d'Antoine Picon · Centre Georges Pompidou · Le Moniteur · Paris · 1997

Mario Pieri
Pieri·1908 ○ *La Geometria Elementare istituita sulle nozioni di 'punto' e 'sfera'.*
1908

Daniel Piker
Piker·2010 ◪ *Kangaroo Physics.*
software · www.food4rhino.com · available since 2010

Albert Pirard
Pirard·1950 ■ *La Statique Graphique - Science introductive à l'art de construire.*
Imprimerie H. Vaillant-Carmanne · S.A. · Liège · 1950

Albert Pirard
Pirard·1960 ◪ *Traité d'hyperstatique analytique et graphique.*
Dunod · 1960

Jean-Victor Poncelet
Traité des propriétés projectives des figures : ouvrage utile à ceux qui s'occupent des applications de la géométrie descriptive et d'opérations géométriques sur le terrain. ○ Poncelet·1822
Bachelier · Paris · 1822

Christoph Pourtois
Pier Luigi Nervi: Architecture as Challenge. ◈ Pourtois·2011
edited by Cristiana Chiorino & Carlo Olmo · CIVA - Silvana Editoriale · 2011

Clemens Preisinger
Karamba. ◪ Preisinger·2013
software · http://www.karamba3d.com · visited in 2013

R

Johann Radon
Mengen konvexer Körper, die einen gemeinsamen Punkt enthalten. ○ Radon·1921
Mathematische Annalen · volume 83 · issues 1-2 · pages 113-115 · 1921

Srinivasa Ramanujan
Squaring the Circle. ○ Ramanujan·1913
Journal of the Indian Mathematical Society · number 5 · page 132 · 1913

Srinivasa Ramanujan
Modular Equations and Approximations to Pi. ○ Ramanujan·1914
quarterly Journal of Mathematics · number 45 · pages 350-372 · 1914

William John Macquorn Rankine
A manual of applied mechanics. ◪ Rankine·1858
Richard Griffin · London Glasgow · 1858

William John Macquorn Rankine
Principle of the Equilibrium of Polyhedral Frames. ■ Rankine·1864
The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science · Fourth Series · volume XXVII · issue 1 · page 92 · 1864

William John Macquorn Rankine
Diagrams of Forces in Frameworks. ■ Rankine·1870
Proceedings of the Royal Society of Edinburgh · issue 7 · pages 171-172 · 1870

Wolfgang Rautenberg
A Concise Introduction to Mathematical Logic. ○ Rautenberg·2010
third Edition · Springer · New York Dordrecht Heidelberg London · 2010

Casey Reas / Ben Fry
Processing, A Programming Handbook for Visual Designers and Artists. ◬ Reas/Fry·2007
The MIT Press · Cambridge, Massachusetts · isbn 978-0262182621 · 2007

C.J. Recordon
Solutions approchées de la trisection de l'angle et de la quadrature du cercle. ○ Recordon·1865
chez Gauthier-Villars à Paris · chez Delafontaine et Rouge à Lausanne · 1865

R.H.S. Reiser / A.C.R. Costa / G.P. Dimuro
First Steps in the Construction of the Geometric Machine Model. ◍ Reiser/...·2002
Tendencias em Matemàtica Aplicada e Computacional · volume 3 · number 1 · pages 183-192 · 2002

Aristides A.G. Requicha
Requicha·1980 ◍ *Representations for Rigid Solids: Theory, Methods, and Systems.*
Association for Computing Machinery · Computing Surveys · volume 12 · number 14 · December, 1980

John Resig
Resig·2006 △ *Pro JavaScript Techniques.*
Apress · isbn 978-1590597279 · 2006

Franz Reuleaux
Reuleaux·1876 ◎ *The Kinematics of Machinery - Outlines of a Theory of Machines.*
translated and edited by Alex. B. W. Kennedy · Macmillan & Co. · London · 1876

Franz Reuleaux
Reuleaux·1899 ◎ *The Constructor, a hand-book of machine design.*
translated by Henry Harrison Suplee · published by H.H. Suplee · NewYork · 1899

Peter Rice
Rice·1994 ◈ *An engineer imagines.*
Artemis · London · 1994

Jürgen Richter-Gebert / Henry Crapo / Ulrich H. Kortenkamp
Richter-Gebert/...·1998 ◍ *Cinderella - The Interactive Geometry Software.*
http://www.cinderella.de · visited on July 25, 2012 · first released in 1998

Jürgen Richter-Gebert
Richter-Gebert/...·2011 ○ *Perspectives on Projective Geometry, a Guided Tour Through Real and Complex Geometry.*
Springer · Heidelberg Dordrecht London New York · isbn 978-3642172854 · 2011

Jürgen Richter-Gebert / Ulrich H. Kortenkamp
Richter-Gebert/...·2012 ◍ *The Cinderella2 Manual — Working with the Interactive Geometry Software.*
Springer · Heidelberg Dordrecht London NewYork · isbn 978-3540349242 · 2012

Matthias Rippmann / Lorenz Lachauer / Philippe Block
Rippmann/...·2012 ◪ *Interactive Vault Design.*
International Journal of Space Structures · volume 27 · number 4 · pages 219-230 · 2012

RISA Technologies
Risa·1987 ◪ *RISA2D.*
software · www.risa.com/p_risa2d.html · since 1987

Xosé Manuel Carreira Rodríguez
Rodríguez·2010 ◈ *The 7 main causes of numerical rubbish.*
http://notonlybridges.blogspot.be/2010/01/7-main-causes-of-numerical-rubbish.html · published on January 16, 2010

Francesca Rossi / Peter van Beek / Toby Walsh
Rossi/...·2006 ◍ *Handbook of Constraint Programming.*
Foundations of Artificial Intelligence · series edited by J.Hendler, H.Kitano & B. Nebel ·Elsevier · Amsterdam Boston Heidelberg London NewYork Oxford Paris SanDiego SanFrancisco Singapore Sydney Tokyo · isbn 978-0444527264 · 2006

George I. N. Rozvany
Rozvany·2009 ◪ *A critical review of established methods of structural topology optimization.*
Struct Multidisc Optim · issue 37 · pages 217–237 · 2009

Miguel Fernández Ruiz / Aurelio Muttoni
Ruiz/Muttoni·2007 ◪ *On Development of Suitable Stress Fields for Structural Concrete.*
ACI Structural Journal · issue 104 · pages 495-502 · 2007

S

ANDREW SAINT
Architect and Engineer, a study in sibling rivalry. ◈ Saint·2007
Yale University Press · New Haven & London · isbn 978-0300124439 · 2007

PHILIPPE SAMYN
Étude de la morphologie des structures à l'aide des indicateurs de volume et de déplacement. ◪ Samyn·2004
Académie royale de Belgique · Classe des Sciences · 3rd serie · tome V · 2004

TOMOKO SAKAMOTO / ALBERT FERRÉ / MICHAEL MEREDITH / AGU / MUTSURO SASAKI / ADAMS KARA TAYLOR / DESIGNTOPRODUCTION / ARANDA-LASCH
From control to design, parametric/algorithmic architecture. ◪ Sakamoto/...·2008
Actar-D · isbn 978-8496540798 · 2008

MUTSURO SASAKI
Morphogenesis of Flux Structure. ◈ Sasaki·2007
Architectural Association Publications · London · isbn 978-1902902579 · 2007

ANUPA M SAXENA / BIRENDRA SAHAY
Computer Aided Engineering Design. ◪ Saxena/Sahay·2005
Springer · isbn 1402025556 · 2005

CHRISTOPH SCHEINER
Pantographice, seu ars delineandi res quaslibet per parallelogrammum lineare seu cavum, mechanicum, mobile : libellis duobus explicata, & demonstrationibus geometricis illustrata quorum prior epipedographicen, sive planorum, posterior stereographicen, seu solidorum aspectabilium vivam imitationem atque proiectionem edocet. ◎ Scheiner·1631
Romae · Ex typographia Ludouici Grignani, sumptibus Hermanni Scheus · 1631

E. SCHOLZ
Graphical Statics. ■ Scholz·1994
in Companion Encyclopedia of the History and Philosophy of the Mathematical Sciences · volume 2 · edited by I. Gratton-Guinness · Routledge · London · pages 978-993 · 1994

UWE SCHÖNING
Logic for Computer Scientists. ◪ Schöning·2008
Birkhäuser · Boston Basel Berlin · isbn 978-0817647629 · 2008

FRANCISCUS À SCHOOTEN
De Organica Conicarum Sectionum. ◎ Schooten·1646
Lugd. Batavor · ex Officina Elzeviriorum · 1646

JOSEPH SCHWARTZ
Structural Theory and Structural Design. ◈ Schwartz·2012
Cooperation: The Engineer and the Architect. edited by Aïta Flury · Birkhäuser · Basel · 2012

PIERLUIGI SERRAINO
Eero Saarinen (1910-1961), un expressioniste structurel. ◈ Serraino·2006
Taschen · Köln ·isbn 978-3822836446 · 2006

PHILIPPE SERRÉ / NABIL ANWER / JIANXIN YANG
On the Use of Conformal Geometric Algebra in Geometric Constraint Solving. ◍ Serré/...·2010
Guide to Geometric Algebra in Practice · 11th chapter · edited by Leo Dorst & Joan Lasenby · Springer · London Dordrecht Heidelberg New York · isbn 978-0857298102 · 2010

Kaleem Siddiqi / S. M. Pizer / J. N. Damon / P. J. Giblin / N. Amenta / G. Borgefors / S. Bouix / R. Broadhurst / E. Chaney / S. Choi / S. Dickinson / P. T. Fletcher / Q. Han / S. Joshi / B. B. Kimia / F. F. Leymarie / D. Macrini / I. Nyström / J. Shah / G. Sanniti di Baja / A. Shokoufandeh / G. Székely / M. Styner / T. Terriberry / A. Thall / P. Yushkevich / J. Zhang
Siddiqi/...·2008 ◍ *Medial Representations: Mathematics, Algorithms and Applications.*
edited by Kaleem Siddiqi & Stephen M. Pizer · Computational Imaging and Vision Series · series edited by Max Viergever · volume 37 · Springer · 2008

Cyrille Simonnet
Simonnet·2005 ◪ *Le béton, histoire d'un matériau.*
Editions Parenthèses · isbn 978-2863640913 · 2005

Károly Simonyi
Simonyi·2012 ◪ *A Cultural History of Physics.*
A.K. Peters/CRC Press · isbn 978-1568813295 · 2012

William R. Spillers / Keith M. MacBain
Spillers/...·2009 ◪ *Structural Optimization.*
Springer · Dordrecht Heidelberg London New York · 2009

STEP (Structural Timber Education Program): Patrick Racher / Jean-Pierre Biger / Frédéric Rouger / Gérard Sagot / Gilbert Vidon / Lucien Andriamitantsoa / Henri Teyssandier / Jean Trinh
STEP·1996 ◪ *Structures en Bois aux Etats Limites - Introduction à l'Eurocode 5.*
Step 1 Matériaux et bases de Calcul · SEDIBOIS · Union nationale française de charpente, menuiserie, parquets · isbn 978-2212118325 · 1996.

Simon Stevin / Albert Girard
Stevin/...·1634 ◪ *Les œuvres mathématiques de Simon Stevin de Bruges.*
le tout revu, corrigé et augmenté par Albert Girard · Leyde · chez Bonaventure et Abraham Elsevier · 1634

John Stillwell
Stillwell·2005 ○ *The Four Pillars of Geometry.*
Undergraduate Texts in Mathematics · Springer · isbn 978-0387255309 · 2005

Wolfram Schwabhäuser / Wanda Szmielew / Alfred Tarski
Schwabhäuser/...·2011 ○ *Metamathematische Methoden in der Geometrie.*
Berlin Heidelberg · Germany · 1983
Reprinted in New-York & Tokyo · Ishi Press International · isbn 978-4871877077 · November, 2011

Ryan E. Smith
Smith·2011 ◪ *Interlocking Cross-Laminated Timber: alternative use of waste wood in design and construction.*
BTES Conference · Convergen and Confluence · 2011

James Joseph Sylvester
Sylvester·1875a ◎ *On the plagiograph aliter the Skew pantigraph.*
Nature · page 168 · July 1st, 1875

James Joseph Sylvester
Sylvester·1875b ◎ *History of the Plagiograph.*
Nature · pages 214-216 · July 25, 1875

Leslaw W. Szczerba
Szczerba·1986 ○ *Tarski and Geometry.*
The Journal of Symbolic Logic · volume 51 · number 4 · pages 907-912 · December, 1986

T

Daina Taimina
Historical Mechanisms for Drawing Curves. ◎ Taimina·2007
Hands On History, edited by Amy Shell-Gellasch · MAA Notes · volume 72 · pages 89-104 · 2007

Alfred Tarski
What is Elementary Geometry. ○ Tarski·1959
Symposium on the axiomatic method · 1959

Alfred Tarski / Steven Givant
Tarski's System of Geometry. ○ Tarski/Givant·1999
The Bulletin of Symbolic Logic · volume 5 · number 2 · pages 175-214 · June, 1999

Karl Terzaghi
Theoretical soil mechanics. ◪ Terzaghi·1966
John Wiley and Sons · NewYork London Sydney · 1943 · fourteenth printing, 1966

Dominique Tournès
Du compas aux intégraphes : les instruments du calcul graphique. ◎ Tournès·2003
Repères · IREM · number 50 · pages 63-84 · January, 2003

Martin Trautz / Christophe Koj
Self-tapping screws as reinforcement for timber structures. ◪ Trautz/Koj·2009a
Evolution and Trends in Design, Analysis and Construction of Shell and Spatial Structures · Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium · Valencia · 2009

Martin Trautz / Christophe Koj
Mit Schrauben Bewheren 2. ◪ Trautz/Koj·2009b
Forschungsbericht 01 · TragKonstruktionen · RWTH Aachen University · 2009

Edward Tsang
Foundations of Constraint Satisfaction. ◍ Tsang·1995
first published by Academic Press Limited in 1993 · available on http://www.bracil.net/edward/FCS.html#FCS%20Download · 1995

U

G. Ungewitter / K. Mohrmann
Lehrbuch der Gotischen Konstruktionen. ◪ Ungewitter/...·1901
two volumes · Chr. Herm. Tauchnitz · Leipzig · 1901

V

Hilderick A. van der Meiden / Willem F. Bronsvoort
An efficient method to determine the intended solution for a system of geometric constraints. ◍ vanderMeiden/...·2005
International Journal of Computational Geometry and Applications · volume 15 · issue 3 · pages 279-298 · June 2005

Hilderick A. van der Meiden / Willem F. Bronsvoort
A constructive approach to calculate parameter ranges for systems of geometric constraints. ◍ vanderMeiden/...·2006
Journal of Computer-Aided Design · volume 38 · issue 4 · pages 275-283 · April, 2006

Tom Van Mele / Matthias Rippmann / Lorenz Lachauer / Philippe Block
VanMele/…·2012 ■ *Geometry-based Understanding of Structures.*
Journal of the International Association of Shell and Spatial Structures · volume 53 · issue 4 · pages 285-295 · 2012

Tom Van Mele / Philippe Block
VanMele/Block·2011 ■ *eQUILIBRIUM.*
http://block.arch.ethz.ch · available since 2011

Pierre Varignon
Varignon·1725 ◪ *Nouvelle mécanique ou statique, dont le projet fut donné en 1687.*
Ouvrage posthume de M. Varignon · 2 tomes · chez Claude Jombert · Paris · 1725

Remco C. Veltkamp
Veltkamp·1995 ◍ *A Quantum Approach to Geometric Constraint Satisfaction.*
Object-Oriented Programming for Graphics, Eurographics · edited by C. Laffra *et al.* · The European Association for Computer Graphics · 1995

Chiara Vernizzi
Vernizzi·2011 ◈ *Il disegno in Pier Luigi Nervi, dal dettaglio della materia alla percezione dello spazio.*
Ricerche di Rappresentazione e Rilievo dell'architecttura, della città e del territorio · Università degli Studi di Parma · Mattioli 1885 · Fidenza · isbn 978-8862612357 · 2011

W

Pierre Wantzel
Wantzel·1837 ○ *Recherches sur les moyens de reconnaître si un problème de géométrie peut se résoudre à la règle et au compas.*
Journal de Mathématiques Pures et Appliquées · volume 1 · issue 2 · pages 366–372 · 1837

Matthew Wells
Wells·2008 ◈ *Engineers: A History of Engineering and Structural Design.*
Routledge · isbn 978-0415325264 · 2008

Douglas Brent West
West·1996 ◪ *Introduction to Graph Theory.*
Prentice-Hall · isbn 0132278286 · 1996

Y

Robert C. Yates
Yates·1941 ◎ *Tools, A Mathematical Sketch and Model Book.*
Louisana State University · Baton Rouge · 1941

Robert C. Yates
Yates·1959 ◎ *A Handbook of Curves and Their Properties.*
J.W. Edwards & Ann Arbor · revised edition · 1959

Z

WACLAW ZALEWSKI / EDWARD ALLEN
Shaping Structures: Statics. ■ Zalewski·1997
Wiley & Sons · isbn 978-0471169680 · 1997

DENIS ZASTAVNI
La conception chez Robert Maillart : morphogenèse des structures architecturales. ◈ Zastavni·2008a
doctoral thesis · directed by Jean-François & Pascal De Beck · Ecole Polytechnique de Louvain · UCLouvain · June 3, 2008

DENIS ZASTAVNI
The structural design of Maillart's Chiasso Shed (1924): A graphic procedure. ◈ Zastavni·2008b
Structural Engineering International · volume 18 · number 3 · pages 247-252 · 2008

DENIS ZASTAVNI
Maillart's design methods and sustainable design. ◈ Zastavni·2009
33rd IABSE International Symposium Bangkok · Sustainable Infrastructure: Environment Friendly, Safe and Resource Efficient · Bangkok · 2009

DENIS ZASTAVNI
An equilibrium approach on a structural scale to structural design. ◪ Zastavni·2010
International Conference on Structures and Architecture · edited by P. Cruz · Guimarães · Portugal · 2010

ALEXANDER ZIWET / PETER FIELD
Introduction to Analytical Mechanics. ◪ Ziwet/Field·1912
MacMillan Company · New-York · 1912

KLAUS ZWERGER
Wood and wood Joints. ◪ Zwerger·2012
Birkhauser · 2012

And, more widely, all contributions visited on www.wikipedia.org, a valuable tool.

remerciements

Denis, je tiens à te remercier, chaleureusement, pour l'invitation à effectuer cette thèse, pour ta confiance et ton aide permanentes. Je te remercie également pour tous les enseignements reçus, en ce compris les compétences satellites qui font le quotidien du chercheur. Ton regard critique sur les méthodes de conception structurale restera pour moi l'enseignement majeur retenu durant ces quatre ans.

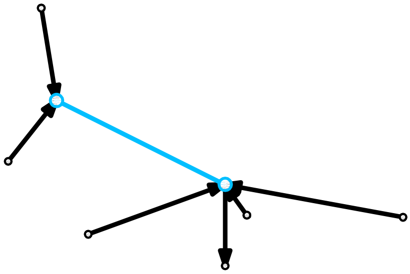
Jean-François, je te remercie pour tes commentaires précis et clairvoyants apportés à chaque entrevue.

Merci aux membres du jury et d'une manière générale, à tous les académiques et praticiens qui, à un moment ou un autre, ont manifesté leur intérêt pour cette recherche. Je pense notamment aux professeurs Aurelio Muttoni, John Ochsendorf, Laurent Ney, Pascal Lambrechts, Tullia Iori, à André Jasienski et Jean-François Denoël de la Febelcem, au jury de sélection du Hangai Prize de l'IASS, à Philippe Block, à Jurg Conzett, à Olivier Burdet et à Albert Mahy. Tous ces encouragements, même les plus succincts et les plus nuancés, me furent d'un grand soutien.

Je remercie le jury des Fonds Spéciaux de la Recherche de l'UCLouvain pour l'octroi et le renouvellement du financement de cette recherche.

Enfin, je souhaite citer dans ces lignes toutes celles et ceux qui ont composé mon environnement de travail: les chercheurs du bureau A.106 (Jean-Philippe, Gérald, Lee et J-P) et ses visiteurs réguliers; Olivier M. pour nos nombreuses excursions; le corps enseignant de la faculté LOCI pour les charges d'enseignement qui me furent confiées; le personnel administratif du Vinci pour leur soutien logistique; tous les ponctuels inclassables (Amandine, Vincent, *etc.*); et, last but not least, Claire pour toute l'énergie donnée à la relecture de mon anglais.

Ça y est Séréna, c'est fait !



The space diagram above represents a rod equilibrated by six forces.
The force diagram on the front cover represents their corresponding magnitudes.

Dragging the magenta point (on the front cover) updates the orientation of the cyan rod and its adjacent forces above, without affecting its static equilibrium.

This rod will also remain in compression provided the magenta point stays within the grey area, the boundaries of which are fixed by geometric rules relying solely on the positions of points.