

UNIVERSITÉ PIERRE ET MARIE CURIE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

THÈSE DE MASTER

Reconstruction numérique de mesures d'ablation en tube d'expansion en conditions de ré-entrée terrestre

Auteur:
Alexandre BEVILACQUA

Superviseurs:
Pénélope LEYLAND
Nikhil BANERJI

MASTER DE SCIENCES ET TECHNOLOGIES
MENTION SCIENCES DE L'INGÉNIEUR
ORIENTATION MÉCANIQUE DES FLUIDES



Année universitaire 2015/2016

Abstract

As part of the understanding of radiative coupling phenomena with ablation phenomena for hypersonic re-entry of Hayabusa or Stardust type of capsules, laboratory tests are very important. Only expansion tubes can reproduce flight conditions at these speeds but for very short testing time, which can prevent the inclusion of ablation. To overcome these difficulties, a capsule model in binary scale flight was manufactured by filling the surface of a pure pre-heated phenolic resin in situ on firing, in the test section of the X2 expansion tube University of Queensland. The numerical reconstruction of the tests is presented here to a terrestrial re-entry condition, with simplified models of radiation and ablation. Flow simulations are performed with Eilmer3 and CFD ++ codes, and modeling assumptions are outlined. The results were compared with experimental measurements in order to validate the modeling, including heat flow estimates on the vehicle, with or without ablative protection.

Résumé

Dans le cadre de la compréhension des phénomènes de couplage radiatif avec des phénomènes d'ablation pour la rentrée hypersonique des capsules de type Stardust ou Hayabusa, les essais en laboratoire sont très importants. Seuls les tubes d'expansion peuvent reproduire des conditions de vol à ces vitesses mais pour des temps d'essais très courts, et ceci peut empêcher la prise en compte de l'ablation. Pour palier ces difficultés, un modèle de capsule à l'échelle binaire de vol a été fabriqué en remplissant sa surface d'une résine phénolique pure pré-chauffée in situ au moment du tir, dans la section d'essai du tube d'expansion X2 de l'université de Queensland. La reconstruction numérique des essais est présentée ici pour une condition de re-entrée terrestre, avec des modèles simplifiés de rayonnement et d'ablation. Les simulations de l'écoulement sont effectuées avec les codes Eilmer3 et CFD++, et les hypothèses de la modélisation soulignées. Les résultats obtenus ont été comparés avec les mesures expérimentales afin d'en valider la modélisation avec notamment des estimations des flux de chaleur sur le véhicule, avec ou sans protection ablative.

Table des matières

1	Introduction	7
1.1	Exploration de l'espace	7
1.2	Ré-entrée atmosphérique	8
1.3	Plan	9
2	Travaux connexes	10
3	Modélisation	11
3.1	Problème physique	11
3.1.1	Régime hypersonique	11
3.1.2	Couche de choc	11
3.1.3	Chimie et transferts d'énergie	11
3.1.4	Ablation et Pyrolyse	12
3.2	Équations gouvernant les écoulements hypersoniques en ré-entrée	13
3.2.1	Équation de conservation	13
3.2.2	Quantités conservées	13
3.2.3	Équations de Navier-Stokes compressible	13
3.2.3.1	Modèle à un mode de température	13
3.2.3.2	Modèle à deux modes de température	15
3.3	Modélisation de l'état thermodynamique du gaz	16
3.3.1	Propriétés thermodynamiques et coefficients	16
3.3.2	Coefficients de transport	18
3.4	Processus de non-équilibre	19
3.4.1	Modèle chimique	19
3.4.2	Modèle thermique	22
3.5	Modélisation du rayonnement	23
3.5.1	Rayonnement thermique	23
3.5.2	Rayonnement du corps noir	23
3.5.3	Modèle de transport	25
3.5.4	Modèle spectral	26
3.6	Méthodes numériques utilisées	26
3.6.1	Discrétisation spatiale: Méthode de volumes finis	26
3.6.2	Discrétisation temporelle	27
3.6.3	Consistance, stabilité et convergence	29
3.7	Implémentation	29
3.7.1	Code Eilmer3	29
3.7.2	Code CFD++	31
3.7.3	Module de rayonnement	32
4	Simulation, modèle "X2" (procédure)	33
4.1	Géométrie conditions aux bords	33
4.2	Conditions initiales	34
4.3	Maillages 2D	34
4.3.1	Maillage Eilmer3	35
4.3.2	Maillage ICEM utilisé pour CFD++	37
4.4	Définition du gaz	38
4.4.1	Choix du modèle de gaz	38
4.4.2	Réactions chimiques et paramètres de réaction	38

4.4.3	Échange d'énergie	39
4.4.4	Modèle de diffusion	39
4.4.5	Configuration du modèle de transport de rayonnement	40
4.5	Exécution d'une simulation	40
4.5.1	Lancement	40
4.5.2	Convergence itérative	41
4.6	Influence du maillage (Eilmer3)	41
4.6.1	Convergence spatiale	41
4.6.2	Influence de la forme des cellules	44
5	Résultats	46
5.0.1	Eilmer3: Solution visqueuse (11 espèces)	46
5.0.2	Eilmer3: Solution visqueuse avec ablation et rayonnement (20 espèces)	46
5.0.3	CFD++: Solution visqueuse (modèle 20 espèces) sans ablation et sans rayonnement	48
6	Discussion des résultats	49
6.1	Distance de choc	49
6.2	Comportement modal	50
6.3	Comportement chimique	51
6.3.1	Répartition des espèces	51
6.3.2	Ajout d'espèces ablatives	51
6.4	Comparaison Eilmer3 - CFD++	54
6.4.1	Comportement des flux de chaleur et impact sur la surface	55
6.5	Rayonnement (Eilmer3)	56
6.5.1	Rayonnement du plasma le long de la ligne d'arrêt (Plans tangents)	56
6.5.2	Nombre de Goulard: couplage du rayonnement au champ d'écoulement	57
6.5.3	Rayonnement reçu par la surface de la capsule (Simulation Monte-Carlo)	57
6.5.4	Identification spectrale des espèces et comparaisons (Plans tangents)	58
7	Conclusion	61
7.1	Grandes lignes	61
7.2	Travail futur	61
A	Annexes	66
A.1	Expression des vecteurs utilisés dans Eilmer3	66
A.2	Processus de résolution d'Eilmer3	68
A.3	Commandes et procédure pour un calcul Eilmer3 sous Bellatrix	69
A.4	Fichiers matlab	72
A.5	Script Python	96

Liste des figures

1	Représentation artistique d'Hayabusa exerçant sa mission d'extraction sur l'astéroïde Itokawa.	8
2	Le bouclier thermique du rover Opportunity couché sur le dos à gauche sur la surface de Mars. Par NASA / JPL / Cornell	9
3	Schéma du tube à expansion X2 de l'université de Queensland (sans être à l'échelle) issu de [4]	10
4	Schéma des processus aéro-thermiques se produisant dans la couche de choc et sur la surface d'une capsule en ré-entrée hypersonique [35]	12
5	Spectre électromagnétique	23
6	Loi de Planck	24
7	Dimensions de la capsule en (mm)	33
8	Schéma du domaine et des conditions limites en amont de la capsule	33
9	Maillage Eilmer3 avec épaulement	35
10	Maillage Eilmer3 sans épaulement	36
11	Maillage Icem avec épaulement	37
12	Comparaison des profils de température et de flux de chaleur pour différents raffinages (nb = 40, 60, 90), maillages Eilmer3 avec épaulements, conditions initiales 2.	42
13	Extrapolation de Richardson	43
14	Tracé des flux de conduction le long de la surface du modèle pour les différents maillages avec épaulement d'Eilmer3, conditions initiales 2	45
15	Maillages et formes des cellules proches de la ligne de stagnation.	45
16	Champ de vitesse en amont de la capsule, calculé par Eilmer3 (gaz à 11 espèces, conditions initiales 1)	46
17	Répartition de la température en amont de la capsule calculé par Eilmer3 (gaz à 20 espèces, conditions initiales 1)	46
18	Répartition de l'énergie en amont de la capsule, calculé par Eilmer3 (gaz à 20 espèces, conditions initiales 1)	47
19	Champs de vitesse en amont de la capsule, calculé par Eilmer3 (gaz à 20 espèces, conditions initiales 1)	47
20	Champ de pression en amont de la capsule, calculé par Eilmer3 (gaz à 20 espèces conditions initiales 1)	48
21	Champs de rayonnement en amont de la capsule, calculé par Eilmer3 (gaz à 20 espèces conditions initiales 1)	48
22	Champ de vitesse en amont de la capsule, calculé par CFD++ (gaz à 20 espèces, conditions initiales 2)	48
23	Image du petit modèle de capsule en résine phénolique lors des essais expérimentaux dans le tube X2 et distance de choc	49
24	Températures modales le long de la ligne d'arrêt obtenues par Eilmer3 avec et sans ablation (modèle à 20 espèces, conditions initiales 1)	50
25	Énergies modales sur la ligne d'arrêt obtenues par Eilmer3 avec ablation (modèle à 20 espèces, conditions initiales 1)	50
26	Fractions molaires à travers la couche de choc obtenues par Eilmer3 (sans protection ablative, modèle à 11 espèces, conditions initiales 1)	52
27	Fractions molaires à travers la couche de choc obtenues par Eilmer3 (avec protection ablative, modèle de Park 20 espèces, conditions initiales 1)	53

28	Comparaison des températures modales sur la ligne d'arrêt obtenues par CFD++ et Eilmer3 (sans d'ablation, conditions initiales 2)	54
29	Flux de chaleur par conduction (FCc) et par diffusion (FCd) le long de la surface du véhicule, obtenu par Eilmer3 avec ablation et sans ablation (modèle à 20 espèces, conditions initiales 1)	55
30	Comparaison de la divergence du flux radiatif par la méthode des plans tangents, dans le cas avec et sans ablation	56
31	Comparaison de flux radiatifs sur la surface du véhicule calculé par la méthode de tracé de rayon Monte-Carlo par le biais de PHOTAUORA dans le cas ablation et sans ablation	58
32	Identification des espèces sur le spectre entre 50 et 220 nm, obtenu par la méthode des plans tangents le long de la ligne d'arrêt, dans le cas avec et sans ablation	59
33	Identification des espèces sur le spectre entre 200 et 460 nm, obtenu par la méthode des plans tangents le long de la ligne d'arrêt, dans le cas avec et sans ablation	59
34	Identification des espèces sur le spectre entre 610 et 680 nm, obtenu par la méthode des plans tangents le long de la ligne d'arrêt, dans le cas avec et sans ablation	60
35	Identification des espèces sur le spectre entre 610 et 680 nm, obtenu par la méthode des plans tangents le long de la ligne d'arrêt, dans le cas avec et sans ablation	60

Liste des Tableaux

1	Expression de la température pour la formule d'Arrhenius, selon Park	21
2	Conditions de ré-entrée atmosphérique terrestre en tunnel d'expansion (Conditions 1)	34
3	Réactions chimiques considérées dans le modèle de Park (2001) [33], et paramètres de taux d'avancement associés	38
4	Condition limite d'ablation	38
5	Modélisation Air-TV Eilmer3	39
6	Modélisation Air-TE Eilmer3	39
7	Configuration de la simulation plans-tangents	40
8	Configuration de la simulation Monte-Carlo	40
9	Données utilisées pour l'étude de convergence spatiale	44
10	Méthode de Roache: étude de convergence spatiale (données calculées par le script nasagrdev.m)	44
11	Valeurs d'épaisseur de choc mesurées sur deux images (avec et sans ablation) de la vidéo obtenue lors des essais de ré-entrée de la capsule en résine phénolique dans le tube X2	49
12	Flux radiatif obtenu par la méthode des plans tangents et nombre de Goulard associé	57

1 Introduction

1.1 Exploration de l'espace

L'histoire de la conquête spatiale commence après la seconde guerre mondiale, où la technologie utilisée par les allemands pour l'envoi de missiles a été reprise par les États-Unis et l'Union soviétique. Afin de montrer le prestige de leur nation, les deux puissances se lancent dans une course vers l'espace. Les soviétiques furent les premiers à avoir mis en orbite autour de la terre un objet artificiel (le satellite "Spoutnik 1", le 4 octobre 1957) et à avoir envoyé un homme dans l'espace (cosmonaute Yuri Gagarin à bord du "Vostok 1", le 12 avril 1961). Les américains furent les premiers à envoyer des hommes sur la Lune (astronautes Amstrong/Collins/Aldrin lors de la mission Apollo 11, le 20 juillet 1969) et ont effectué des missions lunaires jusqu'en 1972.

Le lancement de la première station spatiale mondiale ("Salyut 1", 19 avril 1971) des soviétiques, suivie par la station des américains (Skylab, 14 mai 1973) marquèrent le début de la vie dans l'espace. Peu après, les deux concurrents finissent par réunir leur forces lors d'une mission commune symbolique (Apollo-Soyuz, 15 Juillet 1975) dont l'événement marqua le début d'une coopération internationale. De nouvelles stations de grande taille furent créées ensuite dans l'espace comme la station russe MIR (lancée le 19 Février 1986, assemblée dans l'espace jusqu'en 1996, opérationnelle jusqu'en 2001) et la station spatiale internationale (ISS, lancée le 20 novembre 1998 dont l'assemblage se finira en 2017). La navette spatiale américaine (en service entre 1981-2012) eut le rôle de lanceur pour la construction des ces stations; elle effectua aussi des missions scientifiques et militaires comme l'envoi de sondes, la mise en service de satellites ou du télescope spatial Hubble (développé par la NASA et l'ESA, déployé le 24 avril 1990 et toujours en état de marche).

Nombreuses sont les sondes à avoir quitté l'orbite terrestre afin d'explorer d'autres planètes. Certaines se placent sur leur orbite pour les observer, d'autres effectuent un survol. En 2016, les sondes les plus éloignées de la terre sont: "Voyager 1" se situant à plus de 20 milliards de km et "Voyager 2" à plus 16 milliards de km. Après leur lancement en 1977, elles sont passées au cours des années 1980 près de Jupiter, Saturne, Uranus, Neptune ainsi que 48 de leurs satellites naturels. Avec une vitesse de 62 000 km/h "Voyager 1" a franchi en 2012 l'héliopause (environ 120 fois la distance terre-soleil), frontière délimitant la zone d'influence du vent solaire et le milieu interstellaire. Les deux sondes continuent d'envoyer de précieuses informations sur leur environnement. Elles sont estimées pouvoir tenir jusque dans les années 2020 avant de dériver vers d'autres étoiles, transportant avec elles un disque d'or contenant des données sur la diversité de la vie et de la culture sur terre.

D'autres objets spatiaux ont aussi été conçus pour atterrir sur la surface de planètes ou de leurs lunes. La sonde soviétique "Venera 9" fut la première à renvoyer un photo panoramique depuis la surface de Vénus (impact le 20 octobre 1975) et la sonde américaine "Viking 1", la première à renvoyer une photo claire du sol martien (impact le 20 juillet 1976). Une série de rovers envoyés sur Mars par la NASA suivirent, pour une exploration mobile de la planète avec le "Mars Pathfinder-Sojourner" (4 juillet 1997), "Opportunity" (25 janvier 2004) et "Curiosity" (6 août 2012). La sonde Cassini et son atterrisseur Huygens, développés en collaboration par la NASA, l'ESA et l'ASI ont quant à eux réalisé un atterrissage en douceur sur une des Lune de saturne : Titan (14 janvier 2005).

Certaines sondes ont aussi intercepté des corps traversant le système solaire comme la sonde Stardust de la NASA dont la mission fut de récolter les poussières cosmiques de la comète "Wild 2" et revenir sur terre (7 février 1999 - 15 janvier 2006) ou comme la sonde Hayabusa de l'agence d'exploration aérospatiale japonaise (JAXA), dont la mission était le retour sur terre d'un échantillon de l'astéroïde Itokawa. La ré-entrée atmosphérique est alors une étape inévitable, pour ce type de mission.



Figure 1: Représentation artistique d'Hayabusa exerçant sa mission d'extraction sur l'astéroïde Itokawa.

1.2 Ré-entrée atmosphérique

L'entrée atmosphérique directe est une des étapes les plus délicates d'une mission spatiale. Voyageant à très haute vitesse, l'objet entrant, qui n'était soumis à aucune force de friction, se heurte à l'atmosphère de la planète. En fonction de la composition du gaz rencontré, ceci provoque d'importants effets thermiques et mécaniques. Pour la terre, la ré-entrée a lieu entre 50 et 80 km d'altitude dans la couche atmosphérique nommée mésosphère se situant juste au dessus de la stratosphère.

Pour un objet spatial artificiel, l'atmosphère est utilisée pour freiner l'objet entrant, mais cela doit être totalement maîtrisé au risque de voir l'objet se décomposer, exploser ou provoquer des dégâts au sol. Les capsules ont en général une forme sphérique-conique pour des questions d'aérodynamique et sont équipées d'un bouclier thermique (Thermal protection system TPS) afin de limiter l'impact du fort échauffement. Le développement d'un matériau TPS adapté et optimisé est donc un sujet de recherche de premier plan pour les agences spatiales.

Les capsules subissent des conditions plus extrêmes comparées aux navettes spatiales et sont protégées par des matériaux ablatifs non réutilisables. Parmi les plus utilisés, on peut citer [30]: "Avcoat5026-39G", matériau ayant équipé les modules Apollo; SLA561V (Super Lightweight Ablator) historiquement utilisé sur tous les objets ayant atteint le sol martien comme Viking, Pathfinder ou le rover Opportunity; PICA154 (Phenolic Impregnated Carbon Ablator) utilisé sur la capsule Stardust et sur celle du tout dernier rover Martien Curiosity. Ce matériau a été sélectionné avec Avcoat pour la future capsule Orion dont l'objectif serait de transporter des hommes sur Mars.

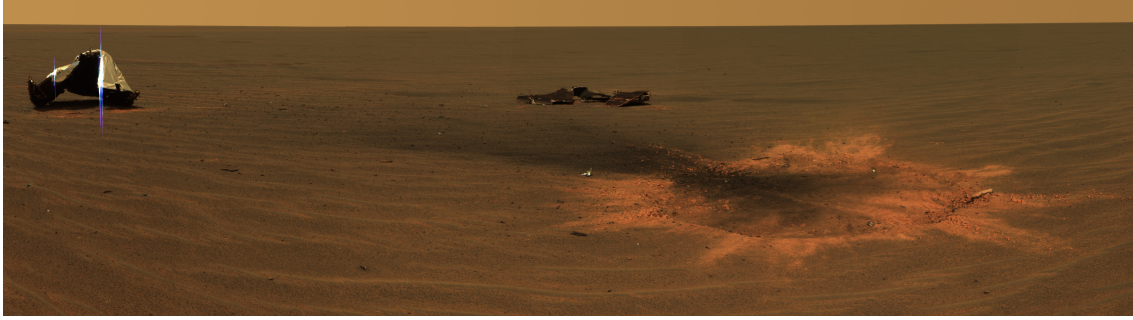


Figure 2: Le bouclier thermique du rover Opportunity couché sur le dos à gauche sur la surface de Mars. Par NASA / JPL / Cornell

1.3 Plan

Dans le cadre de la compréhension des phénomènes de couplage radiatif avec des phénomènes d'ablation pour la rentrée hypersonique des capsules de type Stardust ou Hayabusa, des essais en tube d'expansion ont été réalisés sur une petite capsule protégée conçue avec matériau TPS en résine phénolique, soumise à des conditions de ré-entrée terrestre. Ce projet a pour but de reconstruire numériquement cette expérience avec les codes Eilmer3 et CFD++.

Afin de vérifier la modélisation choisie, les résultats obtenus seront étudiés et comparés aux expérimentaux. Pour arriver à cet objectif, le corps du rapport est divisé en cinq parties:

- **Travaux connexes:** Cette partie cite certaines recherches déjà effectuées dans le domaine et explique le contexte de l'étude. Plus d'explications sur le problème exposé sont données, avec notamment une présentation du tube expérimental.
- **Modélisation:** Cette partie reprend le bagage théorique nécessaire pour recréer le phénomène numériquement. Cela inclut la physique du problème, les équations régissant l'écoulement, une description des modèles, des méthodes numériques, ainsi que les codes utilisés.
- **Simulation:** Cette partie, retrace la méthodologie suivie lors de la réalisation d'une simulation. Elle reprend toute la configuration adoptée avec la géométrie du domaine et son maillage, les conditions initiales et aux limites, la définition du gaz et les réactions chimiques, ainsi que les paramètres choisis pour les transferts par rayonnement et diffusion. Convergence itérative, spatiale et effets de maillage y sont aussi étudiés.
- **Résultats:** Cette partie regroupe les résultats bruts issus de la simulation. Une représentation graphique du phénomène de ré-entrée, pour diverses variables, dans tout le domaine d'étude est présentée.
- **Discussion:** Cette partie concerne les observations et interprétations liées aux résultats. Le comportement du fluide y est étudiée et une comparaison est faite avec les résultats expérimentaux.

2 Travaux connexes

L'ablation et la pyrolyse font l'objet de nombreuses recherches dans l'optimisation de nouveaux matériaux TPS en condition d'entrée atmosphérique. La conception et l'utilisation de matériaux très poreux, moins denses, permettrait une réduction de la charge structurale et de carburant pour une augmentation de la capacité de charge utile. Les matériaux composites poreux modernes tels que le carbone-phénolique et les céramiques de très hautes températures sont à l'étude [3]. Les mécanismes de transfert de chaleur par rayonnement à l'intérieur du matériau hétérogène poreux, ou le blocage au rayonnement des espèces ablatives constituent aussi d'autres pistes de recherche [3].

Une comparaison systématique entre numérique et expérimental est nécessaire afin de mettre au point et valider ces nouveaux modèles. Nombreuses sont les simulations menées du champ d'écoulement hypersonique hors équilibre avec rayonnement et ablation, notamment pour des capsules ayant vraiment voyagé telles que Apollo 4 [32], Stardust [12] ou Hayabusa [39] ou la capsule d'essai Fire II [38] de la NASA.

Dans cette optique, une étude expérimentale a été effectuée dans le tunnel d'expansion X2 de l'Université de Queensland (en Australie) où l'objectif était d'étudier les effets de couplage du champs de rayonnement avec l'ablation sur le plasma hors équilibre thermique entourant un petit modèle de capsule en carbone et résine phénolique, dans des conditions de ré-entrée terrestre [4].

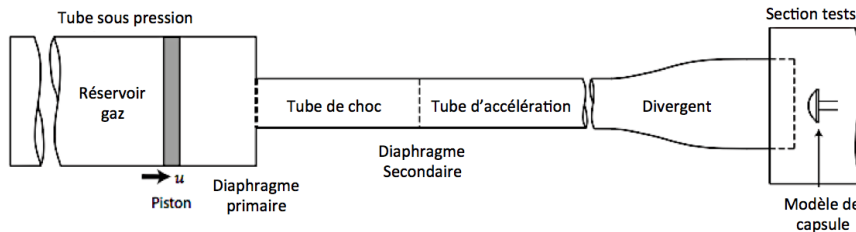


Figure 3: Schéma du tube à expansion X2 de l'université de Queensland (sans être à l'échelle) issu de [4]

Le tube à expansion X2 est un dispositif d'essai au sol permettant de recréer des conditions de ré-entrée atmosphérique. Un réservoir de gaz est porté à haute pression avec un piston. Sous l'effet de la compression un premier diaphragme en fer va exploser, puis un second plus fin en aluminium expulsant du gaz à très haute vitesse vers la capsule depuis le tube d'accélération. Le gaz représentant l'atmosphère que la capsule rencontrera est placé dans le tube de choc. Les conditions de ré-entrée souhaitées sont obtenues en choisissant d'autres gaz aux propriétés différentes dans le réservoir et le tube d'accélération. Le divergent permet d'augmenter l'enthalpie incidente.

Grâce à ce dispositif, le phénomène de ré-entrée atmosphérique a été recréé et le spectre d'émission du gaz de la couche de choc a été relevé pour le modèle en carbone-résine (modèle équipé de TPS), mais aussi pour un modèle de capsule non protégé, fait simplement d'acier. Les données spectrales ont été obtenues à l'aide d'un spectromètre pour des longueurs d'ondes se situant entre les UV et les infrarouges, en vue de retrouver les lignes d'espèces issues de l'ablation ou de pyrolyse. Une comparaison de ces données avec la simulation numérique puis une analyse, est effectuée lors de ce projet.

3 Modélisation

3.1 Problème physique

3.1.1 Régime hypersonique

Un écoulement peut être classifié en différents régimes par son nombre de Mach:

$$M_\infty = \frac{U_\infty}{c_\infty} \quad (1)$$

où U_∞ et c_∞ représentent la vitesse de l'écoulement et la célérité du son en infini amont.

Lorsque $M_\infty > 1$, la vitesse de l'écoulement dépasse celle du son et le régime devient supersonique. Pour $M_\infty > 5$ il est hypersonique.

Dans le cas d'une ré-entrée atmosphérique le nombre de Mach est nettement supérieur à 1 ($M_\infty > 25$) pour une vitesse atteignant 30000km/h et plus. Le régime est donc hautement hypersonique, il est dit de type ré-entrée. L'écoulement hypersonique diffère de l'écoulement supersonique car il implique de très hautes températures.

3.1.2 Couche de choc

Lorsqu'un objet entre en contact avec l'atmosphère d'une planète, une couche de choc apparaît avec une certaine épaisseur (shock-standoff distance). Elle peut être approximée pour un écoulement hypersonique, par l'équation de de Ambrosio et Wortman [5]:

$$\frac{\Delta}{R} = 0.143 \exp\left(\frac{3.24}{M_\infty^2}\right) \quad (2)$$

où R est le rayon de nez du véhicule et où Δ représente la distance du choc en partant de la paroi. Cette distance varie en fonction de l'état thermo-chimique régnant dans la couche de choc.

3.1.3 Chimie et transferts d'énergie

Les espèces du gaz possèdent plusieurs degrés de liberté (modes) sur lesquels l'énergie peut se répartir. Lorsque le flux de particules en infini amont rencontre le gaz de la couche de choc plus dense, la plupart de son énergie cinétique est convertie en énergie de translation à cause de l'impact. Les collisions entre particules dans la couche de choc viennent ensuite répartir l'énergie en un mode de rotation, de vibration et un mode électron-électronique. Contrairement à un gaz en équilibre thermique cette répartition d'énergie entre les modes est déséquilibrée dans le cas d'une ré-entrée et une seule température ne permet plus de décrire le gaz. Celui-ci est donc dans un état de non-équilibre thermique avec des modes à des températures différentes. Des transferts d'énergie efficaces ont lieu entre le mode de rotation et de translation et s'équilibrent rapidement, alors que l'énergie de vibration et électronique mettent plus de temps. En général les modes sont donc regroupés en un mode translation-rotation (tr) et un mode vibration-électronique (ve).

Toute cette énergie provoque une forte augmentation de la température du fait de la forte vibration des espèces du gaz qui finissent par se dissocier. Atomes, ions et électrons se détachent et viennent percuter les autres molécules provoquant des réactions chimiques en cascade. Le gaz devient donc ionisé, il s'agit d'un plasma.

La chaleur vient impacter la surface de la capsule par convection, qui est constituée d'une composante de conduction lié aux gradients de température, et d'une composante de diffusion lié aux gradients de concentration dans le plasma. Cependant lors d'une ré-entrée, les changements d'états énergétiques moléculaires impliquent aussi un flux de chaleur par rayonnement électromagnétique important.

3.1.4 Ablation et Pyrolyse

Les forts transferts de chaleur arrivant sur la capsule provoquent une pyrolyse (brûlure) de la surface intérieure du matériau protecteur (TPS) empêchant justement la conduction de cette chaleur vers le véhicule. Du fait de la porosité du matériau, le gaz issu de la pyrolyse est injecté dans l'écoulement en fonction des gradients de température ou de pression environnants. Une couche carbonisée se forme sur la surface extérieure puis se décompose, provoquant des réactions d'ablation (décomposition, fonte), formant une couche d'ablation visqueuse en proche paroi.

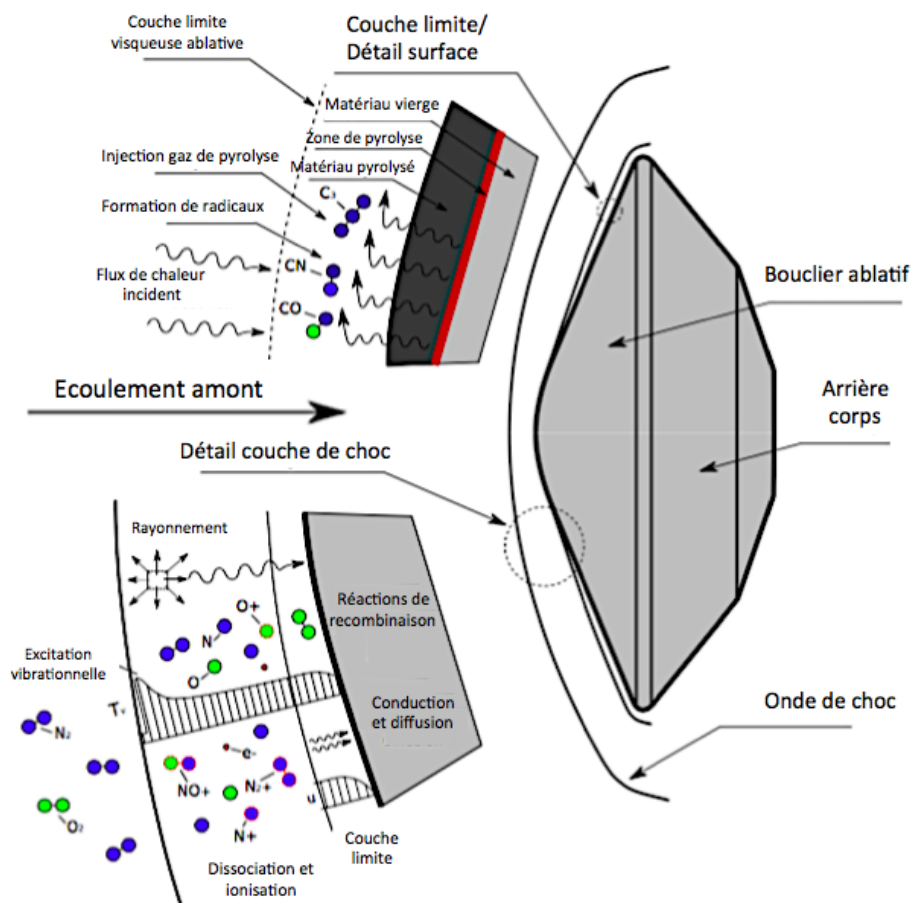


Figure 4: Schéma des processus aéro-thermiques se produisant dans la couche de choc et sur la surface d'une capsule en ré-entrée hypersonique [35]

3.2 Équations gouvernant les écoulements hypersoniques en ré-entrée

3.2.1 Équation de conservation

La relation sous forme locale qui lie la variation spatiale d'une quantité ϕ , par rapport à sa variation temporelle est décrite par l'équation de conservation:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \cdot \vec{u} - \vec{F}) = \Omega \quad (3)$$

où $\phi \cdot \vec{u}$ est le flux de la quantité ϕ , \vec{F} le vecteur densité de flux caractérisant les efforts surfaciques et Ω un terme source de création/destruction de ϕ .

Cette équation traduit le bilan de ϕ . En réalité ϕ n'est pas conservée à cause de la présence du terme source qui produit ou détruit de cette quantité, mais on parle tout de même d'équation de conservation. Cependant si Ω était nul et \vec{F} conservatif, ϕ serait conservée.

3.2.2 Quantités conservées

Dans le cas d'un équilibre thermique et chimique, les équations de Navier-Stokes proviennent de l'injection dans l'équation de conservation, des quantités conservées suivantes: la masse volumique ρ ; la quantité de mouvement totale $\rho \cdot \vec{v}$ et l'énergie totale E .

Lors d'une ré-entrée hypersonique, les effets de non-équilibre du plasma doivent être modélisés. Un non-équilibre chimique et un non-équilibre thermique sont à considérer et respectivement, les termes ρ et E doivent être dissociés:

$$\rho = \sum_s^{N_{\text{speces}}} \rho_s \quad (4)$$

$$E = e + \frac{1}{2} \vec{u} \cdot \vec{u} = \sum_s^{N_{\text{modes}}} e_m + \frac{1}{2} \vec{u} \cdot \vec{u} \quad (5)$$

La relation (4) introduit un mélange miscible d'espèces. La relation (5) traduit le fait que l'énergie totale par unité de masse peut se répartir en une contribution d'énergie interne e , elle même décomposée en un nombre de modes spécifique à la modélisation choisie et d'une contribution cinétique.

3.2.3 Équations de Navier-Stokes compressible

3.2.3.1 Modèle à un mode de température

Conservation de la masse

Pour un non-équilibre chimique et un équilibre thermique, la formulation reste à un seul mode $N_{\text{modes}} = 1$. En substituant (4) dans (3), avec \vec{F} nul, $\Omega = \omega_s$ (qui est le terme source dû aux réactions chimiques) et en ré-exprimant les dérivées associées au mouvement du fluide en dérivées associées au mouvement de l'espèce s , on a:

$$\frac{\partial \rho_s}{\partial t} + \nabla \cdot (\rho_s \vec{u} + \vec{J}_s) = \dot{\omega}_s \quad (6)$$

Avec \vec{u} le vecteur vitesse du fluide, $\vec{J}_s = \rho_s(\vec{u} - \vec{u}_s)$ le vecteur flux massique de diffusion de l'espèce s, $\vec{u} - \vec{u}_s$ étant la vitesse de diffusion de l'espèce s et $\dot{\omega}_s$ le terme source de production de masse de l'espèce s.

Conservation de la quantité de mouvement

Le bilan de la quantité de mouvement traduit le principe fondamental de la dynamique. Des efforts sont à considérer tel que la pression ou les contraintes et \vec{F} aura une expression. Il n'y a en revanche pas de terme source dans ce cas. En substituant $\rho \vec{u}$ dans (3) on a:

$$\frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot (\rho \vec{u} \vec{u}) = -\nabla p + \nabla \tau \quad (7)$$

où p est la pression du gaz, τ est le tenseur de contraintes visqueuses. D'après l'approximation de Chapman Enskog, τ peut être de la forme:

$$\tau = \mu(\nabla \vec{u} + \nabla \vec{u}^t)(\kappa + \mu_1)(\nabla \vec{u} \cdot \hat{I}) \quad (8)$$

μ étant le coefficient de viscosité, $\mu_1 = -2\mu/3$ le coefficient de viscosité secondaire et κ la contribution due à la viscosité apparente, souvent négligée.

Conservation de l'énergie

Le bilan de l'énergie traduit le premier principe de la thermodynamique pour un système ouvert. Si $N_{modes} = 1$:

$$E = e + \frac{1}{2} \vec{u} \cdot \vec{u} \quad (9)$$

On peut donc faire apparaître E dans l'équation de conservation, en faisant $\vec{u} \cdot (eq.6)$. L'ajout d'un terme source d'énergie tenant compte des transferts de chaleur est nécessaire dans le bilan. On a:

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot (\vec{u}(\rho E + p)) = \nabla \cdot (\tau \cdot \vec{u}) - \nabla \vec{q} \quad (10)$$

où \vec{q} tient compte des transferts par: conduction; diffusion (le plasma est un mélange, il s'agit ici de l'inter-diffusion d'enthalpie entre les espèces [10] [34]) et rayonnement:

$$\vec{q} = \vec{q}_{cond.} + \vec{q}_{diff.} + \vec{q}_{rad.} = -K \nabla T + \sum_s^{N_{especes}} h_s \vec{J}_s + \int_v \vec{I}_v dv \quad (11)$$

et où K est le coefficient de conductivité totale, $h_s = e_s + p_s/\rho_s$ l'enthalpie spécifique de l'espèce s et \vec{I}_v le vecteur de luminance énergétique spectrale.

3.2.3.2 Modèle à deux modes de température

La formulation à deux modes ($N_{modes} = 2$) découple la température en un mode translation-rotation et en un mode vibration-électron-électronique. Les équations de conservation restent valables, mais l'expression de l'énergie totale E et du flux conductif \vec{q}_{cond} changent:

$$E = e_{tr} + e_{ve} + \frac{1}{2} \vec{u} \cdot \vec{u} \quad (12)$$

$$\vec{q}_{cond.} = \vec{q}_{cond.-tr} + \vec{q}_{cond.-ve} = -K_{tr} \nabla T_{tr} - K_{ve} \nabla T_{ve} \quad (13)$$

Conservation de l'énergie vibration-électron-électronique

La formulation de l'équation de conservation de l'énergie vibration-électron-électronique donne:

$$\frac{\partial \rho e_{ve}}{\partial t} + \nabla \cdot (\vec{u}(\rho e_{ve} + p_{ve})) = -\nabla \cdot \vec{q}_{ve} + \dot{\Omega}_{VT} + \dot{\Omega}_{ET} + \dot{\Omega}_{EC} + \dot{\Omega}_{VC} \quad (14)$$

Avec \vec{q}_{ve} le vecteur flux de chaleur du mode v-e-e:

$$\vec{q}_{ve} = \vec{q}_{cond.-ve} + \vec{q}_{diff.-ve} + \vec{q}_{rad.} = -K_{ve} \nabla T_{ve} + \sum_s^{N_{esp.}} h_{ve-s} \vec{J}_s + \int_v \vec{I}_v dv \quad (15)$$

$\dot{\Omega}_{VT}$, $\dot{\Omega}_{ET}$, $\dot{\Omega}_{EC}$, $\dot{\Omega}_{VC}$ sont des termes source d'échange d'énergie thermique. Leur apparition vient du fait que le problème modélise désormais un non-équilibre thermique. Respectivement les deux premiers listés doivent représenter les échanges vibration-translation et électronique-vibration, tandis que les deux derniers doivent modéliser le couplage d'énergie chimique.

Les équations des Navier-Stokes en compressible gouvernant les écoulements de type ré-entrée sont donc établies. Une modélisation appropriée est nécessaire pour:

- les propriétés thermodynamiques du gaz dans son ensemble (tels que les conductivités thermiques K_{tr}, K_{ve} , le coefficient de viscosité μ , le flux massique de diffusion \vec{J}_s);
- les taux d'évolution de réaction du terme $\dot{\omega}_s$, déterminant la cinétique chimique;
- les termes sources d'échange d'énergie thermique et les termes source de flux de chaleur (radiatif inclus).

3.3 Modélisation de l'état thermodynamique du gaz

3.3.1 Propriétés thermodynamiques et coefficients

La section suivante explique brièvement comment les propriétés thermodynamiques globales telles que la pression, la température, les chaleurs spécifiques dérivent de la physique statistique et quantique [42].

Résoudre la fonction d'onde pour toutes les particules du plasma, par l'équation de Schrödinger, serait théoriquement la façon la plus complète de déterminer son énergie. Mais à cause du nombre énorme de particules présentes, cette approche ne peut être mise en pratique. Une modélisation possible pour le modèle à deux modes de température serait de considérer que chaque mode thermique possède une collection d'états quantiques en équilibre thermique, disposés selon la distribution de Boltzmann avec une température associée.

Ainsi, la physique statistique peut être mise en pratique (dans l'hypothèse d'une pression modérée) et permet d'obtenir une relation thermodynamique pour: l'énergie libre de Helmholtz F ; l'entropie S ; l'énergie E ; la pression p et le potentiel chimique par molécule $\tilde{\mu}$. Ces relations font intervenir: la fonction de partition Q groupant les états quantiques individuels; la population N , N étant le nombre de ces états quantiques et la température T associée au cas de figure.

Une fonction de partition est exprimée comme une somme qui s'effectue sur tous les micro-état quantiques, mais elle peut aussi être exprimée en fonction d'énergie de niveau (i.e un groupe de micro-états ayant une même énergie) et la dégénérescence lui étant associée. La chimie quantique et des données expérimentales issues de la littérature [19] permettent d'obtenir ces derniers et sont utilisés comme paramètres à la fonction de partition. Pour le mode translation (trans): une énergie de translation d'état quantique simple est donnée comme paramètre. Pour les modes rotation (rot) électronique (el) vibration (vib): dégénérescences, énergies de niveau de rotation, électronique et de vibration sont donnés. On trouvera donc une fonction spécifique selon le mode. En injectant les fonctions de partition dans les relations de thermodynamique statistique précédentes, les variables thermodynamiques sont obtenues pour chaque mode. L'expression de la chaleur spécifique à volume constant et à pression constante dérive de ces résultats.

Les propriétés thermodynamiques globales peuvent être retrouvées en réalisant la combinaison appropriée de tous ces derniers éléments issus de la mécanique statistique et quantique (l'indice "m" concerne les molécules, "s" toutes les espèces, "e" l'électron libre):

$$e_{tr} = \sum_s^{N_{esp.}} f_s e_{trans,s} + \sum_m^{N_{mol.}} f_m e_{rot,m} \quad (16)$$

$$e_{ve} = \sum_m^{N_{mol.}} f_m e_{vib,m} + \sum_s^{N_{esp.}} f_s e_{el,s} + f_e e_{trans,e} \quad (17)$$

où e_{tr} et e_{ve} sont les énergies internes modales globales; f la fraction massique; e_{trans} , e_{rot} , e_{el} , e_{vib} les énergies internes pour chaque modes.

$$c_{v,tr} = \sum_s^{N_{esp.}} f_s c_{v,trans,s} + \sum_m^{N_{mol.}} f_m c_{v,rot,m} \quad (18)$$

$$c_{v,ve} = \sum_m^{N_{mol.}} f_m c_{v,vib,m} + \sum_s^{N_{mol.}} f_s c_{v,el,s} + f_e c_{v,el,e} \quad (19)$$

$$c_{p,tr} = \sum_s^{N_{esp.}} f_s c_{p,trans,s} + \sum_m^{N_{mol.}} f_m c_{p,rot,m} \quad (20)$$

$$c_{p,ve} = \sum_m^{N_{mol.}} f_m c_{p,vib,m} + \sum_s^{N_{mol.}} f_s c_{p,el,s} + f_e c_{p,el,e} \quad (21)$$

où $c_{v/p,tr}$ et $c_{v/p,ve}$ sont les chaleurs spécifiques modales globales à volume/pression constant(e); $c_{v/p,trans}$, $c_{v/p,rot}$, $c_{v/p,el}$, $c_{v/p,vib}$ les chaleurs spécifiques à volume/pression constant(e) pour chaque modes.

L'expression de la température translationnelle globale est:

$$T_{tr} = \frac{e_{tr}}{c_{v,tr}} \quad (22)$$

La température a une influence sur la valeur des chaleurs spécifiques de vibration, électronique et d'électron libre. Une méthode itérative (Newton-Raphson) a été implémentée afin d'obtenir T_{ve} en utilisant son énergie:

$$T_{ve}^{n+1} = T_{ve}^n - \frac{f_{ve}(T_{ve}^n)}{f'_{ve}(T_{ve}^n)} \quad (23)$$

où $f_{ve}(T_{ve}) = e_{ve}(T_{ve}) - e_{ve}^*$ avec e_{ve}^* énergie modale donnée, et $f'_{ve}(T_{ve}) = c_{v,ve}(T_{ve})$

La pression est donnée par l'équation d'état, mais la température pour les électrons libres n'est pas la même que celle des particules plus lourdes. On a donc:

$$p = \sum_{s \neq e-}^{N_{esp.}} p_s + p_e = \sum_{s=e-}^{N_{esp.}} \rho_s R_s T_{tr} + \rho_e R_e T_{ve} \quad (24)$$

où R est la constante universelle des gaz parfaits. Dans le traitement proposé chaque espèce se comporte comme un gaz parfait, est contribuent à la pression totale p .

Le coefficient isentropique et la vitesse du son sont par ailleurs:

$$\gamma = \frac{c_{p,tr} + c_{p,ve}}{c_{v,tr} + c_{v,ve}} \quad (25)$$

$$c^2 = \gamma \left(\frac{p}{\rho} \right) + (\gamma - 1) \left(\frac{T_{tr}}{T_{ve}} - 1 \right) \frac{p_e}{\rho} \quad (26)$$

L'équation (26) a été formulée par Cinnella et Grossman [9] afin de prendre en compte une formulation à deux modes de température. Si $T_{tr} = T_{ve}$ on retrouve la formule classique $c^2 = \gamma(p/\rho)$.

Ces propriétés thermodynamiques globales pour le mode translation-rotation et vibration-électronique-électron libre sont nécessaires aux équations de Navier-Stokes compressible. Cependant le découplage des modes thermiques mais aussi certaines approximations faites lors de la construction de la fonction de partition affectent la précision des propriétés thermodynamiques moléculaires calculées à l'aide de la formulation multi-température.

3.3.2 Coefficients de transport

Les coefficients de transport globaux tels que les conductivités thermiques modales K_{tr} et K_{ve} , le flux massique de diffusion \vec{J}_s pour le calcul des flux \vec{q}_{cond} et \vec{q}_{diff} ainsi que le coefficients de viscosité μ pour le tenseur des contraintes visqueuses τ ont besoin d'être modélisés pour les équations de Navier-Stokes gouvernant le plasma à l'échelle macroscopique. A l'échelle microscopique la dynamique du milieu est dictée par l'équation de Boltzmann. Le changement d'échelle est possible selon la méthode de Chapman-Enskog où la fonction de Boltzmann est approximée par une expansion polynomiale dotée d'un petit paramètre (comparable au nombre de Knudsen). Généralement à l'ordre un de l'expansion, la méthode permet d'établir le lien entre les coefficients de transport globaux et les potentiels d'interaction à l'échelle moléculaire. Ces potentiels sont eux mêmes modélisés à travers les paramètres de collision binaires suivants: la section droite de collision de diffusion $\sigma^2\Omega_{ij}^{(1,1)*}$ et de viscosité $\sigma^2\Omega_{ij}^{(2,2)*}$ [29]. Cependant, le poids numérique des équations ainsi obtenues reste lourd. De ce fait, il existe des modèles de mixture comme ceux établis par Wilke [43], Armaly et Sutton [2] ou Gupta, Yos, Thompson et Lee [18] permettant de reformuler l'expression de $\sigma^2\Omega_{ij}^{(1,1)*}(T)$ et de $\sigma^2\Omega_{ij}^{(2,2)*}(T)$, en de nouvelles fonctions qu'on notera par la suite $\Delta_{ij}^{(1)}(T)$ et $\Delta_{ij}^{(2)}(T)$. Ceci permet, selon le type de transformation (loi de mixture) de trouver un compromis entre temps de calcul et précision.

D'après la méthode Chapman-Enskog, pour une espèce i , une loi de mixture donnée, la viscosité globale est:

$$\mu_i = \frac{X_i m_i}{\sum_j^{Nesp.} X_j \Delta_{ij}^{(2)}(T_{trans,i})} \quad (27)$$

avec T_{trans} la température translationnelle, X_i la fraction molaire et m_i la masse de la particule pour l'espèce i ;

La conductivité de translation est:

$$K_{trans,i} = \frac{15}{4} k_B \left[\frac{X_i}{\sum_j^{Nesp.} \alpha_{ij} X_j \Delta_{ij}^{(2)}(T_{trans,i})} \right] \quad (28)$$

où K_B est la constante de Boltzmann, $\alpha_{ij} = 1 + [(1 - r_m)(0.45 - 2.54r_m)]/(1 + r_m)^2$ avec $r_m = m_i/m_j$ [33];

Les conductivités modales de rotation, vibration, électronique sont:

$$K_{(rot/vib/el),i} = k_B \left[\frac{\frac{(c_{v,rot/vib/el})_i X_i}{R_i}}{\sum_j^{Nesp.} \alpha_{ij} X_j \Delta_{ij}^{(1)}(T_{trans,i})} \right] \quad (29)$$

et enfin le coefficient de diffusion binaire D_{ij} [18], utilisé par la suite lors du calcul du flux chaleur par diffusion \vec{J}_s est:

$$D_{ij} = \frac{kT}{p\Delta_{ij}^{(1)}} \quad (30)$$

Pour le plasma entier on aura:

$$\mu = \sum_i^{Nesp.} \mu_i \quad (31)$$

$$K_{tr} = \sum_{i \neq e-}^{Nesp.} K_{trans,i} + \sum_m^{Nmol.} K_{rot,i} \quad (32)$$

$$K_{ve} = \sum_m^{Nmol.} K_{vib,i} + \sum_i^{Nmol.} K_{el,i} + K_{trans,e} \quad (33)$$

3.4 Processus de non-équilibre

Le processus de non-équilibre est représenté à travers le terme source de production de masse $\dot{\omega}_s$ de l'équation (6), mais aussi à travers les termes source d'échange d'énergie inter-modaux $\dot{\Omega}_{VT}$, $\dot{\Omega}_{ET}$, $\dot{\Omega}_{EC}$, $\dot{\Omega}_{VC}$ de l'équation (14).

3.4.1 Modèle chimique

La formule d'une équation de réaction chimique est:



où α_i et β_i sont les coefficients stœchiométriques des réactifs et produits, X_i est la concentration de l'espèce chimique i . La réaction peut se faire dans les deux sens, avec un taux d'avancement de réaction dans le sens direct k_f ou indirect k_b .

Le terme source de production de masse $\dot{\omega}_s$ (par unité de volume) de l'espèce s est issu d'une contribution de chaque réaction [33]:

$$\dot{\omega}_s = \sum_r^{N_{react.}} \dot{\omega}_{s,r} = \sum_r^{N_{react.}} \left(\frac{d[X_s]}{dt} \right)_r \quad (35)$$

Selon la loi d'action de masse [16] de C.Guldberg et P.Waage, le terme source de production de masse $\dot{\omega}_{s,r}$ de l'espèce s par la réaction r , peut s'écrire comme une différence entre un terme source de production $\dot{\omega}_{s,r}^{pr}$ et de destruction $\dot{\omega}_{s,r}^{des}$:

$$\dot{\omega}_{s,r} = \dot{\omega}_{s,r}^{pr} - \dot{\omega}_{s,r}^{des} = (\beta_s - \alpha_s) \left\{ k_f \prod_s [X_s]^{\alpha_s} - k_b \prod_s [X_s]^{\beta_s} \right\} \quad (36)$$

Le taux d'avancement de réaction dans le sens direct et soumise à la température T , est donné par l'équation d'Arrhenius:

$$k_{f,r}(T) = A_{f,r} T^{N_{f,r}} e^{-B_{f,r}/RT} \quad (37)$$

où $A_{f,r}$ (nommé aussi facteur de fréquence) et $N_{f,r}$ sont des paramètres pre-exponentiels; $B_{f,r}$ est l'énergie d'activation (énergie nécessaire pour que la réaction ait lieu) et R la constante des gaz parfaits. La formule (37) peut aussi s'exprimer en fonction de la température d'activation: $T_a = -B_{f,r}/R$. Un terme $(P/P_{atm})^{N_p}$ peut être rajouté en plus à l'équation dans le cas où la réaction dépendrait de la pression, avec N_p un autre facteur exponentiel.

Le taux dans le sens indirect est lié à une condition d'équilibre:

$$k_{b,r}(T) = \frac{k_{f,r}(T)}{K_{c,r}(T)} \quad (38)$$

où pour chaque réaction r : $K_{c,r}$ est la constante d'équilibre. Cette constante dépend de l'enthalpie libre G de la réaction concernée (également énergie de Gibbs). Lorsque G reste négative (pour une pression et température fixe), la réaction évolue. En pratique $K_{c,r}$ est définie dans le code comme une somme de coefficients que l'on peut trouver dans la littérature [31].

Les taux de réaction précédents (cas d'équilibre) peuvent être modifiés afin de capturer les phénomènes de non-équilibre thermochimiques du fluide hypersonique. Un facteur de correction est alors introduit afin de transformer $k_f = K_{eq}$ en K_{neq} qui dépend de chaque contribution modale:

$$k_{neq}(T_{trans}, T_{vib}, T_{rot}, T_{el}, T_e) = Z(T_{trans}, T_{vib}, T_{rot}, T_{el}, T_e) k_{eq}(T_{trans}) \quad (39)$$

Il existe plusieurs type de réactions. Quand une molécule interagit avec un atome, la réaction peut être soit de type "dissociation de particules lourdes par impact" où la molécule est dissociée en atomes, soit "d'échange neutre" si les atomes se recombinent en une nouvelle molécule sans charges.

Pour une réaction faisant intervenir des ions, elle peut être soit de type "ionisation par

impact d'électron" où un électron percute un atome en lui extrayant un autre électron supplémentaire pour former un ion, soit de type "ionisation associative", cas où deux atomes interagissent pour former une molécule ionisée et un électron libre. Pour les réactions de dissociation, la vibration de la molécule dissociée et la translation des espèces entrant en collision sont les principaux acteurs d'énergie. Le facteur de correction concerné est donc exprimé juste en fonction de T_{trans} et T_{vib} .

$$k_{f,neq}(T_{trans}, T_{vib}) = Z(T_{trans}, T_{vib})k_f(T_{trans}) \quad (40)$$

De nombreux modèles de dissociation de particules lourdes par impact (pour une formulation à deux modes de température $T_{tr} - T_{ve}$) on été énoncés [11] et comparés afin de déterminer un facteur de correction approprié en fonction des conditions d'écoulement. Il y a celui de Park, Hansen, Marrone–Treanor, Hammerling, Knab, Macheret–Fridman... Mais le modèle utilisé pour ce projet est celui de park.

Modèle de Park

Le modèle de Park modélise le non équilibre des réactions de dissociation en redéfinissant la température T de l'équation (37):

$$T = \sqrt{T_{trans}T_{vib}} \quad (41)$$

généralisée par la suite comme:

$$T = T_{trans}^{s-1} T_{vib}^s \quad (42)$$

Un paramètre $s = 0.3$ semble mieux adapté au températures supérieures à 8000K [11] que $s=0.5$ (équivalent à l'équation (41)).

Le facteur de correction $Z(T_{trans}, T_{vib})$ est déterminé en injectant l'équation de Park (42) dans l'équation du taux d'avancement dans la formulation d'Arrhenius (37) et en égalisant avec l'équation du taux de non-équilibre (40). D'un point de vue pratique le passage par le facteur de correction n'est pas toujours appliqué. Les taux d'avancement dans le sens direct et indirect sont calculés directement en remplaçant T dans (37) par les données du tableau ci-dessous (ce qu'applique le code CFD++, où chaque exposant de T_{trans} et T_{vib} est défini par l'utilisateur):

Type de réaction	Température T_f , taux sens direct	Température T_b , taux sens indirect
Dissociation de particules lourdes par impact	$T_{vib}^s T_{trans}^{s-1}$	T_{trans}
Dissociation par impact d'électron	$T_{vib}^s T_e^{s-1}$	T_e
Echange neutre	T_{trans}	T_{trans}
Ionisation associative	T_{trans}	T_{trans}
Ionisation par impact d'électron	T_e	T_e

Table 1: Expression de la température pour la formule d'Arrehenius, selon Park

3.4.2 Modèle thermique

Les interactions entre les modes thermiques des espèces entrant en collision et les réactions chimiques provoquent des échanges d'énergie entre modes. Comme le modèle à 2 modes de températures utilise la mécanique statistique, les échanges entre les modes thermiques ne sont pas implicites aux équations et doivent être modélisés à travers les termes sources présents dans l'équation (14).

Echange vibration-translation

Le terme source Ω_{VT} est issu de la formule de Landau-Teller, qui exprime le taux de variation de l'énergie vibratoire pour une espèce moléculaire p due à un échange de translation avec une espèce c entrant en collision.

$$\Omega_{VT} = \sum_p^{Nmol.} \rho_p \frac{e_{vib,p}^* - e_{vib,p}}{(\tau_{vib}^p)_{VT}} \quad (43)$$

où $e_{vib,p}$ est l'énergie vibratoire (enthalpie) par unité de masse de l'espèce s (le terme d'énergie possédant un "*" indique une évaluation à la température T_e), $(\tau_{vib}^p)_{VT}$ est le temps de relaxation d'énergie translation-vibration pour l'espèce s qui peut être calculé selon la théorie de Millikan et White [25].

Echange translation-électronique

Le terme source Ω_{TE} d'Appleton et Bray [1] modélise en réalité un échange d'énergie translation-translation entre électrons et particules lourdes subissant des collisions élastiques:

$$\Omega_{TE} = 3\rho_e R(T - T_e) \sum_{s \neq e^-}^{Nmol.} \frac{\nu_{es}}{M_s} \quad (44)$$

où M_s est la masse moléculaire de l'espèce s , ρ_e et T_e respectivement la masse volumique et la température des électrons. ν_{es} est la fréquence de collision efficace (électrons-particules lourdes) [40].

Couplage Chimique-X

Ce terme source comprends Ω_{CV} et Ω_{CE} ($X = \text{vibration}$ ou $X = \text{électronique}$) et permet de modéliser les gains ou pertes d'énergie de vibration ou électronique, liés à la production ou à la destruction d'espèces chimiques:

$$\Omega_{CX} = \sum_i^{Nesp.} \sum_r^{Nreact.} \left[-G_{ir}^{des} (\dot{\omega}_i)_r^{des} + G_{ir}^{pr} (\dot{\omega}_i)_r^{pr} \right] \quad (45)$$

où G_{ir}^{des} et G_{ir}^{pr} sont les changements modaux d'énergie par unité de masse, respectivement associés à la consommation et à la production de l'espèce i due à la réaction j , survenant au taux $(\dot{\omega}_i)_r^{des}$ et $(\dot{\omega}_i)_r^{pr}$. La modélisation la plus simple (dite "non-préférentielle") est d'assumer que cette production ou destruction se fait avec une énergie égale à celle de vibration pour Ω_{CV} , ou électronique pour Ω_{CE} : $G_{ir}^{des} = G_{ir}^{pr} = e_{(vib/el),i}(T_{vib/el})$.

D'autres modélisations permettent d'obtenir G_{ir}^{des} et G_{ir}^{pr} comme celui de Park [31], Knab et al. [13], Treanor et Marrone [41], ou encore Macheret-Fridman [23].

3.5 Modélisation du rayonnement

3.5.1 Rayonnement thermique

Lorsqu'une molécule ou un atome reçoit suffisamment d'énergie, les électrons changent de niveau d'énergie et passent à un niveau excité instable. Ils retrouvent l'instant d'après un niveau plus bas, en émettant un photon, ce qui provoque le rayonnement. Les électrons d'un atome ne peuvent exister que pour certains niveaux d'énergie discrets. Ces paliers sont représentatifs de la couche électronique sur laquelle les électrons se situent.

Le transport d'énergie par rayonnement ne nécessite pas de milieu matériel et peut se faire dans le vide: chaque photon est une onde électromagnétique se propageant à la vitesse de la lumière et est caractérisé par une fréquence ν (Hz), correspondant à une transition d'énergie bien précise. La fréquence peut aussi s'exprimer: $\nu = c/\lambda$, avec λ (nm) la longueur d'onde, c la vitesse du son. L'énergie de transition associée est $\epsilon = h\nu$ avec la constante de Planck $h = 6.62 \times 10^{-34}$ (J.s).

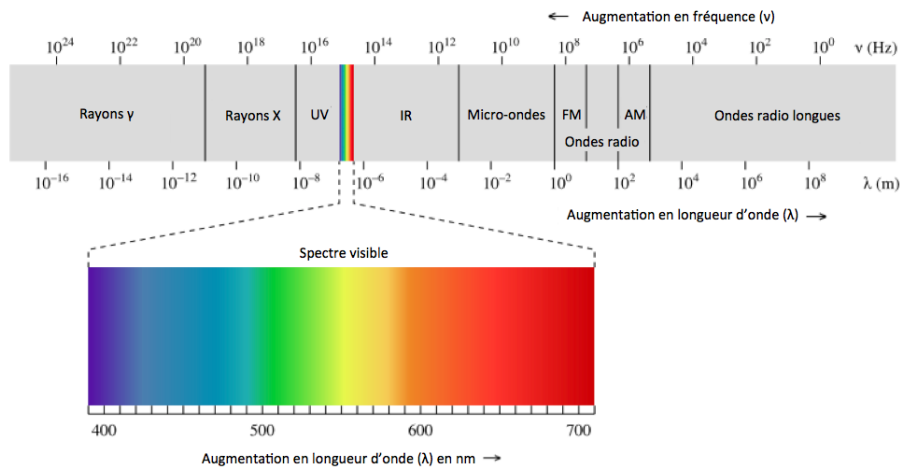


Figure 5: Spectre électromagnétique

Toute matière émet un rayonnement et ainsi perd de l'énergie. Plus la température est élevée, plus le corps perd l'énergie. Un équilibre est atteint quand l'énergie reçue est égale à l'énergie perdue par rayonnement.

Une espèce émet un spectre de fréquences spécifiques des changements d'états d'énergie qu'elle subit. Le rayonnement thermique englobe les UV, le visible et l'infrarouge.

3.5.2 Rayonnement du corps noir

Un corps noir est un corps qui absorbe totalement le rayonnement qu'il reçoit (pas de flux réfléchi ni de flux transmis). Le symbole "o" est utilisé le représenter.

Luminance énergétique spectrale

La luminance énergétique spectrale d'un corps noir (aussi nommée luminance monochromatique directionnelle) correspond à une densité de flux ($W.m^{-2}$) par unité de fréquence (Hz^{-1}) par unité d'angle solide Ω (unité de direction en sr^{-1}). Elle s'exprime (dans le système SI en $W.m^{-2}.sr^{-1}.Hz^{-1}$) par la loi de Planck:

$$L_{\Omega,\nu}^{\circ}(\nu, T) = \frac{2h\nu^3}{c^2} \frac{1}{e^{h\nu/(kT)} - 1} \quad (46)$$

Quelques remarques sur la luminance énergétique spectrale:

- Le terme anglais est "spectral radiance", plus anciennement qualifiée de "specific radiative intensity" notée I_{ν} ;
- La loi de Wien permet de déterminer la longueur d'onde pour laquelle la luminance est maximale, i.e le pic d'une fonction sur la figure (6) pour une température donnée. Ceci permet de déterminer la longueur d'onde dominante émise en connaissant la température de la source et vice versa. Elle s'écrit: $\lambda_{max}^{\circ} T = 2898.77$;
- L'exitance énergétique spectrale $M_{\nu}^{\circ}(\nu, T)$ (en $W.m^{-2}.Hz^{-1}$) est une grandeur hémisphérique qui s'obtient en intégrant la luminance énergétique spectrale (qui est une grandeur directionnelle exprimée en stéradian) suivant toutes les directions de l'hémisphère dans lequel rayonne l'élément de surface;
- La loi de Lambert est obtenue en intégrant l'exitance énergétique spectrale sur toutes les fréquences: $M^{\circ}(T) = \pi L^{\circ}(T)$, avec M° l'exitance énergétique totale (total emissive power en anglais) ($W.m^{-2}$), et L° la luminance énergétique totale ($W.m^{-2}.sr^{-1}$);
- La loi de Stefan-Boltzmann est généralement utilisée pour évaluer le flux de rayonnement émis par une source par unité de surface en connaissant la température de cette dernière. Pour un corps noir, vérifiant la loi de Lambert, l'exitance peut se réécrire: $M^{\circ}(T) = \pi L^{\circ}(T) = \sigma T^4$ où $\sigma = 57.67 \times 10^{-8} W.m^{-2}.K^{-4}$ est la constante de Boltzmann;

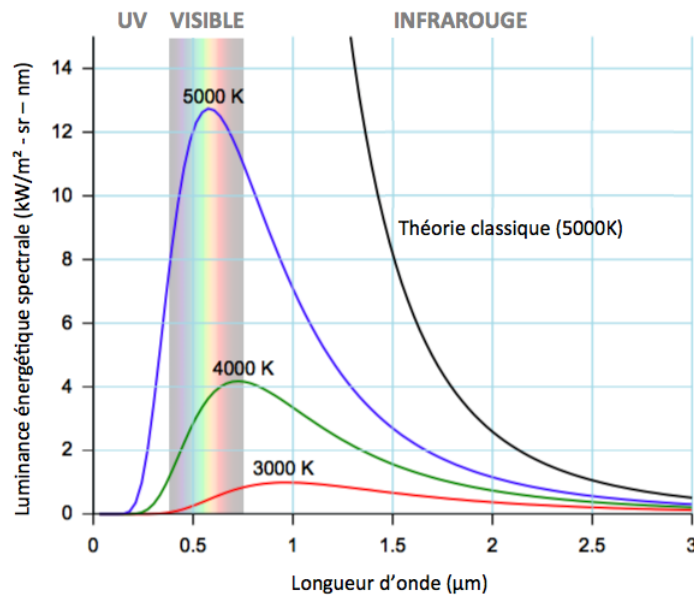


Figure 6: Loi de Planck

3.5.3 Modèle de transport

Les photons se propageant dans une direction de propagation \vec{s} sur une distance "s" (entre une source et une destination), vont subir sur leur chemin l'interaction du gaz environnant.

En traversant le milieu sur le chemin "s" (en considérant ce chemin divisée en volumes élémentaires de longueur dS), certains photons vont être absorbés par d'autres particules, d'autres vont continuer leur chemin vers le prochain volume élémentaire, d'autres vont se diffuser dans d'autres directions. La forme différentielle générale de l'équation de transfert radiatif du milieu permet de décrire cela et s'écrit [26]:

$$\underbrace{\frac{1}{c} \frac{\partial I_\nu}{\partial t}}_{\text{var. temporelle}} + \underbrace{\frac{\partial I_\nu}{\partial s}}_{\text{var. spatiale}} = \underbrace{j_\nu}_{\text{émission}} - \underbrace{\kappa_\nu I_\nu}_{\text{absorption}} - \underbrace{\sigma_{s,\nu} I_\nu + \frac{\sigma_{s,\nu}}{4\pi} \int I_\nu(\hat{s}_i) \phi(\hat{s}_i, \hat{s}) d\Omega_i}_{\text{diffusion}} \quad (47)$$

où I_ν est la luminance énergétique spectrale, j_ν (en $W.m^{-3}.sr^{-1}.Hz^{-1}$) et κ_ν (en $1/m$) sont les coefficients moléculaires spectraux.

La vitesse de la lumière est beaucoup plus rapide que l'évolution en temps du champ d'écoulement, l'équation (47) peut se simplifier de son terme de variation temporelle. Pour les plasmas faiblement ionisés et à pression modérée, le terme de diffusion peut aussi être négligé.

Le terme source de rayonnement dans les équations de Navier-Stokes (15) est la divergence négative du vecteur de flux de chaleur local par rayonnement:

$$-\nabla \cdot \vec{q}_{rad.} = -\nabla \cdot \int_0^\infty \int_0^{4\pi} \vec{I}_\nu \vec{s} d\Omega d\nu \quad (48)$$

qui s'exprime en utilisant l'équation de transport (47) comme une différence entre émission et absorption:

$$-\nabla \cdot \vec{q}_{rad.} = \int_0^\infty \int_0^{4\pi} \kappa_\nu I_\nu d\Omega d\nu - 4\pi \int_0^\infty j_\nu d\nu \quad (49)$$

Le calcul de $\nabla \cdot \vec{q}_{rad}$ et de \vec{q}_{rad} s'effectue par le biais de différents modèles de transport où I_ν est modélisé. Deux modèles seront utilisés pour ce projet (inclus dans Eilmer3):

- La méthode des plans tangents suppose le rayonnement dans la direction de chaque ligne normale à la surface de l'objet. Ceci réduit le problème, au rayonnement d'un plan infiniment fin parallèle à la paroi, permettant une résolution rapide et efficace, mais n'est précise que pour une zone proche de la ligne d'arrêt, pour des capsules au rayon de nez large.
- La méthode Monte-carlo est une méthode statistique et fait partie des méthodes type tracé de rayons. Le domaine est divisé en cellules dans lesquelles des photons sont envoyés en paquets discrets (l'utilisateur fixe leur nombre). Certains seront absorbés, d'autres non, et une loi de probabilité permet la modélisation d' I_ν . Cette méthode permet une résolution sur tout le domaine, mais peut nécessiter des calculs couplés plus longs ou introduire du bruit statistique.

3.5.4 Modèle spectral

Le modèle spectral permet de calculer les coefficients de rayonnements spectraux j_ν et κ_ν qui sont modélisés en fonction du type de transition énergétique dont le rayonnement provient. L'expression des coefficients en fonction de chaque type se trouve dans [36] (pour le modèle spectral PHOTRAURA). La modélisation prend aussi en compte les processus collision-rayonnement. Les 3 grands types de transitions traités sont:

- Les transitions lié-lié: l'électron transite d'une couche électronique instable vers une couche plus stable en émettant un photon.
- Les transitions lié-libre: l'électron arrive du milieu extérieur à l'atome et vient se placer sur une de ses couches électroniques, ce qui provoque l'émission d'un photon.
- Les transitions libre-libre: l'électron arrive du milieu extérieur à l'atome et en ressort, ce qui provoque aussi l'émission d'un photon.

3.6 Méthodes numériques utilisées

3.6.1 Discrétisation spatiale: Méthode de volumes finis

Les codes utilisés lors de ce projet s'appuient sur une discrétisation spatiale de type volume finis. La méthode des volumes finis se base sur une formulation de l'équation de conservation (3) écrite sous forme intégrale:

$$\frac{d}{dt} \int_{\mathcal{D}} \phi dv = \oint_{\partial \mathcal{D}} \vec{F} \cdot \vec{n} ds + \int_{\mathcal{D}} \Omega dv \quad (50)$$

où \vec{n} est le vecteur unitaire normal à la surface $\partial \mathcal{D}$ du volume de contrôle \mathcal{D} .

Si le domaine est divisé en plusieurs petits volumes élémentaires (cellules de maillage), on peut considérer un de ces éléments comme étant le volume de contrôle \mathcal{D} . Si l'élément est un polyèdre, sa surface est donc divisée en plusieurs facettes S_i . Dans ce cas le terme de flux surfacique peut être réécrit (si l'on néglige les variations de \vec{F} sur une facette et si \vec{n} reste constant):

$$\oint_{\partial \mathcal{D}} \vec{F} \cdot \vec{n} ds = \sum_i \oint_{\Gamma_i} \vec{F}_i \cdot \vec{n}_i ds = \sum_i \vec{F}_i \cdot \vec{n}_i \oint_{\Gamma_i} ds = \sum_i \vec{F}_i \cdot \vec{n}_i S_i \quad (51)$$

où \vec{F}_i est une approximation de \vec{F} sur S_i .

La discrétisation spatiale est donc effectuée en calculant le bilan des flux sur chaque maille élémentaire.

Les autres termes ϕ et Ω qui ne sont pas des flux, sont considérés constants et égaux à la valeur au centre de leur volume:

$$\int_{\mathcal{D}} \phi dv = \phi v \quad \text{et} \quad \int_{\mathcal{D}} \Omega dv = \Omega v \quad (52)$$

3.6.2 Discrétisation temporelle

La discrétisation temporelle peut se faire selon deux types de schémas: explicite ou implicite. Les méthodes explicites trouvent une solution de l'état d'un système à une date ultérieure à partir du système à l'heure actuelle, alors que les méthodes implicites trouvent une solution en résolvant une équation impliquant à la fois l'état actuel du système et celui ultérieur. La méthode d'Euler explicite, parfois nommé Forward Euler (FE) et la méthode d'Euler implicite parfois nommé Backward Euler (BE) sont deux approches couramment utilisées.

Afin d'appliquer ces méthodes, le temps est divisé en n pas de temps. Les quantités t_n et y_n sont respectivement le temps et la solution numérique au pas de temps n (i.e $y_n \equiv y(t = t_n)$). La taille de pas de temps Δt (pouvant être constante), est donnée par $\Delta t = t_n - t_{n-1}$.

Schéma temporel explicite

La méthode explicite d'Euler (FE) repose sur un développement en série de Taylor tronquée, où y est approximé au voisinage de $t = t_n$ comme:

$$y(t_n + \Delta t) \equiv y_{n+1} = y(t_n) + \Delta t \frac{dy}{dt} \Big|_{t_n} + O(\Delta t^2) = y_n + \Delta t f(y_n, t_n) + O(\Delta t^2) \quad (53)$$

Si t_n et y_n sont connus au pas de temps n , le schéma d'Euler explicite calcule alors la solution y_{n+1} au pas de temps suivant ($n + 1$) comme:

$$y_{n+1} = y_n + \Delta t f(y_n, t_n) \quad (54)$$

où $f(y_n, t_n)$ est la dérivée discrète de y au temps t_n .

La méthode d'Euler est ici explicite car y_{n+1} est calculé par la simple connaissance des quantités antérieures y_n et $f(y_n, t_n)$. Le fait de priver l'équation (53) de son terme $O(h^2)$ introduit une erreur de troncature locale: $O(h^2)$. En général, si le développement de Taylor est prolongé et si cette erreur est $O(hk + 1)$ alors la méthode est dite d'ordre k . Le schéma précédent est donc de premier ordre. Un ordre supérieur donne lieu à une erreur de troncature locale plus faible pour la même taille de pas.

Les méthodes explicites sont faciles à mettre en œuvre mais ont l'inconvénient d'être stables que pour une taille de pas de temps limitée. Cette condition sur le pas de temps est la condition de Courant–Friedrichs–Lewy (CFL), qui dans le cas 2D donne:

$$C = \frac{u_x \Delta t}{\Delta x} + \frac{u_y \Delta t}{\Delta y} \leq C_{\max} \quad (55)$$

où Δx et Δy sont les pas d'espace dans la direction \vec{x} et \vec{y} , C_{\max} est la contrainte CFL.

Le schéma temporel d'Euler explicite est implémenté dans le code Eilmer3 et le code CFD++.

Schéma temporel implicite

La méthode implicite d'Euler (BE) repose sur un développement en série de Taylor tronquée, où y est approximé au voisinage de $t = t_n + 1$ comme:

$$y_n \equiv y(t_{n+1} - \Delta t) = y(t_{n+1}) - \Delta t \frac{dy}{dt} \Big|_{t_{n+1}} + O(\Delta t^2) \quad (56)$$

La solution au temps $n + 1$ est obtenue par la résolution d'un problème inverse:

$$y_{n+1} = y_n + \Delta t f(y_{n+1}, t_{n+1}) \quad (57)$$

Un schéma temporel implicite permet une résolution plus rapide car le pas de temps est moins contraint et peut être plus grand. Il est aussi plus stable mais peut être compliqué à mettre en œuvre, notamment pour des problèmes non linéaires.

Le schéma d'Euler implicite n'est implémenté que dans le code CFD++ à l'heure actuelle mais est en cours de développement dans le cas d'Eilmer3.

Méthodes de prédiction-correction

La méthode de prédiction-correction, parfois nommée "méthode Euler-trapézoïdale" est une formulation semi-explicite où l'avancement temporel (obtention de y_{n+1}) est d'une part effectué par un schéma explicite (prédicteur) et d'autre part corrigé avec un schéma implicite (correcteur). La combinaison des deux permettrait de meilleures caractéristiques de convergence. Un schéma (FE) et un autre schéma nommé "Adams d'ordre 2" sont souvent utilisés.

La méthode α -QSS de Mott [27] est une méthode de prédiction-correction bien adaptée aux systèmes d'équations chimiques (i.e l'équation (35)) car souvent ces systèmes sont raides [6]. En résumé, une équation raide est une équation différentielle dont les méthodes numériques courantes nécessitent un pas de temps anormalement petit. α -QSS est composé de schémas spéciaux permettant une résolution plus efficace faisant intervenir un paramètre α contrôlant la façon dont la mis-à-jour fonctionne pour l'intégration d'une espèce donnée [15].

Autres schémas temporels

Le schéma de Runge-Kutta d'ordre 4 (disponible aussi dans CFD++) modifie le schéma d'Euler explicite en introduisant des points intermédiaires afin d'avoir une approximation plus précise. Il est très efficace pour les systèmes pas trop raides en utilisant les coefficients de Cash et Karp [21]. L'ordre 4 est très populaire pour de nombreux problèmes avec 3 valeurs intermédiaires introduites.

3.6.3 Consistance, stabilité et convergence

Les schémas numériques obtenus après la discrétisation des dérivées partielles temporelles et spatiales doivent être consistants, stables avec une solution convergente.

Un schéma numérique est consistant si l'erreur de troncature (différence entre l'opérateur de dérivation discret et l'opérateur de dérivation continu) tend vers 0, quand le pas de temps et d'espace tendent vers 0.

Un schéma numérique est stable si le "bruit numérique" (erreurs d'arrondi...) n'est pas amplifié au cours de chaque pas de temps. L'analyse de stabilité d'une équation peut se faire par la méthode de Von-Neumann. Cette méthode permet d'obtenir la condition de stabilité Courant–Friedrichs–Lewy (CFL) des schémas explicites. Pour les schémas implicites, l'analyse montre qu'ils sont inconditionnellement stable, et selon la définition du CFL, il n'y a pas de restriction sur la taille du pas de temps.

Une solution est convergente si l'erreur globale (différence entre la solution calculée et la solution exacte, i.e $G_n = |y^e(t_n) - y_n|$) tends vers 0 quand le pas de temps et d'espace tendent aussi vers 0. Pour un problème bien posé, le théorème de Lax indique que si le schéma est consistant, la stabilité est une condition nécessaire et suffisante pour assurer la convergence.

3.7 Implémentation

3.7.1 Code Eilmer3

Eilmer3 est une collection programmes intégrés pour la simulation d'écoulements compressibles transitoires en deux et trois dimensions spatiales [15], développé par l'Université de Queensland (australie) en partenariat avec le groupe IAG à l'EPFL. Le code se fonde sur une formulation volumes finis des équations de conservation de masse, quantité de mouvement, d'énergie et d'espèces et est implémenté sur des maillages structurés. A partir d'un écoulement initial avec des conditions aux limites, les variables d'état dans chaque cellule volumes finis sont mis à jour dans le temps pour fournir une simulation du champ d'écoulement en évolution (même si le temps continue d'avancer, cette évolution peut atteindre un état considéré stationnaire car il peut ne plus y avoir de variations du choc). Le calcul multiprocesseur (MPI) est par ailleurs possible.

L'implémentation des équations de Navier Stokes par Eilmer3 se fait à partir de l'équation de conservation sous forme intégrale (50). En appliquant la définition de la dérivée convective ($d/dt = \partial/\partial t + \nabla \cdot$) et le théorème de Green-Ostrogradski, cette équation peut aussi s'écrire:

$$\frac{\partial}{\partial t} \int_{\mathcal{D}} U dv = \int_{\partial \mathcal{D}} (\vec{F}_i - \vec{F}_v) \cdot \vec{n} ds + \int_{\mathcal{D}} Q dv \quad (58)$$

où $\vec{F}_i = \phi \cdot \vec{u}$ (il s'agit d'une composante non visqueuse), $\vec{F}_v = F$ (il s'agit d'une composante visqueuse), $U = \phi$ le vecteur des quantités conservées et $Q = \Omega$ le terme source.

Selon la formulation volumes finis (où les interfaces cellulaires sont notés Nord, Est, Sud et Ouest) la forme intégrale est approchée par l'expression:

$$\frac{d}{dt}U = -\frac{1}{V} \sum_{NESO} (\vec{F}_i - \vec{F}_v) \cdot \vec{n} ds + Q \quad (59)$$

où U et Q représentent maintenant des valeurs de cellule moyenne.

Une approche de division d'opérateur [28] permet une application découplée des mécanismes physiques. Par cette méthode, le terme source Q est séparé en plusieurs contributions:

$$Q = Q_{geom.} + Q_{chim.} + Q_{therm.} + Q_{rad} \quad (60)$$

et l'intégration temporelle de (59) est approchée par:

$$\begin{aligned} \int_{\Delta t} \frac{dU}{dt} dt &= \int_{\Delta t} \left(\frac{dU}{dt} \right)_{inv} dt + \int_{\Delta t} \left(\frac{dU}{dt} \right)_{visq} dt \\ &+ \int_{\Delta t} \left(\frac{dU}{dt} \right)_{chim.} dt + \int_{\Delta t} \left(\frac{dU}{dt} \right)_{therm.} dt \end{aligned} \quad (61)$$

où

$$\int_{\Delta t} \left(\frac{dU}{dt} \right)_{inv} dt = -\frac{1}{V} \sum_{NESO} (\vec{F}_i) \cdot \vec{n} ds + Q_{geom.} + Q_{rad} \quad (62)$$

$$\int_{\Delta t} \left(\frac{dU}{dt} \right)_{inv} dt = -\frac{1}{V} \sum_{NESO} (-\vec{F}_v) \cdot \vec{n} ds \quad (63)$$

$$\int_{\Delta t} \left(\frac{dU}{dt} \right)_{chim.} dt = Q_{chim.} \quad (64)$$

$$\int_{\Delta t} \left(\frac{dU}{dt} \right)_{therm.} dt = Q_{therm.} \quad (65)$$

Dans le cas d'une géométrie 2D avec un modèle à 2 modes de température, les équations de Navier-stokes incompressibles sont obtenues avec U :

$$U = [\rho \quad \rho_x \quad \rho_y \quad \rho E \quad \rho_e \quad \rho e_{vm} \quad \rho f_s] \quad (66)$$

L'énergie de translation-rotation ne fait partie du vecteur des quantités conservés car il peut être directement obtenu à partir de l'énergie totale:

$$\rho e_{tr} = \rho E - \frac{1}{2} \rho (u_x^2 + u_y^2) - \rho e_{ve} \quad (67)$$

Les composantes de flux \vec{F}_i et \vec{F}_v sont exprimés avec les contraintes de viscosité axisymétriques. Le terme Q_{chim} tiens compte des termes sources liés à la chimie ($\dot{\omega}$, $\dot{\Omega}_{EC}$ et $\dot{\Omega}_{VC}$), le terme Q_{therm} des échange d'énergie thermique ($\dot{\Omega}_{VT}$, $\dot{\Omega}_{ET}$), le terme Q_{rad} du terme source de rayonnement (∇q_{rad}) et Q_{geom} considère un paramètre géométrique. Les expressions des termes \vec{F}_i , \vec{F}_v et Q sont données en annexe, ou dans le guide théorique de Eilmer3 [15] qui comprends aussi une description des méthodes.

Les termes sources d'échanges thermique, ou chimiques (avec ablation ou pyrolyse, si inclus au problème) sont définis dans Eilmer3 à travers des fichiers dont l'extension est de type ".lua" où l'utilisateur peut définir le modèle souhaité comme par exemple le modèle de Park pour les réactions chimiques.

L'approche division d'opérateur a l'avantage de permettre la sélection d'un schéma d'intégration temporel optimal pour chaque composante selon le cas physique. Les systèmes d'EDP sont intégrés dans le temps grâce à une méthode de prédiction-corrrection pour l'avancement non visqueux, une méthode d'Euler explicite pour l'avancement visqueux, une méthode α -QSS pour l'avancement chimique (bien adapté aux gros systèmes d'équations) et une méthode de Runge-Kutta-Fehlberg (4ème ordre) pour l'avancement des échanges d'énergie thermique.

Ainsi à chaque pas de temps, Eilmer3 réalise une série d'opérations en utilisant ces méthodes tout en tenant compte des conditions aux limites. Le processus de résolution est donné en annexe.

3.7.2 Code CFD++

CFD++ est un logiciel commercial de mécanique des fluides numérique conçu par l'entreprise Metacomp technologies. Le champ d'application est assez large, incluant le domaine de l'aérospatiale, de l'automobile ou du biomédical. Le code utilise une formulation volumes finis mais pour des maillages non-structurés. Ces maillages peuvent être importés depuis un logiciel de génération de maillages comme IcemCFD ou autres. Pour ce code, la résolution temporelle peut se faire par un schéma implicite, impliquant une bonne stabilité et un temps de calcul réduit. Le calcul multiprocesseur (MPI) est aussi possible.

De nombreux modèles physiques sont disponibles dans CFD++ comme le modèle de turbulence, multiphasique, de transferts de chaleurs conjugués... et particulièrement le modèle à deux modes de température (conformément à [17]), qui incorpore aussi l'équation de conservation d'énergie vibration-electronique (14) dans les équations gouvernantes du fluide.

Les schémas implémentées dans le code pour l'intégration temporelle sont: Euler Explicite, Runge-Kutta d'ordre 2 explicite (version normale et multi-stage), Runge-Kutta explicite d'ordre 4 et Euler implicite. Les bons schémas sont sélectionnées par CFD++ dans le cas d'une résolution stationnaire ou instationnaire. Selon CFD++, ceci est lié en grande partie au fait que, une méthode d'intégration n'est pas toujours optimale dans toutes les situations: les méthodes efficaces pour les problèmes à dépendance temporelle sont généralement inefficaces pour résoudre des problèmes stationnaires et les méthodes efficaces pour les problèmes stationnaires sont généralement inexactes ou non physiques lorsqu'elles sont appliquées directement à des écoulements transitoires.

3.7.3 Module de rayonnement

Le module permet l'implémentation du phénomène de rayonnement pour des simulations d'écoulements compressibles sur des gaz soumis à de hautes températures. Il a été développé par le groupe de mécanique des fluides numérique de l'Université de Queensland. Le module regroupe plusieurs modèles spectraux disponibles pour Eilmer3 (mais aussi pour poshax3 qui est un autre code pour le calcul des propriétés d'écoulement après choc).

Le modèle spectral utilisé pour ce projet est le code incorporé PHOTAURA (bibliothèque d'Eilmer3). Il permet d'obtenir les coefficients spectraux moléculaires du terme source de rayonnement (49) en tenant compte du type de transition énergétique dont le rayonnement est issu, du processus collision-rayonnement des particules mais aussi de données expérimentales [36]. Le nom est formé du mot grec "photo" (lumière) et du mot latin "aura" (vent) représentatif des gaz traités, les plasmas.

Le module est compatible avec des modèles spectraux issus d'autres bibliothèques comme le code PARADE qui est la bibliothèque de "Fluid Gravity" (société de conseil scientifique basée au Royaume-Uni) et de l'ESA (Agence spatiale Européenne) ou le code SPRADIAN07 qui est la bibliothèque du KAIST (Institut supérieur coréen des sciences et technologies, qui fait partie des universités sud-coréennes).

4 Simulation, modèle "X2" (procédure)

4.1 Géométrie conditions aux bords

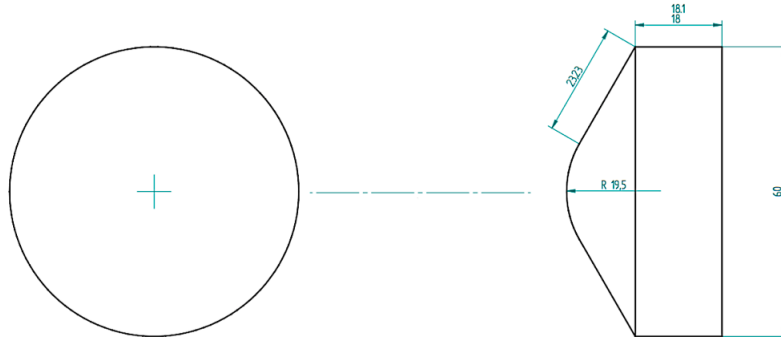


Figure 7: Dimensions de la capsule en (mm)

Du fait de la symétrie de révolution de la capsule, le domaine fluide sera simplement choisi comme la partie supérieure à l'axe horizontal sur un plan 2D $(\vec{x}; \vec{y})$. Il représente la partie avant en contact avec le bouclier thermique, les effets d'arrière corps n'étant pas simulés.

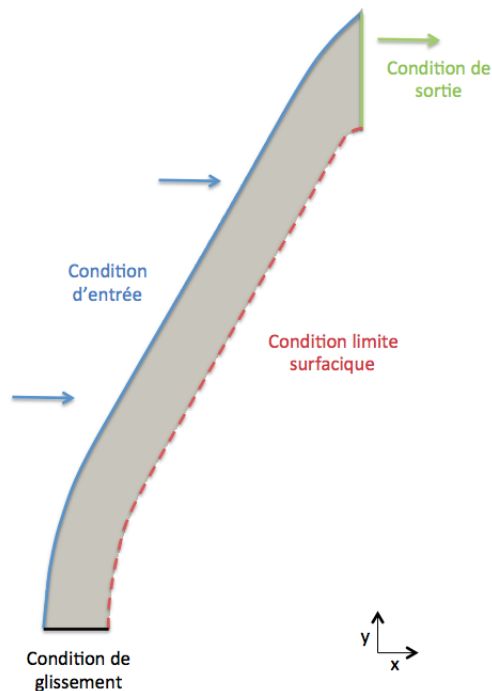


Figure 8: Schéma du domaine et des conditions limites en amont de la capsule

Les conditions limites utilisées lors de ce projet sont de différents types:

- **Condition d'entrée:** Propriétés de l'écoulement en infini amont. Elles représentent les conditions initiales du tableau (2).
- **Condition de sortie:** Condition de type sortie d'écoulement extrapolée. Cette condition utilise les cellules intérieures et n'a pas d'influence sur l'écoulement en amont.
- **Condition de glissement:** Représente un mur solide non-visqueux, pour modéliser l'effet de symétrie.
- **Condition surfacique:** Les conditions surfaciques sont celles d'un mur solide, visqueux (condition d'adhérence à la paroi), à une température fixe $T_{\text{mur}} = 600\text{K}$ conformément à la température de la surface de la capsule lors des expériences dans le tunnel d'expansion. Dans le cas où l'ablation ou la pyrolyse est modélisée, la condition surfacique est modifiée pour permettre l'injection d'espèces carbonisées dans l'écoulement, toujours en gardant une température fixe. Ceci va donc rajouter des termes sources d'expulsion massique lié au carbone avec un taux \dot{m}_c et aux gaz de pyrolyse avec un taux \dot{m}_g aux équations de conservation.

4.2 Conditions initiales

Composition du gaz		Propriétés physiques	
Gaz	Pourcentage	Propriété	Valeur
N2	0.7606	ρ_∞	$2.6 \times 10^{-3} \text{ Kg.m}^{-3}$
O2	0.2197	T_∞	2241 K
O	0.00606	p_∞	1688.3 Pa
NO	0.0136	M_∞	11.4
N	0	u_∞	10440 m.s ⁻¹

Table 2: Conditions de ré-entrée atmosphérique terrestre en tunnel d'expansion (Conditions 1)

Comme les deux codes on une approche de résolution différente (transitoire pour Eilmer3, stationnaire pour CFD++), la condition initiale de vitesse est définie légèrement différemment. Dans Eilmer3 elle est fixée sur la condition limite d'entrée, et nulle dans le domaine. Dans CFD++ elle est spécifiée et en entrée et dans le domaine.

Des conditions initiales légèrement différentes on été utilisées dans CFD++ (Conditions 2): $u_\infty = 10410 \text{ m.s}^{-1}$, $T_\infty = 2231 \text{ K}$, $\rho_\infty = 2.08 \times 10^3 \text{ Kg.m}^{-3}$, $p_\infty = 1344 \text{ Pa}$.

4.3 Maillages 2D

En géométrie 2D les maillages dont les cellules sont des triangles s'adaptent bien à des domaines compliqués. Les maillages formés de quadrilatères le peuvent moins mais permettent une meilleure qualité de résultats. Comme la géométrie de la capsule n'est pas trop complexe, des maillages quadratiques ont été réalisés.

Un bon maillage quadratique doit suivre quelques règles de construction. Les tailles de cellules placées l'une à côté de l'autre dans une direction donnée ne doivent pas être trop différentes, l'augmentation doit être progressive. Les cellules doivent avoir un rapport d'aspect correct et avoir des coins les plus droits possibles, dans les zones où l'écoulement va subir des effets. De plus, un raffinement doit être fait dans les zones de forts gradients avec d'avantage de mailles.

De nombreux maillages ont été réalisés mais leur élaboration suivait un certain cheminement. Chaque maillage fut d'abord conçu sans raffinement. Ceci permettait de déterminer la position du choc qui une fois connue, permettait de placer un raffinement au bon endroit.

Certaines simulations montraient un choc qui venait se placer pile sur la ligne de raffinement. La question était de savoir si ce placement n'était pas attiré par cette ligne. Afin de vérifier que le choc ne se déplaçait pas vers les cellules les plus petites, une zone de raffinement uniforme a été créée (comme pour le maillage ICEM) empêchant ainsi toute influence éventuelle.

4.3.1 Maillage Eilmer3



Figure 9: Maillage Eilmer3 avec épaulement

Les maillages Eilmer3 sont à entièrement codés en python, sans interface graphique. Les coordonnées des nœuds sont calculées et leur positions sont reliées par des fonctions pour former des sous-surfaces divisant le domaine. Celles-ci seront adoptées pour former des super-blocs, sur les-quels, conditions aux limites et répartition des cellules sont spécifiées.

Le maillage avec épaulement sur la figure (9) est divisé en 6 super-blocs (2 sur la longueur \times 3 sur la largeur). 4 d'entre eux sont eux mêmes composés de 12 blocs et les autres 8 blocs. Le total est de 64 blocs, afin de pouvoir faire un calcul (MPI) sur 64 processeurs. Des fonctions demi-cercles ont permis le tracé des courbes le long de la surface et en amont du nez, autrement des courbes de Bezier ont été utilisées pour suivre le choc au niveau de l'épaulement.

Le même maillage sans la partie épaulement a aussi été créé, car l'épaulement n'avait pas d'influence sur l'écoulement en sortie et ce maillage permettait de gagner du temps de calcul. Représenté sur la figure (10), il est composé de 4 super-blocs d'épaisseur ajustable (2 sur la longueur \times 3 sur la largeur) aussi délimités par une combinaison de droites et des courbes de Bezier afin d'ajuster leur épaisseur. L'épaisseur est choisie en faisant en sorte que la limite entre deux super-blocs sur la longueur, se place pile sur le choc pour des questions de raffinement. Le nombre de blocs est aussi égal à 64.

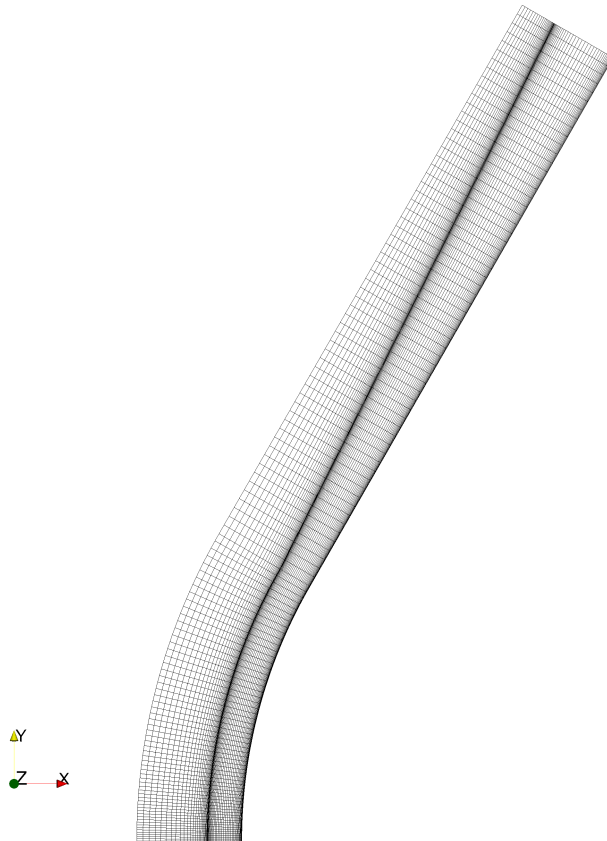


Figure 10: Maillage Eilmer3 sans épaulement

4.3.2 Maillage ICEM utilisé pour CFD++

Icem est un programme développé par Ansys permettant de réaliser des maillages tétraédriques ou hexaédriques, structurés ou non. L'exportation vers de nombreux différents solveurs dont CFD++ est ensuite possible.



Figure 11: Maillage Icem avec épaulement

Les coordonnées des nœuds définissant le domaine de calcul utilisé pour CFD++ ont été repris d'Eilmer3 pour être réutilisés dans Icem. Ceci a permis d'obtenir la même surface, et la même ligne de raffinement au choc. L'interface graphique permet de relier ces points entre eux pour former des bordures et sous-surfaces définissant des blocs de domaine (dans ICEM il y a juste des blocs principaux, il n'y a pas de super-blocs divisés en blocs comme dans Eilmer3).

Le maillage figure (11) est divisé en 9 blocs. Chacune des parois sont segmentées en un nombre défini de cellules réparties uniformément ou selon une loi de raffinement spécifique permettant un quadrillage personnalisé.

4.4 Définition du gaz

4.4.1 Choix du modèle de gaz

De nombreux modèles sont disponibles pour décrire le comportement du gaz dans Eilmer3 et CFD++. Pour les écoulements compressibles, il y a par exemple le modèle de gaz idéal, de deux modes de température ou de gaz réels. CFD++ comprends d'autres modèles pour écoulements incompressibles.

Le modèle de gaz choisi est celui à deux modes de température. Ce modèle est thermiquement parfait avec deux modes thermiques indépendants: une température T_{tr} qui régit les translation et les rotation des modes des particules lourdes, et une autre température T_{ve} qui régit les modes de vibration, électronique et d'électron libre.

4.4.2 Réactions chimiques et paramètres de réaction

Réactions	M	A	T_a [K]	n	Réactions	M	A	T_a [K]	n
Réactions de dissociation									
$N_2 + M \rightleftharpoons N + N + M$	Toutes	7.0 ²¹	113200	-1.6	$O_2 + M \rightleftharpoons O + O + M$	Toutes	2.0 ²¹	59360	-1.5
	C	3.0 ²²	113200	-1.6		C	1.0 ²²	59360	-1.5
	O	3.0 ²²	113200	-1.6		O	1.0 ²²	59360	-1.5
	N	3.0 ²²	113200	-1.6		N	1.0 ²²	59360	-1.5
	H	3.0 ²²	113200	-1.6		H	1.0 ²²	59360	-1.5
	e^-	3.0 ²⁴	113200	-1.6	$H_2 + M \rightleftharpoons H + H + M$	Toutes	2.2 ¹⁴	48300	0.0
						H_2	5.5 ¹⁴	48300	0.0
$C_2 + M \rightleftharpoons C + C + M$	Toutes	3.7 ¹⁴	69900	0.0					
$CN + M \rightleftharpoons C + N + M$	Toutes	2.5 ¹⁴	87740	0.0					
Réaction d'échange neutre									
$N + O \rightleftharpoons NO + N$	-	5.7 ¹²	42938	0.42	$CN + O \rightleftharpoons NO + C$	-	1.6 ¹³	14600	0.10
$NO + O \rightleftharpoons O_2 + N$	-	8.4 ¹²	19400	0.0	$CN + C \rightleftharpoons C_2 + N$	-	5.0 ¹³	13000	0.0
$CO + C \rightleftharpoons C_2 + O$	-	2.0 ¹⁷	58000	-1.0	$CO + C_2 \rightleftharpoons C_3 + O$	-	1.0 ¹²	41200	0.0
$CO + O \rightleftharpoons O_2 + C$	-	3.9 ¹³	69200	-0.18	$C_3 + N \rightleftharpoons CN + C_2$	-	1.0 ¹²	34200	0.0
$CO + N \rightleftharpoons CN + N$	-	1.0 ¹⁴	38600	0.0	$C_3 + C \rightleftharpoons C_2 + C_2$	-	1.0 ¹²	16400	0.0
$N_2 + C \rightleftharpoons CN + N$	-	1.1 ¹⁴	23200	-0.11	$C_2H + H \rightleftharpoons C_2 + H_2$	-	1.0 ¹²	16770	0.0
Réaction d'ionisation par impact d'électron									
$O + e^- \rightleftharpoons O^+ + e^- + e^-$	-	3.9 ³³	158500	-3.78	$C + e^- \rightleftharpoons C^+ + e^- + e^-$	-	3.7 ³¹	130720	-3.00
$N + e^- \rightleftharpoons N^+ + e^- + e^-$	-	2.5 ³⁴	168200	-3.82	$H + e^- \rightleftharpoons H^+ + e^- + e^-$	-	2.2 ³⁰	157800	-2.80
Réaction d'ionisation d'association									
$N + O \rightleftharpoons NO^+ + e^-$	-	5.3 ¹²	31900	0.0	$N + N \rightleftharpoons N_2^+ + e^-$	-	4.4 ⁷	67500	1.5

Table 3: Réactions chimiques considérées dans le modèle de Park (2001) [33], et paramètres de taux d'avancement associés

Espèce	Taux de réaction k_f	Formule de vitesse de réaction
CO	$0.63 \times \exp(-1160.0/T_{mur})$	$k_f[CO] \times \sqrt{(k_B T_{mur} / (2.0\pi m[CO])}$
CN	0.3	$k_f[CN] \times \sqrt{(k_B T_{mur} / (2.0\pi m[CN])}$
C ₃	$30.0 \times \exp(-21490.0/T_{mur})$	$k_f[C_3] \times \sqrt{(k_B T_{mur} / (2.0\pi m[C_3])}$

Table 4: Condition limite d'ablation

où $T_{mur} = 600K$, $N_A = 6.023 \times 10^{23}$ est le nombre d'Avogadro's ($1/mol$), $k_B = 1.381 \times 10^{23}$ est la constante de Boltzmann ($m^2.kg.s^{-1}.K^{-1}$) et m la masse molaire de l'espèce entre crochets (g/mol).

Les intégrales de collision doivent être définies dans les codes, afin d'obtenir une loi de mélange pour chacune des 20 espèces du modèle de Park (2001): C, O, N, H, CO, C₂, N₂, CN, NO, O₂, H₂, C₃, C₂H, C⁺, O⁺, H⁺, N⁺, NO⁺, N₂⁺, e⁻.

4.4.3 Échange d'énergie

Le terme source d'échange d'énergie V-T de l'équation (43) est calculé en fonction du temps de relaxation d'énergie translation-vibration $(\tau_{vib}^p)_{VT}$ exprimé en fonction de la section droite efficace pour la relaxation vibrationnelle σ_v . Le modèle de Millikan et White avec une correction pour gaz hautes températures (HTC), implémentée par Park, est utilisée pour le déterminer (sous Eilmer3).

Echange V-T	Modèle	Type	Paramètre σ_v
N2 $\sim\sim$ (N2, NO, O2, N, O)	Millikan-White (HTC) [25]	Park [20]	3.0×10^{-17}
NO $\sim\sim$ (N2, NO, O2, N, O)	Millikan-White (HTC) [25]	Park [20]	3.0×10^{-17}
O2 $\sim\sim$ (N2, NO, O2, N, O)	Millikan-White (HTC) [25]	Park [20]	3.0×10^{-17}

Table 5: Modélisation Air-TV Eilmer3

Le terme source d'échange d'énergie E-T de l'équation (44) est par ailleurs calculé en utilisant la fréquence de collision efficace ν_{es} , elle même exprimée en fonction de la section droite de collision d'échange d'énergie électron-neutre efficace σ_{es} .

Echange E-T	Modèle	Paramètres σ_{es}		
e ⁻ $\sim\sim$ N ⁺	Appleton-Bray: ion	-	-	-
e ⁻ $\sim\sim$ O ⁺	Appleton-Bray: ion	-	-	-
e ⁻ $\sim\sim$ N	Appleton-Bray: neutre [40]	5.0×10^{-20}	0.0	0.0
e ⁻ $\sim\sim$ O	Appleton-Bray: neutre [40]	1.2×10^{-20}	1.7×10^{-24}	-2.0×10^{-29}
e ⁻ $\sim\sim$ N ₂	Appleton-Bray: neutre [40]	7.5×10^{-20}	5.5×10^{-24}	-1.0×10^{-28}
e ⁻ $\sim\sim$ NO	Appleton-Bray: neutre [40]	1.0×10^{-19}	0.0	0.0
e ⁻ $\sim\sim$ O ₂	Appleton-Bray: neutre [40]	2.0×10^{-20}	6.0×10^{-24}	0.0

Table 6: Modélisation Air-TE Eilmer3

4.4.4 Modèle de diffusion

La diffusion est le processus visant à rendre homogène un milieu dont la densité est non homogène. Elle est caractérisée par le vecteur flux massique de diffusion de chaque espèce \vec{J}_s , qui est utilisé dans l'équation de conservation de la masse (6) et pour calculer le terme source de flux de diffusion dans l'équation (11).

Le calcul de \vec{J}_s peut se faire selon plusieurs modèles en utilisant les coefficients de diffusion binaires de l'équation (30), eux même calculés en fonction d'une loi de mélange spécifique. Parmi ces modèles il y a notamment la loi de Fick ou l'approche diffusion de Stephan-maxwell [7].

Dans le cas de plasmas faiblement ionisés, le flux de diffusion des ions et des électrons doivent être initialement différents du fait de la différence de masse de ces espèces, cependant l'interaction qui peut se produire entre elles peut introduire un phénomène amenant à l'égalisation de leur flux, c'est la diffusion ambipolaire.

Le modèle de diffusion binaire efficace auto-cohérent (en anglais "Self-Consistent Effective Binary Diffusion (SCEBD) model") proposé par Ramshaw et Chang [37] est utilisé dans le cadre du projet car il incorpore une correction ambipolaire et est bien adapté aux plasmas.

4.4.5 Configuration du modèle de transport de rayonnement

Propriété	Simulation plans-tangents
Résolution spectrale	345001 points

Table 7: Configuration de la simulation plans-tangents

Propriété	Simulation Monte-Carlo
Nombre de rayons émis par cellule	400 rayons
Résolution spectrale	28751 points

Table 8: Configuration de la simulation Monte-Carlo

4.5 Exécution d'une simulation

4.5.1 Lancement

Les paramètres de simulation restent encore à être spécifiés. Pour Eilmer3, il y a la condition CFL (qui peut être modifiée par l'utilisateur lors du calcul) ou le critère d'arrêt (comme le temps maximum). Pour CFD++, il s'agit du choix des schémas temporels (CFD++ s'occupe de les choisir en fonction du type de résolution, i.e en fonction de si la résolution est stationnaire ou in-stationnaire) et il s'agit des paramètres de schémas spatiaux (comme le temps resté en schéma de premier ordre avant de passer au second). L'ajustement de la condition CFL se fait automatiquement pour CFD++.

Les sections précédentes de ce chapitre décrivent la configuration choisie et les étapes à suivre pour la réalisation des simulations hypersoniques. Une présentation de comment mettre en place cette configuration dans Eilmer3 sous serveur "Bellatrix" (Bellatrix étant le serveur de l'EPFL) est donnée en annexe. Pour CFD++, tout se fait dans l'interface utilisateur. Le maillage non structuré provenant d'ICEM et le fichier de réactions présents dans le répertoire courant doivent être chargés. Un programme matlab "ReactionConvert.m" a été développé et permet de convertir les fichiers de réaction Eilmer3 en fichiers de réaction CFD++ (en annexe). La procédure est expliquée dans le guide [24].

La simulation est donc configurée et peut être lancée soit en local soit sur un serveur de calcul, en multi-processeurs (MPI). Reste alors à vérifier la convergence itérative.

4.5.2 Convergence itérative

Afin d'arriver aux résultats de simulation, la mécanique des fluides numérique implique l'utilisation de certains schémas itératifs. L'avancement en temps se fait en considérant une itération comme un pas de temps. En partant d'une condition initiale, les itérations se font jusqu'à ce que la solution ne varie plus (atteinte d'un état stationnaire). Afin d'évaluer s'il y a convergence itérative, évolution des résidus et solution sont à contrôler.

Les résidus des équations représentent les changements des équations entre chaque itération. Ces derniers sont généralement mis à l'échelle ou normalisés. Lors du calcul de la solution ces résidus doivent décroître et finir par se stabiliser afin de démontrer une convergence itérative, ceci est représentatif d'un état stationnaire. Autrement un seuil minimal de résidu raisonnablement petit (en général 3 ou 4 ordres de grandeur) peut être fixé par l'utilisateur, si cette stabilisation prend trop de temps.

L'arrêt de l'évolution de la solution au cours du temps est un autre indicatif de convergence. Sous Eilmer3 l'atteinte d'un état stationnaire fut observé après un temps entre 10 et 15 longueurs de corps d'écoulement (R/U_∞), soit une durée entre 18 et 28 μs .

La convergence itérative n'a pas toujours obtenue, sous CFD++. Les problèmes furent cependant résolus grâce à deux solutions différentes. Lors d'un premier calcul, il fut observé que si le temps de résolution avec un schéma d'ordre un est trop court, surtout pour les mélanges avec plusieurs réactions chimiques, les résidus ne convergeaient pas. En augmentant ce temps, ils ont pu converger. Bien que la résolution soit un peu plus longue, il est donc préférable d'attendre que les résidus aient déjà un peu convergé avec un schéma de premier ordre avant de passer à un schéma de second ordre. Lors d'un autre calcul, et après visualisation de la solution, un déplacement local non voulu de la forme de l'onde de choc au niveau du nez du véhicule fut observé. Cet effet caractéristique des simulations en hypersonique est numérique et est connu sous le nom d'effet "carbuncle" [22]. La solution fut donc d'activer l'option de contrôle d'effet carbuncle (rubrique solveur de Riemann dans CFD++).

Dans le cas d'Eilmer3 les problèmes de convergence furent essentiellement liés à des problèmes de maillages, un CFL trop élevé ou un changement trop brusque des paramètres. Un script matlab a été développé afin de tracer l'évolution des résidu et est placé en annexe.

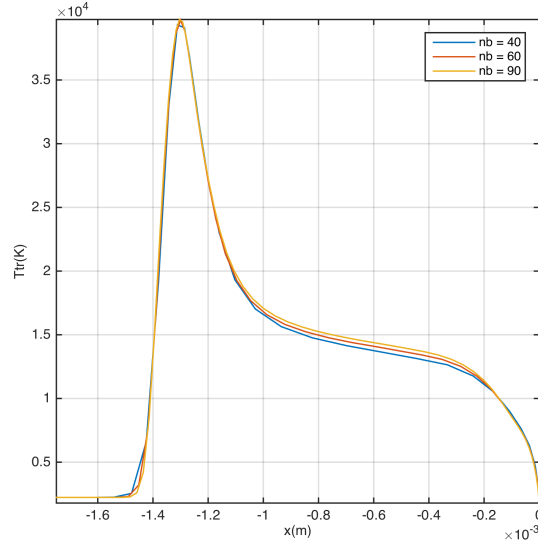
4.6 Influence du maillage (Eilmer3)

4.6.1 Convergence spatiale

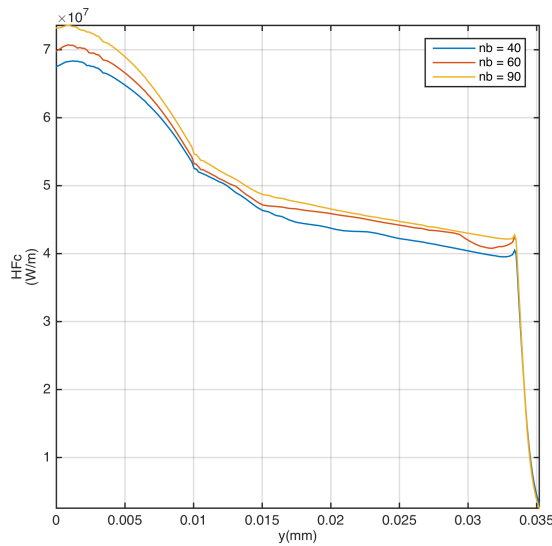
Comme l'extraction de données est faite au niveau de la ligne d'arrêt, une étude de convergence de maillage selon l'axe x a été effectuée. Seul le nombre de cellules a été modifié dans chaque maillage afin d'obtenir différents raffinages, le but étant de ne pas toucher aux lois de répartition des cellules (i.e les paramètres indiquant leur disposition).

La méthode appliquée afin d'examiner la convergence spatiale est tirée du livre de Roache [14] et est fondée sur l'extrapolation de Richardson. Sa mise en pratique a été simplifiée en choisissant un nombre de cellules spécifique selon x pour chaque maillage comparé. De même, le nombre de cellules par blocs a été choisi de façon à conserver

une homogénéité de répartition entre chaque maillages. Des programmes python ont été développés pour calculer ces nombres ainsi que des scripts matlab pour la mise en œuvre de la méthode de Roache. Ils se trouvent en annexe.



(a) Température trans - rot

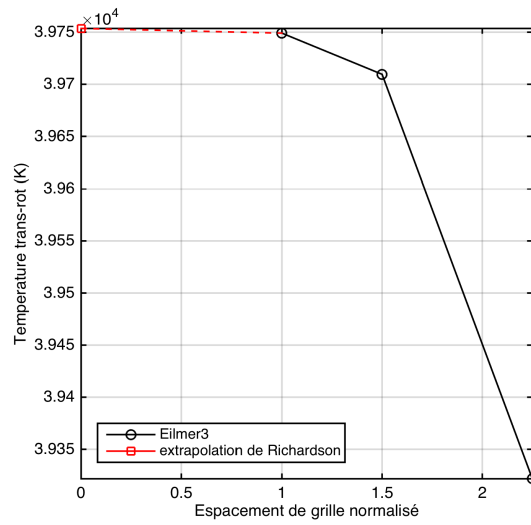


(b) Flux de chaleur par conduction.

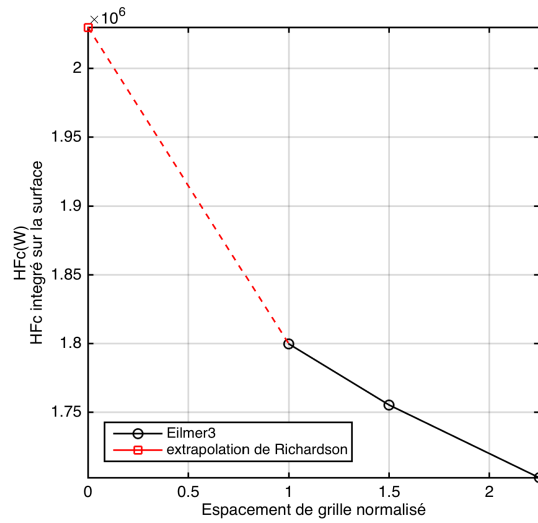
Figure 12: Comparaison des profils de température et de flux de chaleur pour différents raffinages ($nb = 40, 60, 90$), maillages Eilmer3 avec épaules, conditions initiales 2.

La température T_{tr} représentée sur la figure (12a) ne diffère plus vraiment au delà d'un nombre de 60 cellules sur la ligne de stagnation. Une maillage de 90 cellules n'est donc pas nécessaire. La figure (12b) ne montre pas de convergence spatiale flagrante des flux (ici le flux conductif) car les valeurs continuent d'augmenter.

L'extrapolation de Richardson est la valeur théorique correspondant au cas d'un nombre de cellules tendant vers l'infini ou d'une taille de maille tendant vers zéro. Pour des



(a) Valeurs de température maximales



(b) Intégration du flux sur toute la surface

Figure 13: Extrapolation de Richardson

maillages de plus en plus fins, les températures maximales et l'intégration du flux de chaleur par conduction sur la surface de la capsule (pour avoir une donnée d'ensemble) ont été tracés. La courbe figure (13a) montre que les valeurs maximales de température T_{tr} rencontrées dans la couche de choc ont un comportement asymptotique et tendent vers l'extrapolation de Richardson mais ce n'est pas le cas pour la courbe du flux intégré représentée figure (13b).

La méthode de Roache, tableau (10), soutient cette constatation, car une valeur proche de 1 est obtenue pour $T_{tr,max}$ indiquant un placement dans la gamme asymptotique de convergence, ce qui confirme le fait qu'un des maillages utilisé (60 cellules sur la ligne de stagnation) était suffisant pour assurer une convergence spatiale. L'intégrale du flux de chaleur par contre ne s'y situe pas, ce qui soutient le fait qu'aucune convergence notable n'était perçue sur la figure (12b) ni (13b). Ceci pourrait venir du fait que la taille de maille utilisée en paroi est encore trop grande. Des mailles plus petites dans cette zone ont été utilisées dans les maillages des calculs suivants.

Nombre de cellules en x	Espacement normalisé	Température Maximum du profil en (K)	Integration du flux de chaleur par conduction sur la surface (W)
40	1	4.2253×10^4	1.5400×10^6
60	1.5	4.2176×10^4	1.4824×10^6
90	2.25	4.1504×10^4	1.3742×10^6

Table 9: Données utilisées pour l'étude de convergence spatiale

	Températures maximales	Flux de chaleur intégrés
Ordre de convergence	5.632637	0.435343
Extrapolation de Richardson	39753.563638	2029800.552163
Indices de convergence	0.014100 (étape 1-2)	15.981869 (étape 1-2)
GCI (en %)	0.138515 (étape 2-3)	19.579775 (étape 2-3)
Ratio final de Roache	1.000995	1.025308

Table 10: Méthode de Roache: étude de convergence spatiale (données calculées par le script nasagrdev.m)

4.6.2 Influence de la forme des cellules

Le tracé de flux de chaleur aux parois parfois montre une irrégularité proche de la ligne d'arrêt en formant une sorte de bosse. Une recherche de l'origine de ce phénomène a d'abord été effectuée en faisant varier le raffinement le long de la paroi proche de la ligne mais cela n'a pas eu d'impact. Une autre piste fut alors de modifier la forme du maillage. Le maillage étudié est divisé en blocs comme sur la figure (15). Trois blocs en forme d'arcs se situent l'un devant l'autre, au dessus de la ligne d'arrêt devant le nez. Leur épaisseur le long de la paroi a été modifiée afin d'observer l'influence de ce paramètre et il a été observé que si cette épaisseur reste constante dans la zone de couche de choc, le flux à la paroi est corrigé.

En effet, aucun bloc n'a d'épaisseur constante avec le maillage (15a) ou "l'effet bosse" est le plus prononcé, contrairement au maillage (15c) où l'épaisseur est constante pour chaque bloc dans le choc (le bloc en amont du choc n'en faisant pas partie). Le maillage (15b) n'a qu'un seul bloc correct. Il est donc nécessaire de garder une distance constante dans une direction normale à la paroi pour chaque bloc présent dans le choc, dans les zones courbées. Ceci permet d'avoir des cellules d'avantage orthogonales.

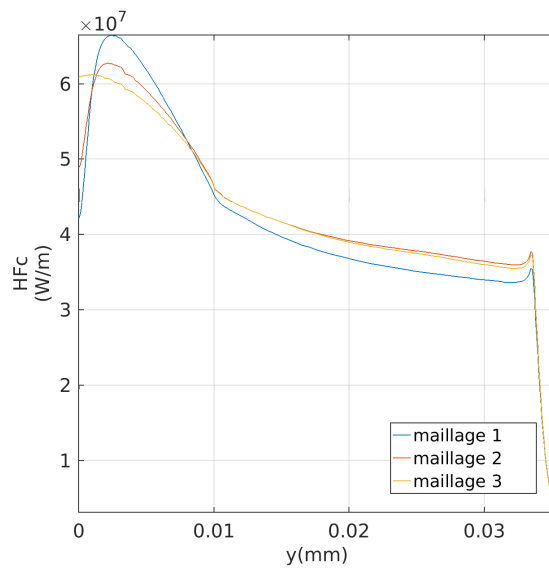


Figure 14: Tracé des flux de conduction le long de la surface du modèle pour les différents maillages avec épaulement d'Eilmer3, conditions initiales 2

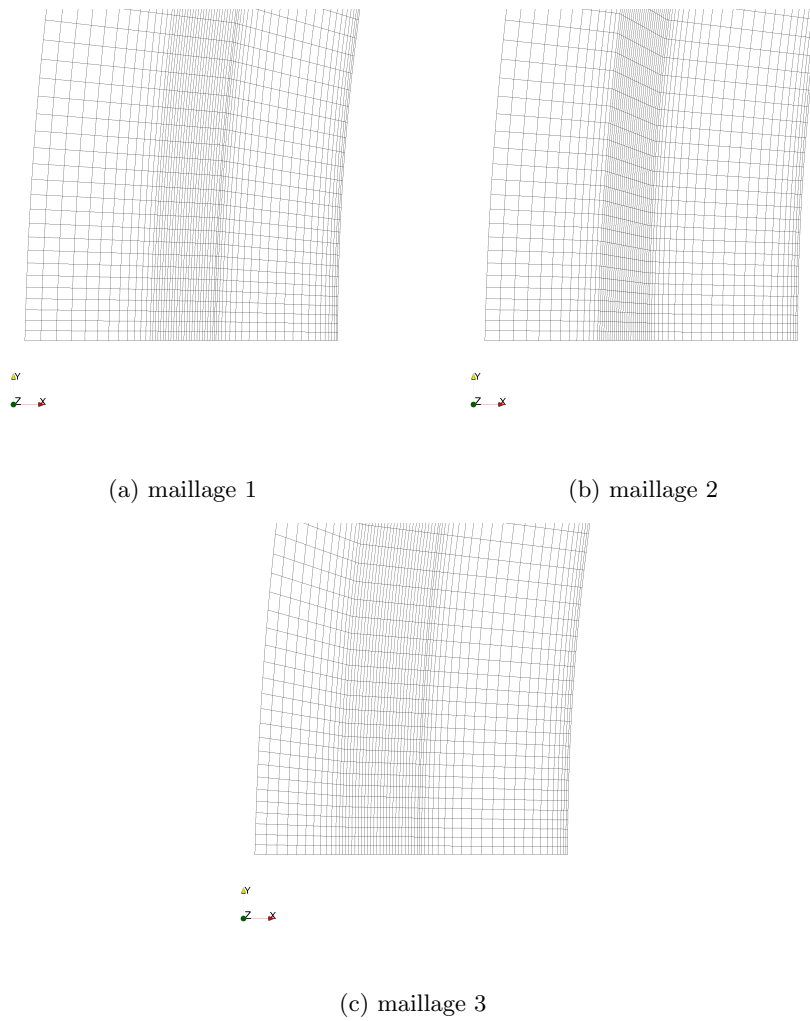


Figure 15: Maillages et formes des cellules proches de la ligne de stagnation.

5 Résultats

5.0.1 Eilmer3: Solution visqueuse (11 espèces)

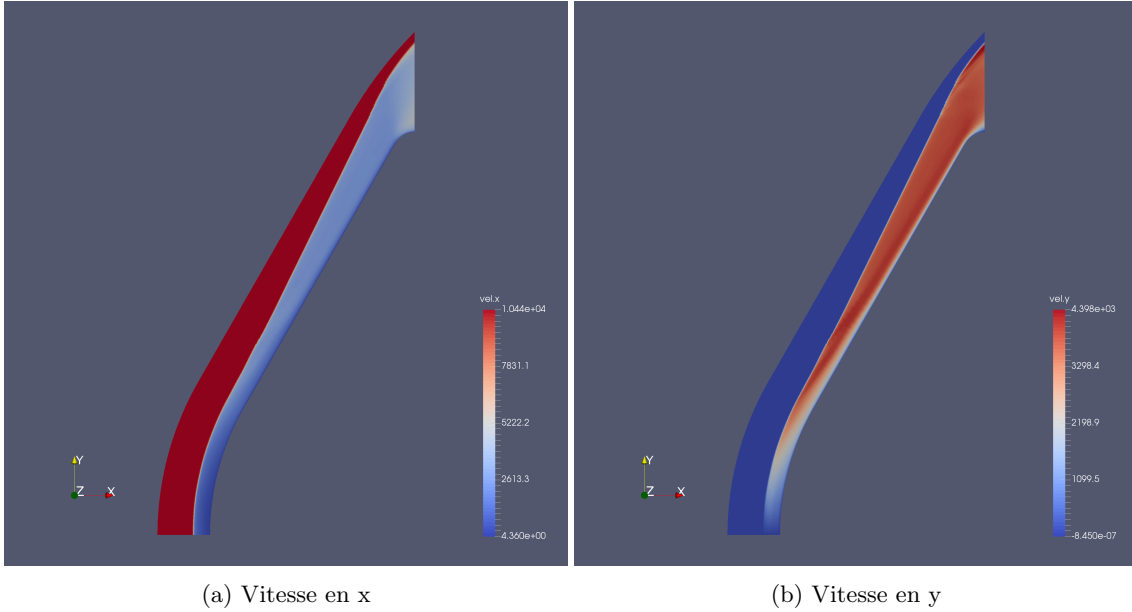


Figure 16: Champ de vitesse en amont de la capsule, calculé par Eilmer3 (gaz à 11 espèces, conditions initiales 1)

5.0.2 Eilmer3: Solution visqueuse avec ablation et rayonnement (20 espèces)

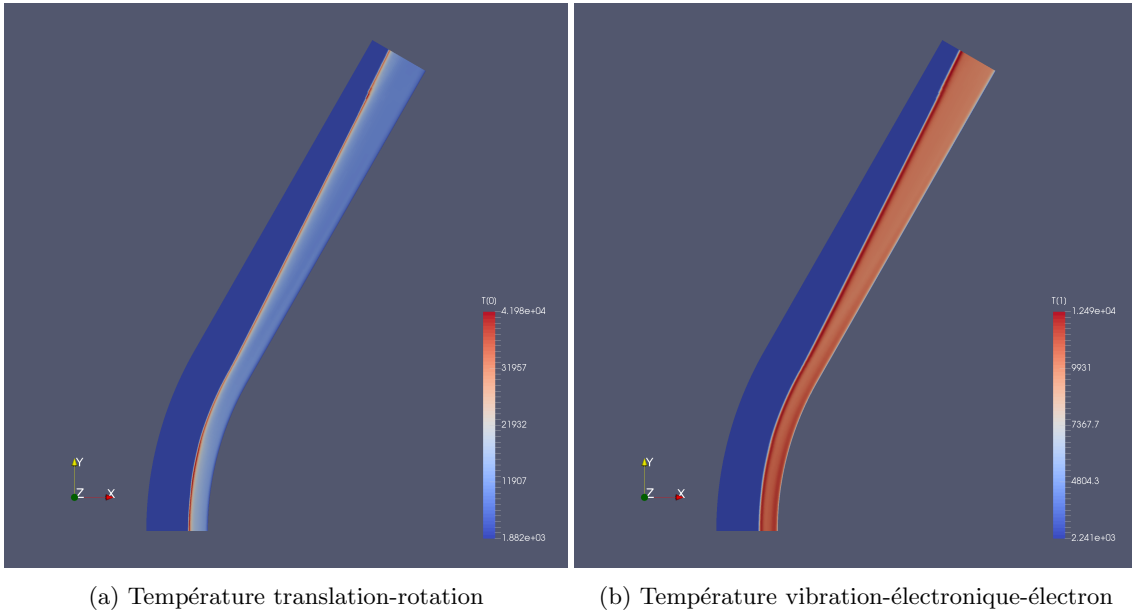
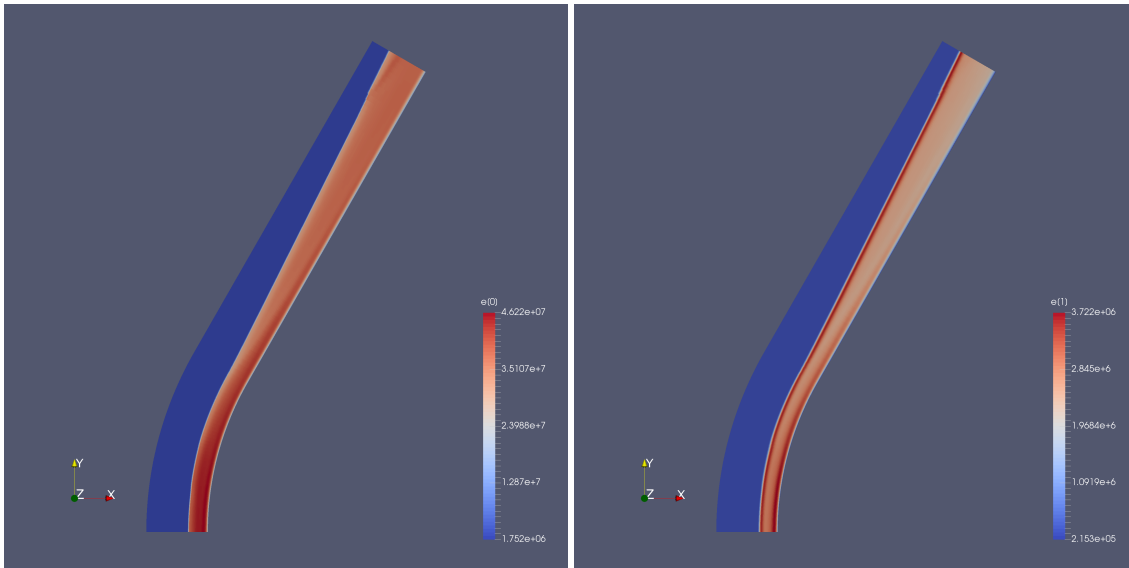


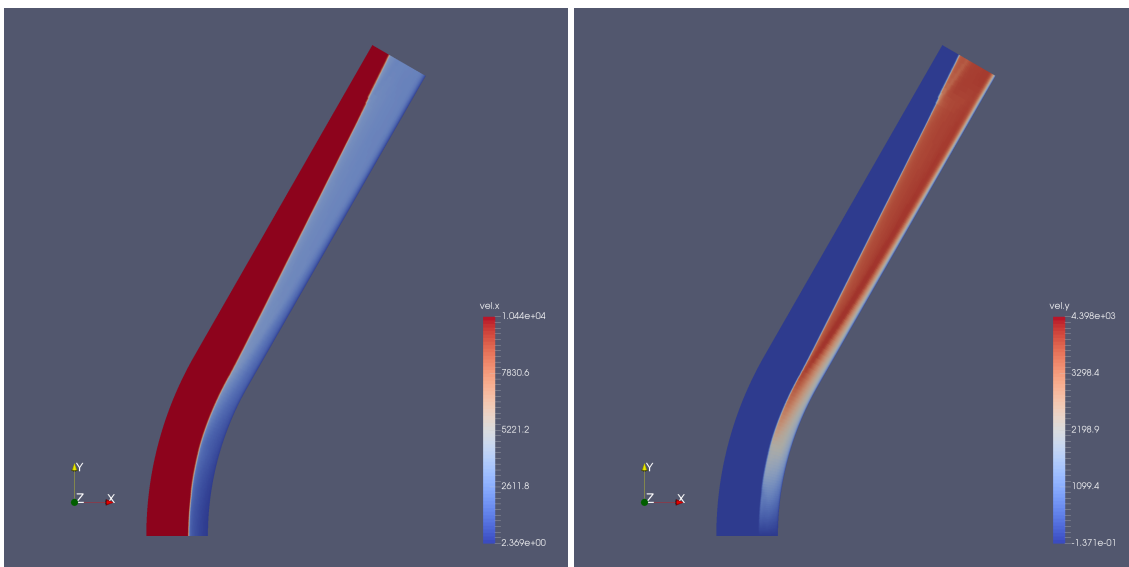
Figure 17: Répartition de la température en amont de la capsule calculé par Eilmer3 (gaz à 20 espèces, conditions initiales 1)



(a) énergie translation-rotation

(b) énergie vibration-électronique-électron

Figure 18: Répartition de l'énergie en amont de la capsule, calculé par Eilmer3 (gaz à 20 espèces, conditions initiales 1)



(a) Vitesse selon x

(b) Vitesse selon y

Figure 19: Champs de vitesse en amont de la capsule, calculé par Eilmer3 (gaz à 20 espèces, conditions initiales 1)

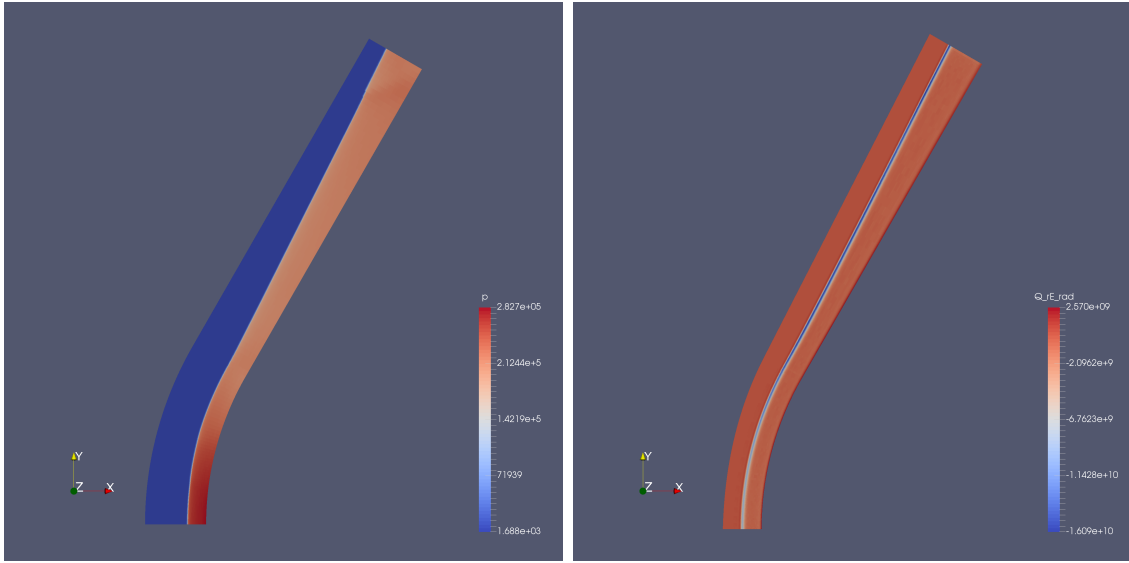
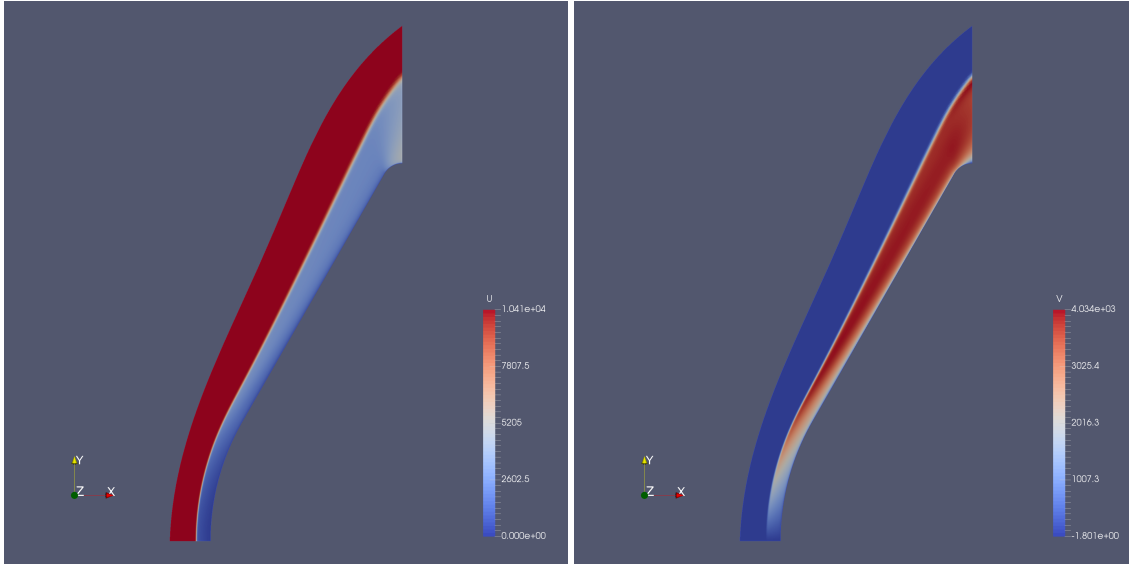


Figure 20: Champ de pression en amont de la capsule, calculé par Eilmer3 (gaz à 20 espèces conditions initiales 1) Figure 21: Champs de rayonnement en amont de la capsule, calculé par Eilmer3 (gaz à 20 espèces conditions initiales 1)

5.0.3 CFD++: Solution visqueuse (modèle 20 espèces) sans ablation et sans rayonnement



(a) Vitesse en x

(b) Vitesse en y

Figure 22: Champ de vitesse en amont de la capsule, calculé par CFD++ (gaz à 20 espèces, conditions initiales 2)

6 Discussion des résultats

6.1 Distance de choc

La figure (23) est l'image de la capsule en résine phénolique soumise à des conditions de ré-entrée terrestre et observée lors des tests dans le tube d'expansion X2. Les taches représentent l'ablation que subit la surface. L'image a été prise pour un temps en évolution stationnaire, et où l'écoulement n'est pas encore perturbée par les débris des diaphragmes ni par les gaz provenant de l'arrière du dispositif. Deux images différentes ont été utilisées (avec matériau ablatif et sans).

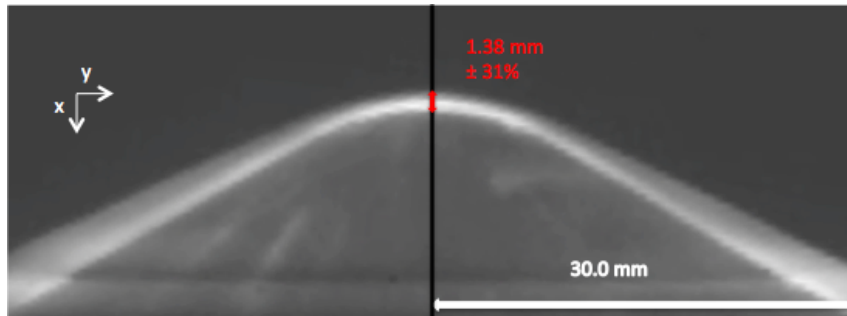


Figure 23: Image du petit modèle de capsule en résine phénolique lors des essais expérimentaux dans le tube X2 et distance de choc

Afin de mesurer l'épaisseur de la couche de choc, le nombre de pixels a été compté sur le rayon de l'objet. Comme cette distance est connue, un facteur d'échelle a pu être déterminé et a servi à évaluer la position du point d'arrêt sur l'image. Le nombre de pixels composant la couche de choc a pu par conséquent être compté plus précisément et la distance de choc a alors été obtenue en multipliant cette dernière valeur par le facteur d'échelle. Elle est égale à 1.38 mm (avec une incertitude relative de 31%, calculée par la formule aux dérivées partielles des incertitudes).

La valeur de référence obtenue par la simulation pour comparer avec l'expérimental, a été prise sur le tracé des pressions et est de 1.28 mm (incertitude relative de 12% si l'on considère la couche la plus grande et la plus petite rencontrée). En prenant en compte les incertitudes, la plage de valeurs expérimentales et issues de la simulation correspondent. De plus, l'approximation (2) de Ambrosio et Wortman, qui donnait après calcul 2.8 mm est du même ordre de grandeur.

	Facteur d'échelle	Distance de choc mesurée	Distance de choc calculée
Avec ablation	$0.06 \pm 1\%$	23 pixels $\pm 21\%$	1.38 mm $\pm 31\%$
Sans ablation	$0.06 \pm 1\%$	23 pixels $\pm 8\%$	1.38 mm $\pm 21\%$

Table 11: Valeurs d'épaisseur de choc mesurées sur deux images (avec et sans ablation) de la vidéo obtenue lors des essais de ré-entrée de la capsule en résine phénolique dans le tube X2

6.2 Comportement modal

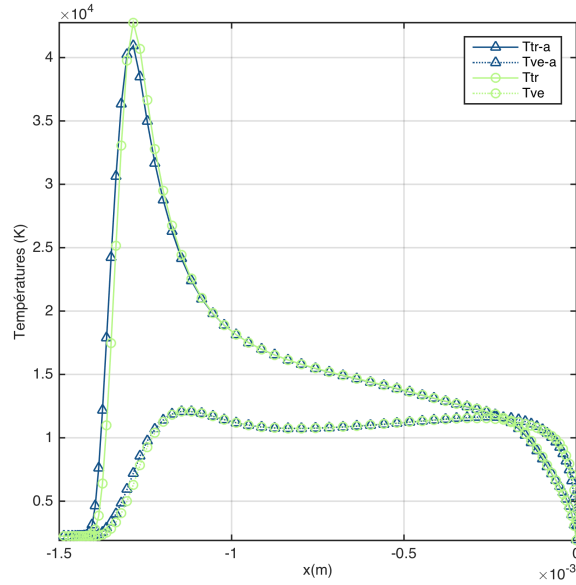


Figure 24: Températures modales le long de la ligne d'arrêt obtenues par Eilmer3 avec et sans ablation (modèle à 20 espèces, conditions initiales 1)

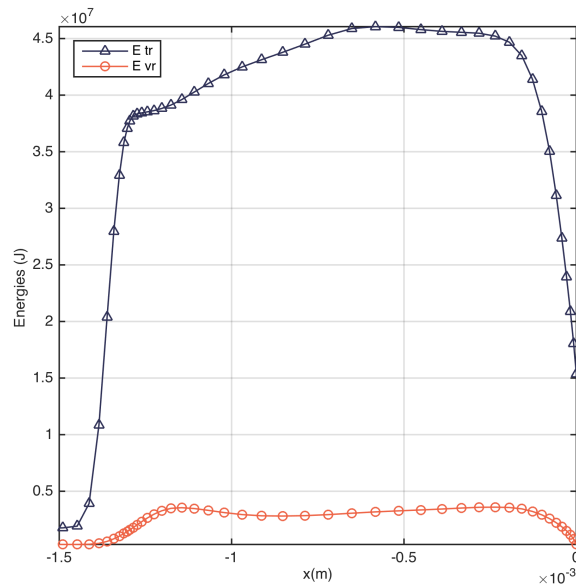


Figure 25: Énergies modales sur la ligne d'arrêt obtenues par Eilmer3 avec ablation (modèle à 20 espèces, conditions initiales 1)

Pour une formulation à deux modes, la température est découplée en une contribution translation-rotation et une contribution vibration-électronique-électron, avec $T_{trans} = T_{rot}$ et $T_{vib} = T_{el} = T_e$. La figure (24) représente le profil de température de ces modes sur la ligne de stagnation. Un pic caractéristique du choc est constaté pour le mode T_{tr} tandis

que le mode T_{ve} n'atteint pas les mêmes valeurs. Le long de la couche de choc, les deux modes de température ne sont pas égaux et par conséquent il n'y pas d'équilibre pour la température. Cependant en se rapprochant de la paroi, le gaz essaye de retrouver un état d'équilibre car les échanges d'énergie thermique font que les deux températures essaient de se rejoindre. Il est observé que l'ajout d'espèces ablatives refroidissent le pic de température d'environ 1800K.

En se heurtant au gaz atmosphérique, l'énergie cinétique du véhicule est transférée aux différents degrés de liberté des molécules. La figure (25) représente la répartition des ces modes d'énergie et montre que les modes translation-rotation sont beaucoup plus élevés et donc plus facile à exciter. L'énergie de translation-rotation ne se comporte pas comme la température le long de la couche de choc, contrairement à l'énergie de vibration-électronique-électron. Ceci vient de la contribution d'énergie cinétique lors de son calcul dans l'équation (67).

6.3 Comportement chimique

6.3.1 Répartition des espèces

La figure (26) montre l'évolution des fractions molaires des espèces le long de la ligne d'arrêt. Lorsque sur cette ligne cette grandeur ne varie pas, cela signifie que l'espèce reste en équilibre chimique, il n'y a ni production, ni destruction.

Il est observé sur la figure (26a) que l'écoulement arrivant sur la capsule en infini amont est bien composé d'air avec essentiellement des molécules de diazote N_2 et du dioxygène O_2 . En passant à travers le choc, la fraction molaire de ces molécules chute et celle des espèces monoatomiques augmente, ce qui montre une dissociation des particules lourdes. En approchant de la surface, la quantité de N_2 augmente de nouveau, ce qui indique que cette molécule se recombine.

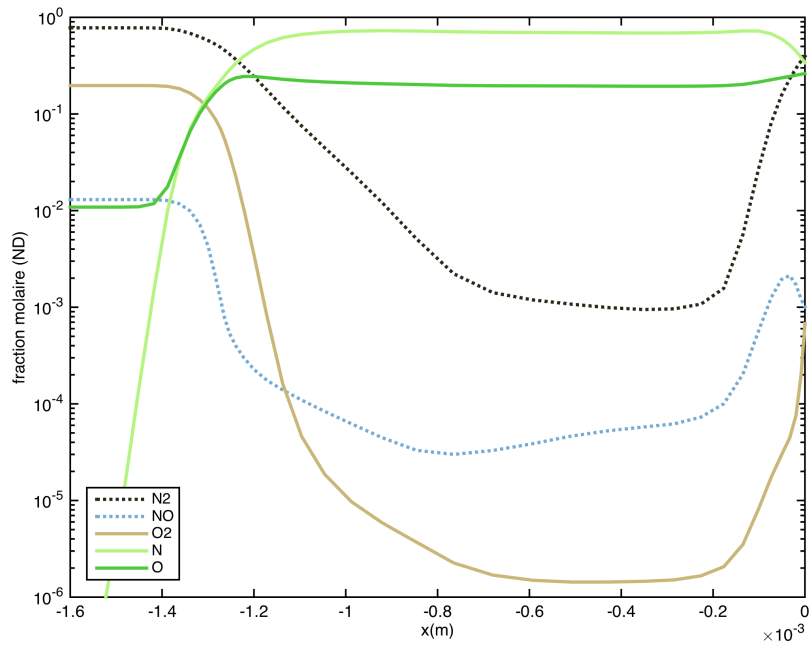
En traversant le choc, les molécules se dissocient aussi pour former des ions. Par conséquent les fractions molaires de ces derniers augmentent significativement (figure (26b)) et le gaz devient un plasma. Dans la zone de relaxation, les ions diatomiques tendent à disparaître et les ions monoatomiques à apparaître, du fait d'une continuation des processus de dissociation. Leur faible quantité montre un gaz faiblement ionisé. En approchant de la surface, les ions disparaissent car se recombinaient aussi en molécules.

6.3.2 Ajout d'espèces ablatives

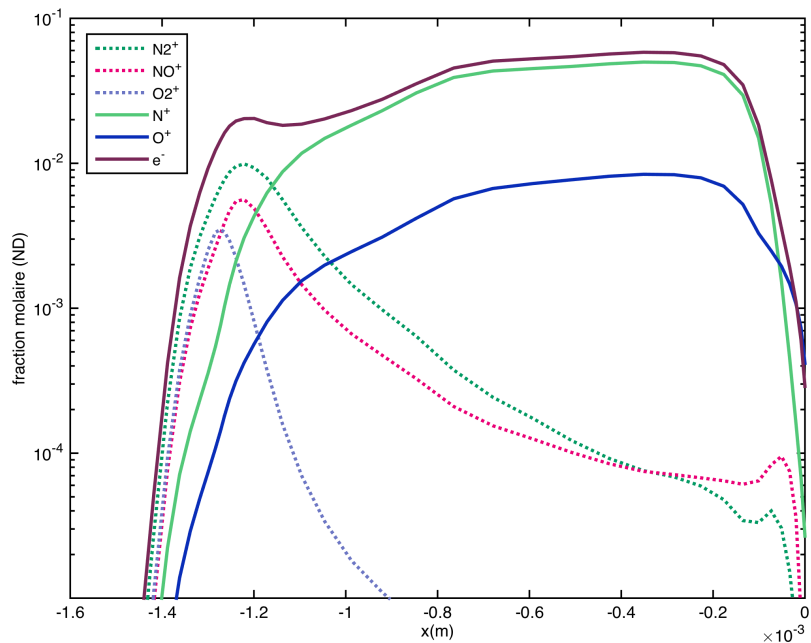
En comparant le cas avec et sans ablation, on retrouve une fraction molaire faible mais non nulle pour les espèces carbonisées comme le C, le CO ou le CN et encore plus faible pour le C_2 , le C_3 , et l'ion C^+ (quantité minime). Ces espèces proviennent de la modélisation de l'ablation, d'où le carbone est issu. En très proche paroi, les molécules de CN, C_2 , C_3 ou l'ion C^+ voient leur quantité augmenter légèrement pour redescendre ensuite alors que les espèces C ou CO décroissent d'emblée. Ceci montre que les espèces ablatives réagissent avec leur environnement.

Les espèces issues de l'ablation montrent un constant non-équilibre chimique. Leur concentration diminue jusqu'à ne laisser que quelques traces en s'éloignant de la paroi. Cela s'explique par le fait que les espèces ablatives sont produites au niveau de la surface,

et en s'éloignant elle réagissent pour former d'autres espèces ou se dissipent. Les réactions ont donc lieu dans la zone proche paroi où ces molécules sont présentes, formant la couche limite visqueuse d'ablation.

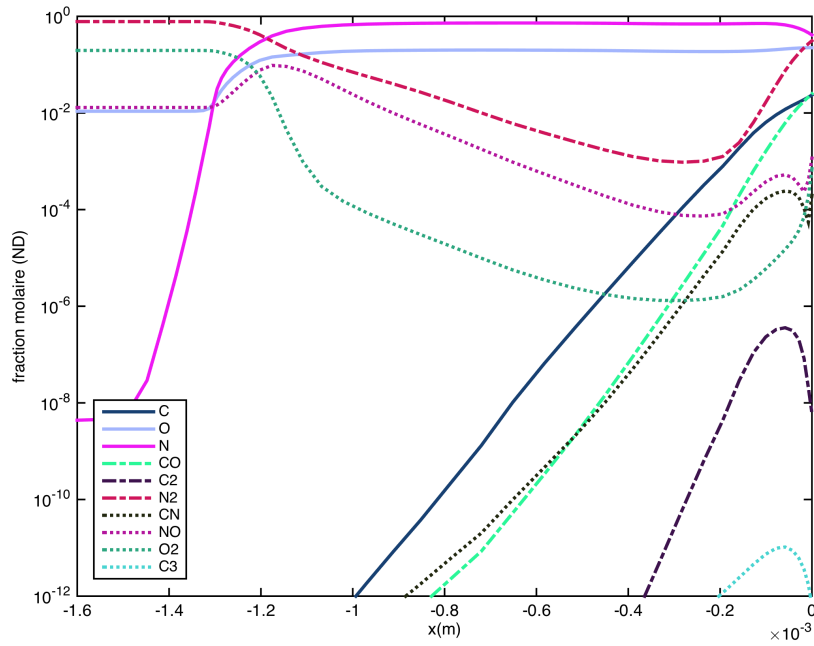


(a) fraction molaire des molécules

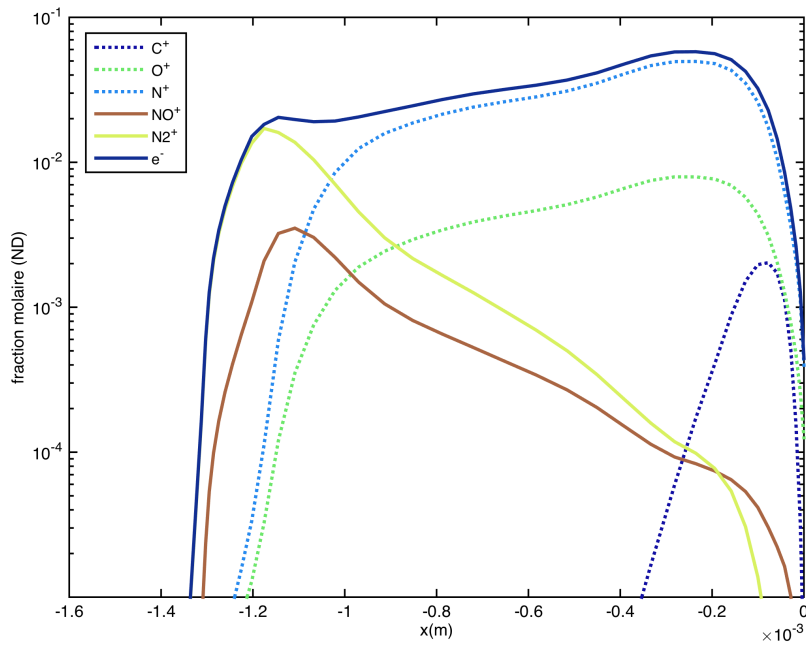


(b) fraction molaire des ions

Figure 26: Fractions molaires à travers la couche de choc obtenues par Eilmer3 (sans protection ablative, modèle à 11 espèces, conditions initiales 1)



(a) fraction molaire des molécules



(b) fraction molaire des ions

Figure 27: Fractions molaires à travers la couche de choc obtenues par Eilmer3 (avec protection ablative, modèle de Park 20 espèces, conditions initiales 1)

Les espèces hydrogénées modélisant la pyrolyse n'ont pas été injectés lors des simulations de ce projet, il est donc normal de ne pas retrouver de H , H^+ , H_2 ou C_2H . Cela ne devrait pas avoir de grandes conséquences car lors des expériences faites dans le tube X2, ces espèces n'ont pas pu être détectées et cela pourrait s'expliquer par un temps de test trop court, ne permettant pas aux espèces d'atteindre le plasma. L'ajout de cette condition doit cependant aussi être étudiée.

6.4 Comparaison Eilmer3 - CFD++

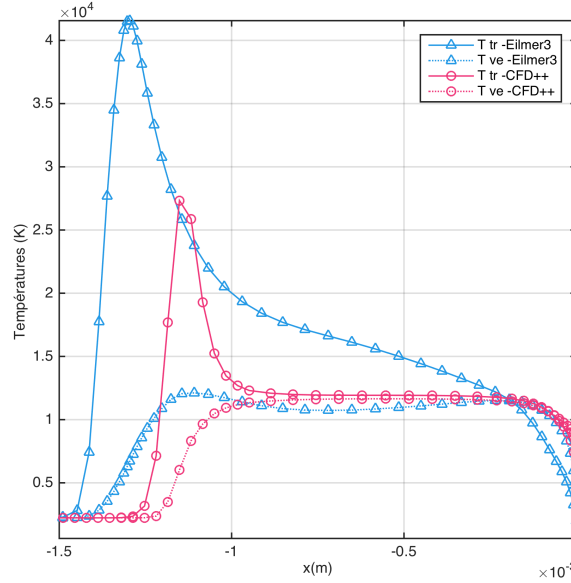


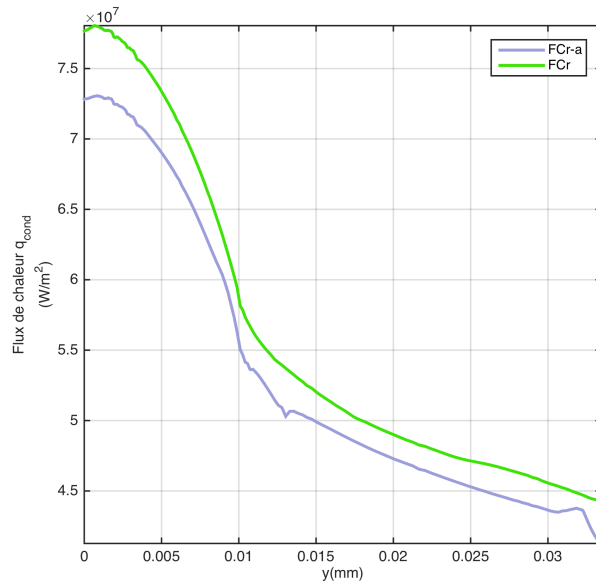
Figure 28: Comparaison des températures modales sur la ligne d'arrêt obtenues par CFD++ et Eilmer3 (sans d'ablation, conditions initiales 2)

Des différences notables sont observées entre les deux codes. Le profil de température en figure (28) montre un équilibre entre les modes thermiques dans la couche de choc alors que pour Eilmer3, un non-équilibre est représenté en figure (24). Des conditions initiales un peu différentes de celles présentées tableau (2) ont été utilisées pour le calcul Eilmer3 et CFD++ (conditions initiales 2).

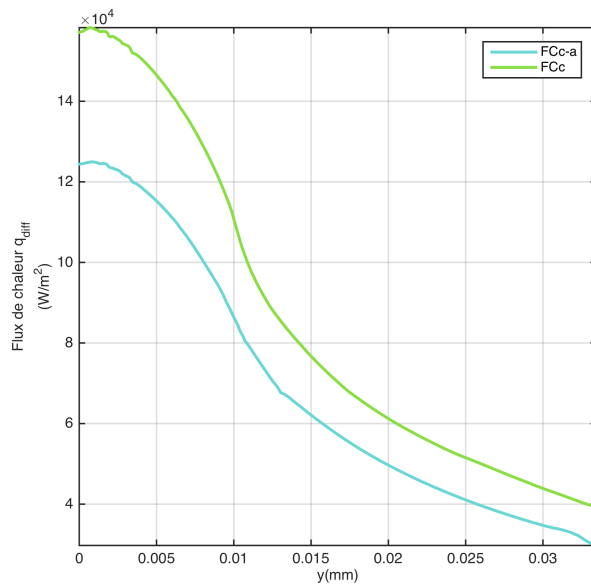
Les équations de conservation du modèle à deux modes de température dérivent des équations de Navier Stokes de la même façon pour les deux codes. La même équation de conservation de l'énergie vibratoire-électronique (14) est établie et implémentée. Mais les termes source d'échange d'énergie thermique de cette équation ou de leurs paramètres associés sont intrinsèques au modèle et non accessibles. La différence pourrait provenir du terme Ω_{VT} ou du terme Ω_{TE} d'Appleton et Bray (44) qui selon le manuel d'utilisation de CFD++, n'a pas été implémenté. Il est donc possible que les échanges entre les modes thermiques de vibration-translation ou de translation-électronique-électron ne soient pas modélisés ne permettant pas de retrouver deux modes de température séparés le long de la couche de choc.

De même, l'épaisseur de cette dernière est liée aux réactions dans le plasma et la différence peut aussi provenir de la façon dont les deux codes calculent les taux d'avancement de réaction k_f , k_b ou les termes sources d'échange d'énergie thermo-chimique Ω_{CX} .

6.4.1 Comportement des flux de chaleur et impact sur la surface



(a) Flux de chaleur par conduction



(b) Flux de chaleur par diffusion

Figure 29: Flux de chaleur par conduction (FC_c) et par diffusion (FC_d) le long de la surface du véhicule, obtenu par Eilmer3 avec ablation et sans ablation (modèle à 20 espèces, conditions initiales 1)

La figure (29) représente les flux de chaleur par conduction et diffusion reçus par la capsule. Comme les températures sont plus élevées au niveau de la ligne d'arrêt, on retrouve bien des flux de chaleur plus élevés dans cette zone. Le flux de chaleur par conduction (FC_c) d'ordre de grandeur 10^7 , reste plus important que le flux de chaleur par diffusion (FC_d), d'ordre de grandeur 10^5 .

Il est par ailleurs observé que les flux sont diminués d'environ 6.25 % pour le plasma contenant les espèces ablatives. En effet la présence de celles-ci affectent le mélange et ont donc une influence sur les coefficients de transport (paramètres de conduction et diffusion). Ceci confirme le fait que la couche limite formée par l'injection d'espèces ablatives provenant du matériau TPS, assure bien une protection contre toutes les formes de flux de chaleur. Ce processus de réduction de flux de chaleur par l'utilisation d'une couche limite est généralement nommé "blocage".

6.5 Rayonnement (Eilmer3)

6.5.1 Rayonnement du plasma le long de la ligne d'arrêt (Plans tangents)

La méthode des plans tangents a permis de tracer la divergence du flux de rayonnement ($\nabla \cdot \vec{q}_{rad}$) le long de la ligne d'arrêt. La figure (30) en est une représentation.

Avec ou sans ablation, un pic premier est observé au niveau de l'onde de choc. Ceci montre une forte émission par rayonnement du plasma qui s'explique du fait que la température de la zone y est beaucoup plus élevée. L'émission décroît ensuite après en raison de l'abaissement de la température.

En fin de couche de choc, l'émission redevient plus intense mais cette fois à cause de l'augmentation de la densité des espèces proche du point d'arrêt, d'avantage présentes. Il est par ailleurs observé que le plasma dans le cas ablation rayonne plus dans cette zone. Ceci est lié aux molécules carboniques, notamment celles du cyanure (CN) ou du monoxyde de carbone (CO) qui sont des espèces fortement rayonnantes.

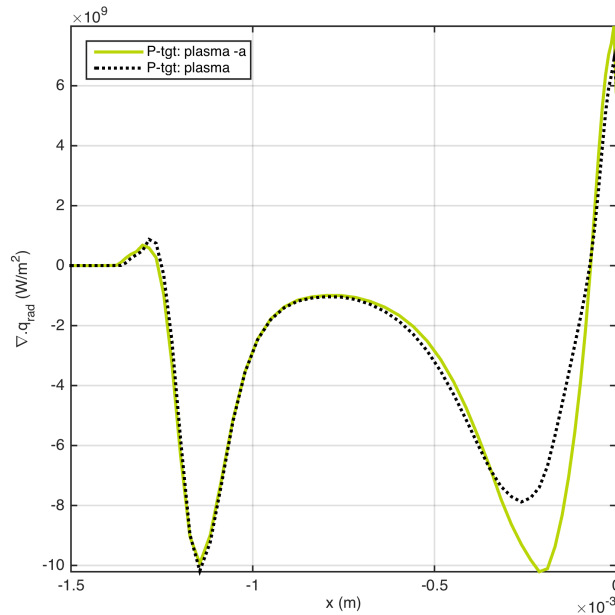


Figure 30: Comparaison de la divergence du flux radiatif par la méthode des plans tangents, dans le cas avec et sans ablation

En zone de très proche paroi la température chute, et par conséquent la zone devient une zone d'absorption. Comme voulu par le bouclier thermique, l'effet d'absorption est augmenté dans le cas avec ablation. Ceci est encore lié au molécules carboniques mais l'effet des conditions locales transforment leur propriétés.

6.5.2 Nombre de Goulard: couplage du rayonnement au champ d'écoulement

Grâce au nombre de Goulard, il est possible d'évaluer le couplage entre le rayonnement et la dynamique du fluide.

$$\Gamma = \frac{2q_{rad}}{1/2\rho_{\infty}u_{\infty}^3} \quad (68)$$

Il représente le rapport du flux de rayonnement provenant de la couche de choc par le flux de l'écoulement à l'infini amont. Si le nombre de Goulard est inférieur à 0.01, le rayonnement est considéré non couplé au fluide [8]; en d'autres termes, le rayonnement n'a pas d'impact sur le fluide. Il ne serait donc pas nécessaire en théorie de faire successivement des calculs de rayonnement entre deux solutions, en avançant dans le temps, comme cela est fait pour une simulation Monte-Carlo (voir annexe).

D'après le tableau (12) ce nombre est supérieur à 0.01 dans le cas d'une ré-entrée atmosphérique terrestre pour la capsule étudiée, en conditions de tube d'expansion X2.

	Sans ablation	Avec ablation
$q_{rad} (W.m^{-2})$	198.669×10^5	219.144×10^5
Γ	2.69×10^{-2}	2.96×10^{-2}

Table 12: Flux radiatif obtenu par la méthode des plans tangents et nombre de Goulard associé

6.5.3 Rayonnement reçu par la surface de la capsule (Simulation Monte-Carlo)

Les calculs couplés faits par la méthode de traçage de rayons (méthode Monte-Carlo) a permis d'ajouter l'effet rayonnement aux calculs précédents et d'obtenir une solution calculée dans tout le domaine. La figure (31) montre que le flux de chaleur par rayonnement reçu par la surface avec et sans ablation ne diffèrent relativement que très peu. A noter que ces résultats ont été obtenues avec une résolution spectrale très limitée (les calculs pour une simulation de rayonnement Monte-Carlo ne fonctionnent que sur 16 processeurs et requièrent énormément de mémoire).

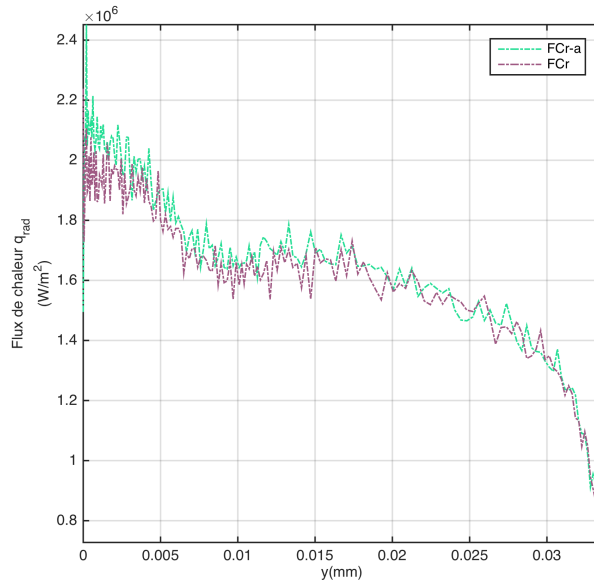


Figure 31: Comparaison de flux radiatifs sur la surface du véhicule calculé par la méthode de tracé de rayon Monte-Carlo par le biais de PHOTAURA dans le cas ablation et sans ablation

6.5.4 Identification spectrale des espèces et comparaisons (Plans tangents)

La comparaison de l'intensité des raies issues de la simulation par rapport aux intensités issues de l'expérience en tube X2 n'est pas directement possible. En effet la méthode des plans-tangents capture les spectres par rapport à une vue face à l'objet, alors que le spectromètre avait pris une vue de profil. De même les intensités issues de la base de donnée NIST ne sont pas toujours disponibles. La comparaison a alors été faite au niveau de la position des raies afin de déterminer les espèces.

L'oxygène, l'azote et le carbone sont les espèces principalement identifiées et correspondent aux données expérimentales ou issues des bases de données NIST ou PHOTAURA. Toutes les raies qui apparaissent en plus sur les spectres où il n'y avait pas d'ablation simulée sont forcément émises par des espèces carboniques. Il est alors possible que celles qui n'ont pas été identifiées comme étant l'atome de carbone soient issues de molécules. A confirmer, car la base de donnée NIST ne contenait pas une liste complète de raies.

Lors des essais en tube X2, les pics de CN étaient nombreux autour de 358 nm et 386 nm (figure(33)), mais ne sont ici pas observables, pour le spectre obtenu par simulation. Il est possible que ces raies soient présentes, mais seraient alors cachées par les nombreuses raies des autres espèces dont principalement le di-azote (N_2). Les spectres obtenus expérimentalement n'avaient pas montré de pics caractéristiques d'espèces issues de la pyrolyse, et les simulations effectuées n'ont pas pris en compte l'injection de ces espèces non plus. On ne peut donc pas comparer les raies. cependant il serait intéressant de pouvoir simuler cet effet.

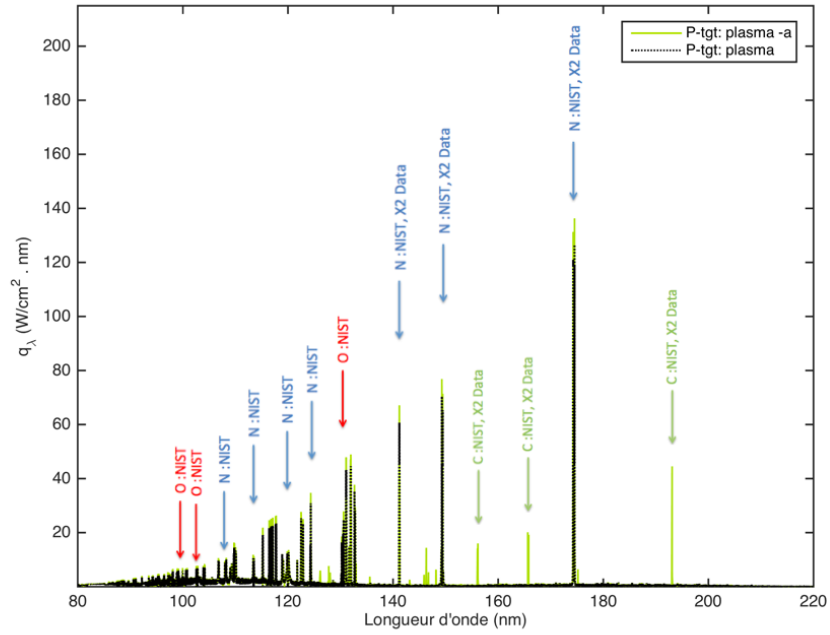


Figure 32: Identification des espèces sur le spectre entre 50 et 220 nm, obtenu par la méthode des plans tangents le long de la ligne d'arrêt, dans le cas avec et sans ablation

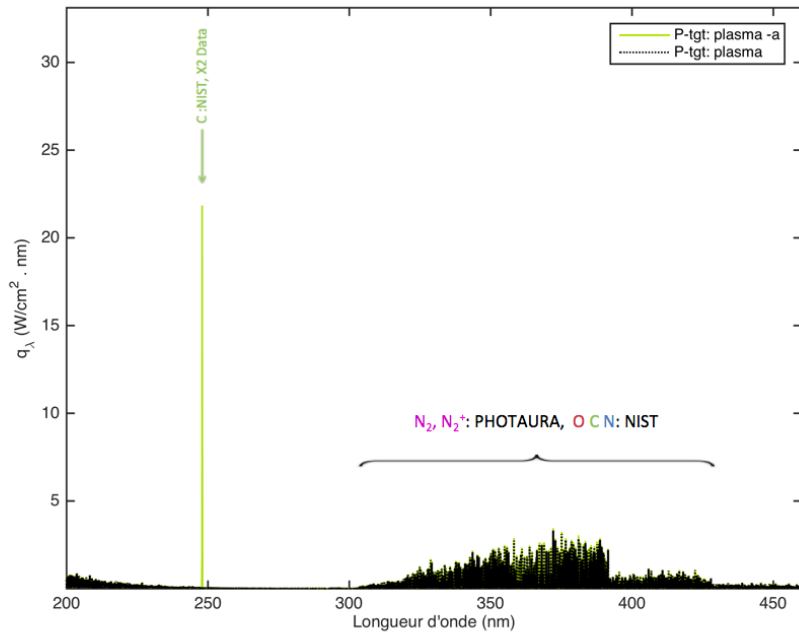


Figure 33: Identification des espèces sur le spectre entre 200 et 460 nm, obtenu par la méthode des plans tangents le long de la ligne d'arrêt, dans le cas avec et sans ablation

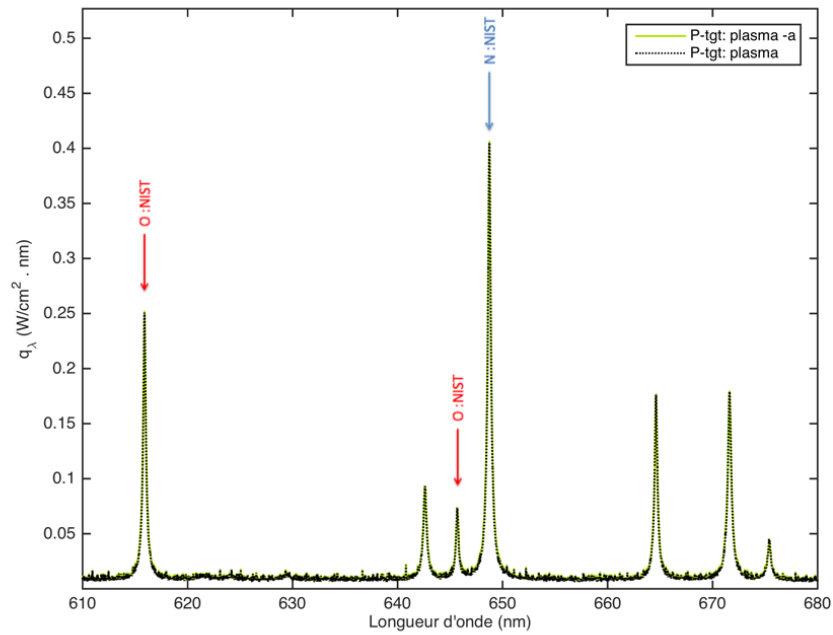


Figure 34: Identification des espèces sur le spectre entre 610 et 680 nm, obtenu par la méthode des plans tangents le long de la ligne d'arrêt, dans le cas avec et sans ablation

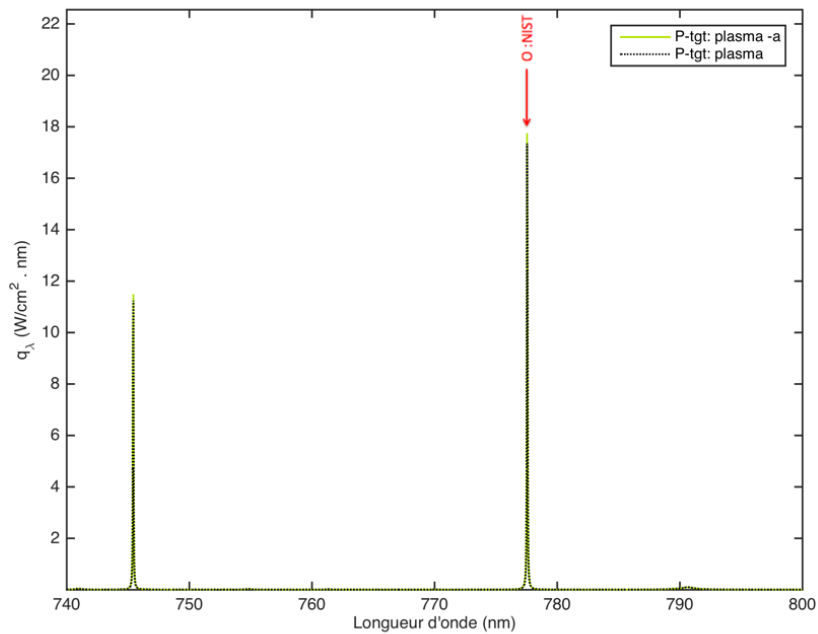


Figure 35: Identification des espèces sur le spectre entre 610 et 680 nm, obtenu par la méthode des plans tangents le long de la ligne d'arrêt, dans le cas avec et sans ablation

7 Conclusion

7.1 Grandes lignes

La compréhension des phénomènes de couplage radiatif avec les phénomènes d'ablation sur la couche de choc en amont de capsules est un domaine de recherche nécessaire pour la conception de nouveaux boucliers thermiques. Les simulations réalisées lors de ce projet sont la reconstruction numérique d'une expérience où une petite capsule, faite d'un matériau en résine phénolique, a été testée en conditions de ré-entrée terrestre. Des mesures spectrométriques avaient été obtenues afin d'identifier les espèces issues de l'ablation et de la pyrolyse de ce matériau, dans un tube d'expansion, seul dispositif permettant de retrouver des conditions quasi-similaires aux conditions de vol réelles.

Les équations de Navier Stokes, formulées pour deux modes de température, décrivent le comportement d'un fluide en ré-entrée atmosphérique. Elles font intervenir différents paramètres et termes sources devant être modélisés. La physique statistique et une modélisation à l'échelle microscopique sont les principaux outils pour mener cela à bien.

Les solutions caractéristiques du champ d'écoulement furent obtenues à l'aide des codes Eilmer3 et CFD++. Les équations ont pu être résolues spatialement par une approche volumes finis et temporellement par une méthode explicite sur maillages structurés (pour Eilmer3) ou implicite sur maillages non structurés (pour CFD++).

Les résultats obtenus décrivent bien le comportement attendu des plasmas en conditions de ré-entrée. Une séparation en deux modes de température fut observée et les réactions incluant des processus de dissociation ont bien été simulées et observés. Les résultats attendus par CFD++ n'ont cependant pas pu être obtenus et la cause est sûrement liée aux termes sources d'échange d'énergie thermique implémentés dans le code.

Il a été aussi observé que l'ajout d'espèces ablatives dans l'écoulement, pour cette configuration, diminue les flux de chaleur par conduction et diffusion impactant la paroi, confirmant le fait que le matériau TPS modélisé joue un rôle protecteur.

Les spectres d'émission des espèces présentes dans le gaz de couche de choc du plasma de ré-entrée ont pu être comparés et montrent une bonne similitude avec les données du NIST ainsi que pour quelques données récupérées lors de l'expérience X2, notamment pour les espèces mono-atomiques. Les pics caractéristiques du cyanure (CN) n'ont cependant pas pu être retrouvés dans les résultats provenant de la simulation pour des longueurs d'ondes dans l'UV-Visible. Ces pics peuvent toutefois être dissimulés parmi les nombreuses raies provenant du N₂.

7.2 Travail futur

Une condition de pyrolyse peut être ajoutée aux simulations afin d'en étudier les effets.

Les spectres obtenus par la méthode des plans-tangents et ceux obtenus lors des essais dans le tube X2 ne sont pas issus de la même ligne de mire. La correspondance des spectres reste encore à être entreprise, afin de pouvoir comparer les intensités. Ceci permettrait éventuellement de mettre en valeur les pics de CN.

Bibliographie

- [1] J. Appleton and K. Bray. The conservation equations for a nonequilibrium plasma. *Journal of Fluid Mechanics*, 20(4):659–672, 1964.
- [2] B. Armaly and L. Sutton. Viscosity of multicomponent partially ionized gas mixtures, 1980. AIAA Paper 80-1495.
- [3] N. Banerji. A hybrid approach to radiation ablation coupling, January 2016.
- [4] N. Banerji and P. Leyland. Expansion tunnel ablation testing in venus entry conditions. AIAA Paper.
- [5] F.S. Billig. Shock-wave shapes around spherical-and cylindrical-nosed bodies. *Journal of Spacecraft and Rockets*, 4:822–823, 1967.
- [6] T. Young J. Boris. A numerical technique for solving stiff ordinary differential equations associated with the chemical kinetics of reactive-flow problems. *Journal of Physical Chemistry*, 81(25):2424–2427, 1977.
- [7] D. Bothe. On the maxwell-stefan approach to multicomponent diffusion. <https://arxiv.org/pdf/1007.1775.pdf>, July 2010. Darmstadt, Germany.
- [8] G. Palmer , G. Allen, C. Tang, J. Brown. Coupled fluids-radiation analysis of a high-mass mars entry vehicle, 2011. 42nd AIAA Thermophysics Conference, Fluid Dynamics and Co-located Conferences.
- [9] P. Cinnella and B. Grossman. Upwind technique for flow with multiple translational temperatures, 1990. 21st Fluid Dynamics, Plasma Dynamics and Lasers Conference, Fluid Dynamics and Co-located Conferences, AIAA Paper 90-1660.
- [10] A.W. Cook. Enthalpy diffusion in multicomponent flows. *Physics of Fluids*, 24(5), December 2008.
- [11] M.Lino da Silva, V. Guerra and J. Loureiro. Two-temperature models for nitrogen dissociation. *Chemical Physics*, 342(1-3):275–287, December 2007.
- [12] A. Martin, E.D. Farbar and I.D. Boyd. Numerical modeling of the cn spectral emission of the stardust re-entry vehicle, 2011. 42nd AIAA Thermophysics Conference, AIAA Paper, volume 3125.
- [13] O. Knab, H.H. Fruhauf, and S. Jonas. Multiple temperature description of reaction rate constants with regard to consistent chemical-vibration coupling, 1992. 27th Thermophysics Conference, Fluid Dynamics and Co-located Conferences, AIAA Paper 1992-2947.
- [14] P.J. Roache, K. Ghia and F. White. Editorial policy statement on the control of numerical accuracy. *ASME Journal of Fluids Engineering*, 108(1), March 1986.
- [15] R.J. Gollan and P.A. Jacobs. About the formulation, verification and validation of the hypersonic flow solver eilmer. *International Journal for Numerical Methods in Fluids*, 17(1):19–57, 2013.
- [16] C. Guldberg and P. Waage. Studies concerning affinity. *Forhandlinger: Videnskabs - Selskabet i Christinia*, 35, 1864.

- [17] P.A. Gnoffo, R.N Gupta and J.L. Shinn. Conservation equations and physical models for hypersonic air flows in thermal and chemical nonequilibrium. Technical report, NASA, 1989. Paper 2867.
- [18] R. N. Gupta, J. M. Yos, R. A. Thompson, and K.-P. Lee. A review of reaction rates and thermodynamic and transport properties for an 11-species air model for chemical and thermal nonequilibrium calculations to 30000 K. *NASA STI/Recon Technical Report N*, 90, August 1990.
- [19] K. Huber and G. Herzberg. *Molecular Spectra and Molecule Structure IV. Constants of Di-atomic Molecules*. Van Nostrand Reinhold, New York, 1950.
- [20] C. Park, J. Howe, R. Jaffe, and G. Candler. Review of chemical-kinetic problems of future nasa missions, ii: Mars entries. *Journal of Thermophysics and Heat Transfer*, 8(1):9–23, March 1994.
- [21] J.R. Cash, A.H. Karp. A variable order runge-kutta method for initial value problems with rapidly varying right-handsides. *ACM Transactions on Mathematical Software*, 16(3):201–222, 1990.
- [22] R. MacCormack. The carbuncle cfd problem, January, 2011. 49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Aerospace Sciences Meetings.
- [23] S. Macheret and W. Rich. Nonequilibrium dissociation rates behind strong shock waves: classical model. *Chemical Physics*, 174:25 – 43, 1993.
- [24] Metacomp Technologies. *CFD++ User Manual - Version 11.1*.
- [25] R. Millikan and D. White. Systematics of vibrational relaxation. *Journal of Chemical Physics*, 39(12):3209 – 3213, 1963.
- [26] M.F. Modest. *Radiative Heat Transfer*. Academic Press, 2 edition, 2003.
- [27] D.R. Mott. *New quasi-steady-state and partial-equilibrium methods for integrating chemically reacting systems*. PhD thesis, University of Michigan, 1999.
- [28] E.S. Oran and J. Boris. *Numerical Simulation of Reactive Flow*. Cambridge University Press, New York, USA, 2 edition, November 2000.
- [29] M. Wright, D. Bose, G. Palmer, and E. Levin. Recommended collision integrals for transport property computations, part 1: Air species. *AIAA Journal*, 43(12):2558–2564, 2005.
- [30] H.K. Tran, C.E. Johnson, D.J. Rasky, Frank C.L. Hui, M. Hsu, T. Chen, Y.K.Chen, D. Paragas and L. Kobayashi. Phenolic impregnated carbon ablators (pica) as thermal protection systems for discovery missions, 1997. NASA Technical memorandum 110440.
- [31] C. Park. *Nonequilibrium Hypersonic Aerothermodynamics*. John Wiley & Sons Inc., New York, 1990.
- [32] C. Park. Stagnation-point radiation for apollo 4. *Journal of Thermophysics and Heat Transfer*, 18(3):349–357, 2004.

- [33] C. Park. Chemical-kinetic parameters of hyperbolic earth entry. *Journal of Thermophysics and Heat Transfer*, 15(1):76–90, January 2001.
- [34] E. Pfender. Heat transfer from thermal plasmas to neighboring walls or electrodes. <https://www.iupac.org/publications/pac/1976/pdf/4802x0199.pdf>, 1976.
- [35] D.F. Potter. *Modelling of radiating shock layers for atmospheric entry at Earth and Mars*. PhD thesis, University of Queensland, 2011.
- [36] D.F. Potter. *A High Temperature Gas Radiation Module for Compressible Flow CFD: Theory and User's Guide*, January 2013.
- [37] J.D. Ramshaw. Self-consistent effective binary diffusion in multicomponent gas mixtures. *Journal of Non-Equilibrium Thermodynamics*, 15(3):295–300, 1990.
- [38] D. Siva K. Reddy and Krishnendu Sinha. Hypersonic turbulent flow simulation of fire ii reentry vehicle afterbody. *Journal of Spacecraft and Rockets*, 46(4):745–757, 2009.
- [39] M. W Winter, R. D McDaniel, Y. Chen, Y. Liu, D. Saunders, and P. Jenniskens. Radiation modeling for the reentry of the hayabusa sample return capsule, 2012. 50th AIAA Aerospace Sciences Meeting.
- [40] P.A. Gnoffo, R.N. Gupta, J.L. Shinn. Conservation and physical models for hypersonic air flows in thermal and chemical nonequilibrium. Technical report, NASA, Febuary 1989.
- [41] C. Treanor and P. Marrone. Effect of dissociation on rate of vibrational relaxation. *Physics of Fluids*, 5(9):1022 – 1026, 1962.
- [42] W. Vincenti and C. Kruger. *Introduction to Physical Gas Dynamics*. John Wiley and Sons, New York, USA, 1965.
- [43] C. Wilke. A viscosity equation for gas mixtures. *Journal of Chemical Physics*, 18:517–519, 1950.

A Annexes

A.1 Expression des vecteurs utilisés dans Eilmer3

Les vecteurs de flux sont divisés en une contribution non visqueuse et visqueuse. La composante non visqueuse en équilibre thermique est:

$$\bar{F}_i = \begin{bmatrix} \rho u_x \\ \rho u_x^2 + p \\ \rho u_y u_x \\ \rho E u_x + p u_x \\ \rho e_{vm} u_x \\ \rho e_e u_x + p_e u_x \\ \rho f_s u_x \end{bmatrix} \hat{i} + \begin{bmatrix} \rho u_y \\ \rho u_x u_y \\ \rho u_y^2 + p \\ \rho E u_y + p u_y \\ \rho e_{vm} u_y \\ \rho e_e u_y + p_e u_y \\ \rho f_s u_y \end{bmatrix} \hat{j},$$

La composante visqueuse est:

$$\bar{F}_v = \begin{bmatrix} \tau_{xx} \\ \tau_{yy} \\ \tau_{xx} u_x + \tau_{yy} u_y + q_x \\ q_{x,vm} \\ q_{x,e} \\ J_{x,s} \end{bmatrix} \hat{i} + \begin{bmatrix} \tau_{xy} \\ \tau_{yx} \\ \tau_{xy} u_x + \tau_{yx} u_y + q_y \\ q_{y,vm} \\ q_{y,e} \\ J_{y,s} \end{bmatrix} \hat{j}.$$

où les contraintes visqueuses axisymétriques sont:

$$\begin{aligned} \tau_{xx} &= 2\mu \frac{\partial u_x}{\partial x} + \lambda \left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{u_y}{y} \right), \\ \tau_{yy} &= 2\mu \frac{\partial u_y}{\partial y} + \lambda \left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{u_y}{y} \right), \\ \tau_{xy} = \tau_{yx} &= \mu \left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right), \end{aligned}$$

et où le coefficient de viscosité secondaire λ est exprimé en fonction du coefficient primaire μ par l'hypothèse de Stokes $\lambda = -2/3\mu$. Les flux de chaleur visqueux sont:

$$\begin{aligned} q_x &= k_{tr} \frac{\partial T}{\partial x} + \sum_{s=\text{mol.}} k_{v_s} \frac{\partial T_{v_s}}{\partial x} + k_e \frac{\partial T_e}{\partial x} + \sum_{s=\text{all}} J_{x,s} h_s, \\ q_y &= k_{tr} \frac{\partial T}{\partial y} + \sum_{s=\text{mol.}} k_{v_s} \frac{\partial T_{v_s}}{\partial y} + k_e \frac{\partial T_e}{\partial y} + \sum_{s=\text{all}} J_{y,s} h_s, \\ q_{x,vm} &= k_{vm} \frac{\partial T_{vm}}{\partial x} + J_{x,m} h_{vm}, \\ q_{y,vm} &= k_{vm} \frac{\partial T_{vm}}{\partial y} + J_{y,m} h_{vm}, \\ q_{x,e} &= k_e \frac{\partial T_e}{\partial x} + \sum_{s=\text{all}} J_{x,s} h_{e_s}, \\ q_{y,e} &= k_e \frac{\partial T_e}{\partial y} + \sum_{s=\text{all}} J_{y,s} h_{e_s}. \end{aligned}$$

Le vecteur terme source géométrique, pour des géométries axisymétriques est:

$$Q_{\text{geom.}} = \begin{bmatrix} 0 \\ 0 \\ (p - \tau_{\theta\theta}) A_{xy}/V \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

où A_{xy} est l'aire projetée de la cellule dans le plan (x, y) et

$$\tau_{\theta\theta} = 2\mu \frac{u_y}{y} + \lambda \left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{u_y}{y} \right).$$

Pour les géométries planes $Q_{\text{geom.}}$ est un vecteur nul.

Le vecteur terme source chimique est:

$$Q_{\text{chem.}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \Omega_m^{VC} \\ \sum_{s=\text{ion.}} \Omega_s^{EC} \\ \dot{\omega}_s \end{bmatrix},$$

Le vecteur terme source d'énergie d'échange thermique est:

$$Q_{\text{therm.}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \Omega_m^{VT} + \Omega_m^{VV} + \Omega_m^{VE} \\ \sum_{s=\text{mol.}} \Omega_s^{EV} + \sum_{s=\text{all.}} \Omega_s^{ET} \\ 0 \end{bmatrix},$$

Le vecteur terme source de rayonnement est:

$$Q_{\text{rad.}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\nabla \cdot q_{\text{rad}} \\ 0 \\ -\nabla \cdot q_{\text{rad}} \\ 0 \end{bmatrix}$$

où toute composante purement vibratoire de perte de chaleur par rayonnement (ou gain) a été négligé.

A.2 Processus de résolution d'Eilmer3

1. Calcul du transport gazeux lié au flux non visqueux:
 - (A) appliquer des conditions aux limites non visqueuses ou échanger des données au niveau des limites pour chaque bloc.
 - (B) reconstituer l'état du champ d'écoulement des deux côtés de chaque interface.
 - (C) calculer le flux non visqueux $\vec{F}_i \cdot \vec{n}$.
 - (D) calculer le terme source radiatif $-\nabla \cdot \vec{q}_{rad}$ pour chaque cellule.
 - (E) intégrer l'équation (62) sur le pas de temps.
 - (F) décoder les quantités conservées par l'intermédiaire d'un appel à l'équation d'état.
 - (G) répéter pour la mise à jour correctrice.
2. Calcul du transport gazeux lié au flux visqueux:
 - (A) appliquer des conditions aux limites visqueuses aux parois.
 - (B) calculer les flux visqueux $\vec{F}_v \cdot \vec{n}$.
 - (C) intégrer l'équation (63) sur le pas de temps.
 - (D) décoder les quantités conservées par l'intermédiaire d'un appel à l'équation d'état.
 - (E) répéter pour la mise à jour correctrice.
3. Calcul du changement d'état du gaz lié aux réactions chimiques:
 - (A) calculer tous les termes de source chimiques.
 - (B) intégrer l'équation (64) sur le pas de temps.
 - (C) décoder les quantités conservées par l'intermédiaire d'un appel à l'équation d'état.
 - (D) refaire par des petits sous-cycles en cas d'échec et faire appel à l'équation d'état plus fréquemment.
4. Calcul du changement d'état du gaz lié aux échanges d'énergie thermique:
 - (A) calculer tous les termes sources chimiques.
 - (B) intégrer l'équation (65) sur le pas de temps.
 - (C) décoder les quantités conservées par l'intermédiaire d'un appel à l'équation d'état.
 - (D) refaire par des petits sous-cycles en cas d'échec et d'appliquer l'appel à l'équation d'état plus fréquemment.

A.3 Commandes et procédure pour un calcul Eilmer3 sous Bellatrix

Simulations Eilmer3 sous Bellatrix:
`$ cd /scratch/utilisateur/`

Simulation maillage grossier, répertoire 1:

- **fichier.py**
 - Définition du modèle de gaz, C.I.C.L;
 - géométrie et maillage;
 - paramètres de contrôle de simulation.
- **Echange.lua**
 - Paramètres d'échanges V-T ou T-E.
- **Reaction.lua**
 - Paramètres pour le calcul des taux de réaction.
- **Eilmer3_cas1.run**
 - Chemin du répertoire;
 - Nombre de processeurs (égal au nombre de blocs du maillage);
 - Commande de lancement du calcul sous Bellatrix.

- Préparation: `$ e3prep.py --job=fichier`
- Visualisation du maillage: `$ e3post.py --job=fichier --vtk-xml --tindx=all`
*Visualiser le fichier
généralisé sur Paraview.*
- Lancement: `$ sbatch Eilmer3_cas1.run`
- *Lors du calcul:*
 - Affichage des itérations: `$ tail -f out`
 - Modifier paramètres de contrôle: `$ nano fichier.control`
 - Affichage des solutions (tindx = temps): `$ tail -f fichier.times`
- Visualisation des solutions: `$ e3post.py --job=fichier --vtk-xml --tindx=all`

Simulation maillage raffiné, répertoire 2:

- **Fichier.py** (modifications)
 - Indiquer la reprise du calcul, à partir de la solution convergée du maillage grossier (depuis le dernier tindx dans « fichier.times »).
 - Raffiner le maillage
- **Echange.lua** & **Reaction.lua**
- **Eilmer3_cas2.run** (reprendre *Eilmer3_cas1.run*)
 - Modifier chemin du répertoire 2.

- Même procédure mais avec: `$ sbatch Eilmer3_cas2.run`

Simulation ablation, répertoire 3:

- **fichier.py** (modification)
 - Indiquer une reprise du calcul, à partir de la solution du maillage raffiné (i.e son dernier tindx relevé, dans « *fichier.times* »);
 - Changer la condition limite surfacique en paroi.
- **Echange.lua** & **Reaction.lua**
- **Condition_surfacique.lua**
 - Paramètres des taux de réaction d'injection d'espèces ablatives.
- **Eilmer3_cas3.run** (reprendre *Eilmer3_cas2.run*)
 - Modification: mettre chemin du répertoire 3

➤ Même procédure mais avec: `$ sbatch Eilmer3_cas3.run`

Simulation rayonnement Monte-Carlo, répertoire 4:

- **Fichier.py** (Modifications)
 - Ajouter la ligne de chargement d'un modèle de rayonnement;
 - Indiquer que la reprise du calcul se fait à partir d'une solution convergée, pour un maillage raffiné choisi (à partir de son dernier tindx relevé, dans « *fichier.times* »).
- **Echange.lua** & **Reaction.lua**
- **Eilmer3_cas4.run** (reprendre *Eilmer3_cas3.run*):
 - Modification: mettre chemin du répertoire 4.
 - Si le calcul est repris (calcul couplé), le tindx de la commande doit être changé et doit correspondre au dernier tindx relevé dans « *fichier.times* ».
- **Rad_Script.py**:
 - Configurer un modèle de rayonnement.
- **e3rad.run**
 - Mettre chemin du répertoire;
 - Paramètres Bellatrix (mémoire, processeurs...) adaptés;
 - Commande de lancement de calcul rayonnement
 - Si le calcul est repris (calcul couplé), le tindx de la commande doit être changé et doit correspondre au dernier tindx relevé dans « *fichier.times* ».

➤ Création de rad-model.lua: `$ rad_model='Rad_Script.py'`
`$ radmodel.py -i $rad_model -L rad-model.lua`

➤ Préparation: `$ e3prep-.py --job=fichier`

Procédure calcul couplé:

- 1^{er} calcul rayonnement: `$ e3rad.run`
- Une fois le calcul terminé, réduire le « time_max »: `$ nano fichier.control`
(le calcul suivant sera effectué jusqu'à ce « time_max »)
- 1^{er} calcul (reprise): `$ sbatch Eilmer3_cas4.run`
- 2nd calcul rayonnement (reprise): `$ e3rad.run`
Une fois le calcul terminé, prolonger le « time_max »: `$ nano fichier.control`
- 2nd calcul (reprise): `$ sbatch Eilmer3_cas4.run`
- Continuer la procédure de calcul couplé jusqu'à convergence

Le fichier python principal, *fichier.py*:

1. Configuration du modèle de gaz :
 - a. Indiquer le modèle de gaz.
 - b. Indiquer la liste des espèces (ce n'est pas parce qu'elles sont présentes dans cette liste qu'elles seront présentes dans les résultats. tout dépend des conditions initiales et fichiers de réactions).
 - c. Indiquer le nom du fichier d'échanges (*nom_fichier.lua*).
 - d. Indiquer le nom du fichier de réaction (*nom_fichier.lua*).
2. Définir les conditions de fluide :
 - a. Indiquer les conditions et pourcentages d'espèces en amont.
 - b. Indiquer les conditions initiales dans le domaine (soit à partir des conditions d'entrée, soit à partir d'une solution déjà existante).
3. Géométrie et maillage :
 - a. Définir les noeuds et créer un domaine.
 - b. Créer des sous-surfaces en reliant les noeuds avec des fonctions.
Conseil: Ne pas utiliser de courbe de Bezier proche de la ligne d'arrêt dans la couche de choc, seulement des arcs.
4. Conditions limite, discrétisation et blocs
 - a. Les vecteur utilisés sont de la forme: vecteur = [nord, ouest, sud, est]. Pour chaque sous surface créer un vecteur:
 - i. *cf_list* c: composé de fonctions de répartitions afin de pouvoir discrétiser chaque bords. Cela permet de créer des raffinements.
 - ii. *bc_list*: composé des conditions limites sur chaque bord (*ExtrapolateOutBC()* pour la sortie, *AdjacentBC()* si la surface est collée à une autre surface, *SupInBC(inflow)* pour l'entrée, *SlipWallBC()* pour une condition de glissement dans le cas d'une symétrie, *FixedTBC()* pour un mur à une température fixée, *UserDefinedMassFluxBC("nom_fichier.lua")* pour une conditions surfacique d'ablation.
 - b. Définir chaque super blocs avec:
 - i. Une sous surface.
 - ii. Une *cf_list*.
 - iii. Une *bc_list*.
 - iv. Une condition initiale de domaine (voir le point 2).
 - v. Le nombre de cellules en x et en y (qui doit correspondre au nombres de cellules des autres blocs aux frontières).
 - vi. Le nombre de blocs en x et y (les super blocs sont divisés en blocs, le nombre total de blocs dans le domaine doit être égal au nombre de processeurs lors du calcul MPI).
 - c. Rajouter *identify_block_connections()* à la suite pour relier les super blocs.
5. Paramètres de simulation
 - a. Mettre le Flag=1 si activation de la viscosité ou de la diffusion dans la simulation.
 - b. Choisir le modèle de diffusion si présent.
 - c. Indiquer le critère de fin de simulation (soit *gdata.max_step*, soit *gdata.max_time*).
 - d. Indiquer l'intervalle de temps pour un relevé de solution (*gdata.dt_plot* = temps maximum divisé par le nombre choisi de solutions relevées)
 - e. Fixer le cfl (*gdata.cfl*)

A.4 Fichiers matlab

```
% -----  
% Script MatLab  
% -----  
%Ce script permet de tracer  
% Ensemble les modes de tempÈratures  
% Les Flux de chaleur (ensemble ou pas)  
% Les fractions molaires pour les molecules non chargÈes et ions  
  
%Les fichiers de donnees lus par ce script doivent provenir des commandes Eilmer3 suivantes:  
%1. e3post.py--job=x2 --output-file=PF60 --add-molef --slice-list=...  
%2. e3post.py --job=x2 --output-file=HF60 --heat-flux-list=...  
  
close all;  
clear all;  
  
% Sous dossier (si le fichier "l'output-file" est dans un repertoire un cran en dessous dans  
l'arborescence):  
%addpath(genpath('./')) %nom du repertoire  
  
% Nom de la figure  
schar='...';  
  
% Plot proprietes  
% Choisir: 1 actif - 0 non actif  
% {Profile, Heatflux, MF molecules, HF ions}  
prop = [1,1,0,0]; % Activer certaines proprietes pour les plots.  
savefig = 1.0; % Activer sauvegarde figure.  
traiton = 1.0; % TracÈ pointillÈs  
  
limxw = -0.0016; % Limite en x + l'ouest.  
loc = ['southwest';'southwest';'southwest';'northwest']; %Positions legendes sur chaque figure  
  
molaxis_mol = [-0.0016 inf 10e-13 1]; %axes mole fractions abla  
molaxis_ion = [-0.0016 inf 10e-6 10e-2]; %axes mole fractions abla  
  
%Nom donne a l'output-file:  
PFon = 1;  
filename1 = 'PF60';  
% Tracer energie + la place de tempÈrature? 1 = oui, 0 = non.  
pf_e = 0;  
  
% Traiter les flux de chaleur? 1 = oui, 0 = non.  
HFon = 0;  
filename2 = 'HF60'; % Valable si HFon = 1;  
% Le(s) flux souhaite(s)?  
HFcon = 1;  
HFdon = 0;  
HFron = 0;  
  
% Ne plus remplir -----  
% Chargement HF  
if HFon == 1  
    delimiterIn = ' ';  
    headerlinesIn = 17;  
    HF = importdata(filename2,delimiterIn,headerlinesIn);  
  
    %Vecteurs pour HF  
    y = HF.data(:,1);  
    HFc = HF.data(:,2);  
    HFd = HF.data(:,3);  
    HFr = HF.data(:,4);  
end  
  
if PFon == 1  
% Chargement P  
    delimiterIn = ' ';  
    headerlinesIn = 1;  
    P = importdata(filename1,delimiterIn,headerlinesIn);  
  
% Recuperation et traitement des labels des donnees, definition des indices  
    Ptext = strsplit(P.textdata{1,:});  
    ChPtext = char(Ptext);
```

```

for i=3:length(ChPtext) %partir de 3 car les 2 premieres colones sont inutiles.
    if pf_e == 1
        A = findstr(ChPtext(i,:), 'e[0]');
    else
        A = findstr(ChPtext(i,:), 'T[0]');
    end
    TF = isempty(A);
    if TF == 0
        indxT0 = i-2;
        indxT1 = i;
    end

    B = findstr(ChPtext(i,:), 'molef[0]');
    TF2 = isempty(B);
    if TF2 == 0
        indxMf0 = i-2;
    end
end

% Vecteurs pour P
x = P.data(:,1);
if x<=0
    x = x + abs(max(x));
end
T0 = P.data(:,indxT0);
T1 = P.data(:,indxT1);

% Vecteurs pour les fractions molaires
mflabels = ChPtext(indxMf0+2:end,9:end);
mf = P.data(:,indxMf0:end);
for j=1:size(mflabels,1)
    C(j,:) = strsplit(mflabels(j,:), '-');
    C1(j,:) = C(j,2);
end

C1 = regexprep(C1, '_plus', '^+');
C1 = regexprep(C1, '_minus', '^~');
C1 = char(C1);

% Separe les ions
mf_mollabel = {};
mf_ionlabel = {};
mf_ion = {};
mf_m = {};
for k = 1:size(C1,1)
    if isempty(findstr(C1(k,:), '+')) == 0 | isempty(findstr(C1(k,:), '^~')) == 0 %c'est un ion
        %Super important!! construction de vecteur
        mf_ionlabel = [mf_ionlabel, C1(k,:)];
        mf_ion = [mf_ion,mf(:,k)];
    else
        mf_mollabel = [mf_mollabel,C1(k,:)];
        mf_m = [mf_m,mf(:,k)];
    end
end
mf_mollabel = char(mf_mollabel);
mf_ionlabel = char(mf_ionlabel);

%-----
% Fait la liste des espèces manquantes et supprime les espèces pour lesquelles la valeur = 0
l1=0;
disp('For fig(3) plot:')
for l = 1:size(mf_m,2)
    if mf_m(l,1)==0
        disp(strcat('missing: ',mf_mollabel(l,:)));
        l1=[l1,l];
    end
end
l1(1)=[];
mf_mollabel(l1,:)=[];
mf_m(:,l1)=[];

l2=0;
disp('For fig(4) plot:')

```

```

for m = 1:size(mf_ion,2)
if mf_ion{1,m}==0
disp(strcat('missing: ',mf_ionlabel(m,:)));
l2=[l2,m];
end
end
l2(1)=[];
mf_ionlabel(l2,:)=[];
mf_ion(:,l2)=[];
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PLOTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Temperature %%%

if PFon == 1
% Propriete figure
fig = figure(1);
colordef(fig,'none');
whitebg(fig,'w');
if prop(1) == 1
orient portrait;
set(fig,'papertype','a4letter');
set(fig,'units','centimeters','paperunits','centimeters');
set(fig,'paperposition',[0.5 10. 15. 15.]);
set(fig,'position',[0.5 10. 15. 15.]);
axes('units','centimeters','position',[3. 2. 10. 10.]);
end

% Les plots
plot(x,T0,'b','linewidth',2,'markersize',10)
hold on
plot(x,T1,'r','linewidth',2,'markersize',10)

% Les labels,limites d'axes, legendes
axis([limxw inf -inf inf])
xlabel('x(m)','FontSize',10)
if pf_e == 1
ylabel('e(J)','FontSize',10) %,'rot',0)
legend('e-tr','e-vee','Location',loc(1,:))
e = '_E_';
else
ylabel('T(K)','FontSize',10) %,'rot',0)
legend('Ttr','Tvee','Location',loc(1,:))
e = '';
end
grid on
%title('Temperature modes upstream')

% Sauvegarde
if savefig == 1
formatspec = strcat(schar,e, 'PF.png');
print(gcf,formatspec,'-dpng','-r300');
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Heat flux %%%

if HFon == 1
% Propriete figure
fig = figure(2);
colordef(fig,'none');
whitebg(fig,'w');
if prop(2) == 1
orient portrait;
set(fig,'papertype','a4letter');
set(fig,'units','centimeters','paperunits','centimeters');
set(fig,'paperposition',[0.5 10. 15. 15.]);
set(fig,'position',[0.5 10. 15. 15.]);
axes('units','centimeters','position',[3. 2. 10. 10.]);
end

% Les plots

```

```

leg = {'HFc','HFd','HFf'}; %legende
hold on
if HFcon == 1
    plot(y,HFc,'b','linewidth',2,'markersize',10)
else
    leg{1} = 'null';
end
if HFdon == 1
    plot(y,abs(HFd),'r','linewidth',2,'markersize',10)
else
    leg{2} = 'null';
end
if HFron == 1
    plot(y,HFr,'m','linewidth',2,'markersize',10)
else
    leg{3} = 'null';
end
leg(ismember(leg,'null'))=[];
if size(leg,2) == 1
    y_HF_label = char(leg);
else
    y_HF_label = 'HF';
end

% Les labels,limites d'axes, legendes
xlabel('y(mm)','FontSize',10)
ylabel({y_HF_label,'(W/m)'},'FontSize',10) %12,'rot',0)
axis tight
if length(leg) ~= 1
    legend(leg,'Location',loc(2,:))
end
grid on
box on
if savefig == 1
    formatspec = strcat(schar,'HF.png');
    print(gcf,formatspec,'-dpng','-r300');
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%0Ω
%%% Mole fractions %%%

if PFon == 1
% Propriete figure
fig = figure(3);
colordef(fig,'none');
whitebg(fig,'w');
if prop(3) == 1
    orient portrait;
    set(fig,'papertype','a4letter');
    set(fig,'units','centimeters','paperunits','centimeters');
    set(fig,'paperposition',[0.5 10. 15. 15.]);
    set(fig,'position',[0.5 10. 15. 15.]);
    axes('units','centimeters','position',[3. 2. 10. 10.]);
end

% Les plots
for row = 1:size(mf_m,2)
    if traiton == 1
        if size(mf_m,2)<=6
            if row > size(mf_m,2)/2
                trait = '-';
            else
                trait = ':';
            end
        else
            if row <= size(mf_m,2)/3
                trait = '-';
            elseif row > size(mf_m,2)/3 && row <= 2*size(mf_m,2)/3
                trait = '-.';
            else
                trait = ':';
            end
        end
    end
end

```

```

        end
    else
        trait = '-';
    end
    semilogy(x,mf_m{1,row},trait,'color',[rand(1), rand(1),
rand(1)],'linewidth',2,'markersize',10)
    hold all
end

% Les labels,limites d'axes, legendes
ylabel({'fraction molaire (ND)'},'FontSize',10)
xlabel({'x(m)'},'FontSize',10) %12,'rot',0)
axis(molaxis_mol)
legend(mf_mollabel,'Location',loc(3,:))
%grid on

% Sauvegarde
if savefig == 1
    formatspec = strcat(schar, 'MFm.png');
    print(gcf,formatspec,'-dpng','-r300');
end

%-----
% Proprietes plot
fig = figure(4);
colordef(fig,'none');
whitebg(fig,'w');
if prop(4) == 1
    orient portrait;
    set(fig,'papertype','a4letter');
    set(fig,'units','centimeters','paperunits','centimeters');
    set(fig,'paperposition',[0.5 10. 15. 15.]);
    set(fig,'position',[0.5 10. 15. 15.]);
    axes('units','centimeters','position',[3. 2. 10. 10.]);
end

% Les plots
for row = 1:size(mf_ion,2)
    if traiton == 1
        if row > size(mf_ion,2)/2
            trait = '-';
        else
            trait = ':';
        end
    else
        trait = '-';
    end
    semilogy(x,mf_ion{1,row},trait,'color',[rand(1), rand(1),
rand(1)],'linewidth',2,'markersize',10)
    hold all
end

% Les labels,limites d'axes, legendes
ylabel({'fraction molaire (ND)'},'FontSize',10)
xlabel({'x(m)'},'FontSize',10) %12,'rot',0)
axis(molaxis_ion)
legend(mf_ionlabel,'Location',loc(4,:))
%grid on

% Sauvegarde
if savefig == 1
    formatspec = strcat(schar, 'MFi.png');
    print(gcf,formatspec,'-dpng','-r300');
end
end

```

```

% -----
% Script MatLab Compare_data
% -----
%Ce script permet de comparer:
% A. nb fichiers de données issus de la commande:
%   1. e3post.py--job=x2 --output-file=PF --add-molef --slice-list=...
%   2. e3post.py --job=x2 --output-file=HF --heat-flux-list=...
% B. des fichiers output contenant les profils CFD++

%Les fichiers a comparer doivent être du même type:
%1. Copier les fichiers à comparer dans le répertoire courant.
%2: Définir les paramètres du script, indiquer le nom des fichiers a comparer.
%3. Run

close all;
clear all;

% -----
%%% Paramètres script %%%

%Propriétés (1 actif - 0 non actif)
prop = 1.0;           %Faire un plot dans une fenêtre carrée.
savefig = 0.0;       %Activer sauvegarde figure.
PFon = 1;            %Activer la comparaison des profils (fichiers PF nécessaires).
HFon = 0;           %Activer la comparaison des flux de chaleur (fichiers HF nécessaires).

%Axes pour le plot
limxw = -0.0015; %-inf; %           %Limite en x a l'ouest
loc = ['northeast','northeast']; %Positions légendes sur chaque figure

%%%%%%%% Valable si PFon = 1:
% Sélectionner profil a comparer (data):
% (pour cela run matlab une fois et taper en commande: ChPtext, pour voir les données
disponibles)
% (si ca ne marche pas du premier coup regarder le nom de la donnée dans le fichier PF)
% Rentrer l'expression exacte donnée par ChPtext

% Donnée(s):
%data = 'e[0]';
%data = 'T[0]';
data = '9:p';

% Remplir si une deuxième donnée doit être tracée (mettre data2 en commentaire sinon)
%data2 = 'e[1]';
%data2 = 'T[1]';

%Le label de(s) donnée(s) en question
% ylab1 = 'Energies (J) ';
ylab1 = 'Pression (Pa) ';
%ylab1 = 'Températures (K)';

%marker (rajouter si plus de 4 cas)
marker={'^','o','s','x'};

% legendes custom PF
%custom_leg={'T tr -Eilmer3','T ve -Eilmer3','T tr -CFD++','T ve -CFD++'};
%custom_leg={'T tr -a ','T ve -a ','T tr ','T ve'};
custom_leg={'P -a ','P'};
%custom_leg={'...','...','...','...'}

% legendes pour HF (# dÉfinir dans tout les cas)
%custom_legHF = {'avec ablation','sans ablation'};
%custom_legHF = {'FCd-a','FCd','FCd-a-r','FCd-r'};
custom_legHF = {'FCc-a','FCd-a','FCc','FCd'};

%%%%%%%% Valable si HFon = 1:
% Choisir le(s) flux de chaleur(s) souhaite:
% (la légende personnalisée doit correspondre)
active_HFc = 1.0;
active_HFd = 1.0;
active_HFr = 0.0;

```

```

##### Entrer informations #####
% Mettre le nom des fichiers présents dans le répertoire (issus de e3post.py)
% Toujours commencer par PFn1 ou/et HFn1 et ainsi de suite...
% Cas à comparer:

%Calcul ablation et sans ablation:
PFn1 = 'PF_20raf_abla'; % Fichier PF numéro 1
HFn1 = 'HF_20raf_abla'; % Fichier HF numéro 1
%vs
PFn2 = 'PF20raf'; % Fichier PF numéro 2
HFn2 = 'HF20raf'; % Fichier HF numéro 2

%vs (rajouter si plus)

% -----
% Comparaison avec fichiers CFD++ (HFon doit être = 0 obligé)
% (valable si fichiers data présents dans le répertoire courant).
PFn3 = 'BC2s05stp3e4.txt';
cfhpp_colone1 = 4; % P
% cfdpp_colone2 = 5; %Tvr
% cfdpp_colone2 = 12; %Tve

%vs...

#####
% Traitement des données (ne plus remplir)

%Compte le nombre de variables PFn et HFn dans le workspace
nb=sum(strncmpi(who, 'PFn', 3));
if HFon == 1 && PFon == 1
    nb = min([nb sum(strncmpi(who, 'HFn', 3))]);
elseif HFon == 1 && PFon ~= 1
    nb=sum(strncmpi(who, 'HFn', 3));
end

for k= 1:nb
%Chargement
if HFon == 1
    HF_name = eval(sprintf('HFn%i',k));
    HF = importdata(HF_name, ' ',17);
end
if PFon == 1
    PF_name = eval(sprintf('PFn%i',k));
    P = importdata(PF_name, ' ',1);
end

% Recuperation et traitement des labels des données
if PFon == 1
%Si fichier CFD++
if strcmp('XYZPTUVWRMCPU_Number',strcat(P.textdata{1,:})) == 1
    cfdpp_file = 1;
    x = P.data(:,1);
    x = x + abs(max(x));
    D0 = P.data(:,cfdpp_colone1);
    if exist('cfdpp_colone2') == 1
        D02 = P.data(:,cfdpp_colone2);
    end
else %si fichier Eilmer3
Ptext = strsplit(P.textdata{1,:});
ChPtext = char(Ptext);
% Retrouver la colonne souhaitee
for i=3:length(ChPtext) %partir de 3 car les 2 premières colonnes sont inutiles.
    A = findstr(ChPtext(i,:),data);
    TF = isempty(A);
    if TF == 0
        indx0 = i-2;
    end
    if exist('data2') == 1
        A2 = findstr(ChPtext(i,:),data2);

```

```

        TF2 = isempty(A2);
        if TF2 == 0
            indx02 = i-2;
        end
    end
end

%Vecteurs pour P
x = P.data(:,1);
if x <=0
    x = x + abs(max(x));
    D0 = P.data(:,indx0);
    if exist('data2') == 1
        D02 = P.data(:,indx02);
    end
end
end

%Vecteurs pour HF
if HFon == 1
    y = HF.data(:,1);
    if active_HFc == 1.0
        HFc = HF.data(:,2);
    end
    if active_HFd == 1.0
        HFd = abs(HF.data(:,3));
    end
    if active_HFr == 1.0
        HFr = HF.data(:,4);
    end
end
end

%-----
%***** Les plots *****

if PFon == 1
%Le plot 1: Data
    fig=figure(1);
    % Si la donnee a comparer est une fraction molaire alors le trace sera logarithmique
    if strfind(data,'molef') == 1
        semilogy(x,D0)
        hold on

        % Sinon le trace reste normal
    else
        color = [rand(1),rand(1),rand(1)];
        % Gestion des lignes
        if exist('data2') ~= 1
            line='-';
        end
        if exist('data2') == 1
            line='-';
            line2=': ';
        end
        end

        plot(x,D0,line,'color',color,'linewidth',1,'marker',marker{1,k})
        %plot(x,D0,'-', 'color',color,'linewidth',2) %(tracé personnalisé)
        hold on
        if exist('D02') == 1
            plot(x,D02,line2,'color',color,'linewidth',1,'marker',marker{1,k})
            %plot(x,D02,'-', 'color',[rand(1),rand(1),rand(1)], 'linewidth',2)
            %plot(x,D02,'-', 'color',[rand(1),rand(1),rand(1)], 'linewidth',1, 'marker',marker{1,2})
        end
    end

    if k == nb
        %Les labels,limites d'axes
        xlabel('x(m)', 'FontSize',10)
        ylabel(ylabel, 'FontSize',10) %,'rot',0)
        axis([limxw inf -inf inf])
    end
end

```



```

    if prop == 1
        axis square
    end

    %legendes
    legend(custom_leg, 'location', loc(1,:))
    grid on
    box on

    %sauvegarde
    if savefig == 1.0 && k==nb
        formatspec = strcat('PF_', 'compare_', data, '.png');
        if exist('data2')
            formatspec = strcat('PF_', 'compare_', ylab1, '.png');
        end
        print(gcf, formatspec, '-dpng', '-r300');
    end
end
end

if HFon == 1
    %Le plot 2: Flux de chaleur
    figure(2);
    color = [rand(1), rand(1), rand(1)];

    hold all
    if active_HFc == 1.0
        plot(y, HFc, '-', 'color', color, 'linewidth', 2)
        flux_de_chaleur = 'Flux de chaleur q_{cond}';
    end

    if active_HFd == 1.0
        plot(y, HFd, ':', 'color', color, 'linewidth', 2)
        flux_de_chaleur = 'Flux de chaleur q_{diff}';
    end

    if active_HFr == 1.0
        plot(y, HFr, '-.', 'color', color, 'linewidth', 2)
        flux_de_chaleur = 'Flux de chaleur q_{rad}';
    end

    %sauvegarde
    if k==nb
        %Les labels, limites d'axes, legendes
        xlabel('y(mm)', 'FontSize', 10)
        ylabel({flux_de_chaleur, '(W/m^2)'}, 'FontSize', 10) %12, 'rot', 0)
        axis tight
        if prop == 1
            axis square
        end
        grid on
        box on
        legend(custom_legHF, 'location', loc(2,:))
        if savefig == 1.0 && k==nb
            formatspec = strcat('HF_', 'compare_', '.png');
            print(gcf, formatspec, '-dpng', '-r300');
        end
    end
end
end
end

```

```

% -----
% Script MatLab Résidus contrôle
% -----

close all;
clear all;

% Ce script permet le contrôle des résidus
%
% Etape 0:
% Copier get_residuals.py dans le dossier ou tourne la simulation
% (commande: scp get_residuals.py host@bellatrix.epfl.ch:/scratch/(chemin
% du fichier)
%
% Etape 1:
% Dans le dossier ou tourne la simulation générer le fichier résidus:
% (commande: python get_residuals.py global res.txt)
%
% Etape 2:
% Coller le fichier res.txt et (job).times dans le répertoire ou se trouve ce script
% (commande: scp host@bellatrix.epfl.ch:/scratch/(chemin du fichier)/x2.times .)
% (commande: scp host@bellatrix.epfl.ch:/scratch/(chemin du fichier)/res.txt .)

%Configuration:
%-----
% Si le contrôle update est active, relever le step quand les paramètres de
% contrôle sont changés. Exemple: CFL de 0.05, facteur visqueux de 1, jusqu'au pas 81340 etc..

%char:ctrl update -Step-
%cfl 0.05, vf:1.0 81340
%....
%cfl 0.20, vf:0.01 (laisser vide)

Activer_ctrl_pt = 1.0; %pour activer le trace des points ou job.control a été changé
%job.ctrl update at step:
if Activer_ctrl_pt == 1.0
    max_old_step= [81340]; %[1ststep 2ndstep 3rdStep] Maximum 3 valeurs
    char={'cfl 0.05, vf:1.0' 'cfl 0.20, vf:0.01'}; % {' ' ' ' ...}
    SUB = 1.0; %activer le subplot (0 = non, 1 = oui)
end

% -----
% Sauvegarde du plot?:
savefig = 0; schar='nom';

%Propriétés légende
loc='northeast'; lgd = {'Densité','Energie','Solution'};
%lgd = {'Densité', 'Energie', 'Solution', 'Modification job.control'};

% Ne plus modifier -----
%Chargement
filename1 = 'res.txt';
delimiterIn = ' ';
headerlinesIn1 = 6;
RES = importdata(filename1,delimiterIn,headerlinesIn1);

filename2 = 'x2.times';
delimiterIn = ' ';
headerlinesIn2 = 1;
T = importdata(filename2,delimiterIn,headerlinesIn2);

% Vecteurs pour RES
step = RES.data(:,1);t = RES.data(:,2); resd = RES.data(:,3); rese = RES.data(:,4);
% Vecteurs pour les points de solution
tindx = T.data(:,1); t_sol = T.data(:,2); dt = T.data(:,3);

%vérification de RES.step
vrf = find(step == min(step));
if length(vrf)>1 && length(vrf)<=2
    pb = (find(step == step(vrf(find(vrf > 1))-1)));
    for i=1:length(pb)
        if (step(pb(i) +1)== min(step))
            for j=(pb(i)+1):(length(step))

```

```

        step(j) = step(j) + step(pb(i));
    end
end
end
elseif length(vrf)==1
    %tout est en ordre
else
    %il y a plus de 2 coupures
end

cible = zeros(1,length(t_sol)-1);
for k=2:length(t_sol)
[cible(k-1)] = t(max(find(t <= t_sol(k)))); %valeur la plus proche de t juste avant t_sol
end
cible=cible.';

step_sol = zeros(1,length(t_sol)-1);
for k=1:length(t_sol)-1
[step_sol(k)] = step(find(t == cible(k))); %valeur la plus proche de t juste avant t_sol
end
step_sol=step_sol.';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fig = figure(1); colordef(fig,'none'); whitebg(fig,'w');

%Les plots
semilogy(step,resd,'b')%,'linewidth',2,'markersize',10)
hold on
semilogy(step,rese,'r')%,'linewidth',2,'markersize',10)
cst(1:length(step_sol))=1e-1;
plot(step_sol,cst,'r o')

if Activer_ctrl_pt == 1.0
    plot(max_old_step,1e-1,'b o')
end
%Les labels, limites d'axes, légendes
xlabel('step','FontSize',10); ylabel('Residual(ND)','FontSize',10)
legend(lgd,'location',loc),grid on
if savefig == 1.0
    formatsec = strcat(schar, '_PlotRES.png'); print(gcf,formatsec,'-dpng','-r300');
end

if SUB == 1.0
    if length(max_old_step)<=4
        r=2;
    else
        r=3;
    end
    for q = 1:(length(max_old_step))+1
        figure(2)
        subplot (2,r,q)
        if q == 1
            dots = 1;
            dotf = find(step == max_old_step(1));
        elseif q == (length(max_old_step))+1
            dots = find(step == max_old_step(q-1));
            dotf = find(step == max(step));
        else
            dots = find(step == max_old_step(q-1));
            dotf = find(step == max_old_step(q));
        end
        semilogy(step(dots:dotf),resd(dots:dotf),'b')%,'linewidth',2,'markersize',10)
        hold on
        semilogy(step(dots:dotf),rese(dots:dotf),'r')%,'linewidth',2,'markersize',10)
        cst(1:length(step_sol))=1e-1;
        %Les labels, limites d'axes, légendes
        xlabel('step','FontSize',10); ylabel('Residus (ND)','FontSize',10); axis tight;
        legend('Density','Energy','location',loc),grid on
        if savefig == 1.0
            formatsec = strcat(schar, '_SubPlotRES.png'); print(gcf,formatsec,'-dpng','-r300');
        end
    end
end
end
end

```

```

% -----
% Script MatLab: Compare les spectres obtenus par la methode plan-tangents
%                 d'Eilmer3. Certaines fonctionnalités peuvent encore être développées.
% -----
close all; clear all;
%Description
% a. Ce script permet de tracer les données de 1 ou plusieurs fichiers Tangent-slab
%     issus de la commande: e3post.py --job=x2 --output-file=TS --tangent-slab-list=...
%     La commande crée un fichier TS et un fichier exact-TS-flux-spectra.data
% b. Il permet de comparer les spectres avec ceux du NIST, et d'identifier les raies.
%     Pour une comparaison avec le NIST, rajouter la fonction NIST_spectra plus
%     un éventuel fichier d'espèces pour les molécules dans le répertoire courant

%Méthode
% a. Copier ces fichiers dans le répertoire courant (où se situe ce script),
%     avec des noms différents pour chaque cas (Exemple: TS1 et TS2).
% b. Remplir la partie paramètres du script
% c. Run

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Parametres script %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fichiers présents dans le répertoire (issus de e3post.py):
nb = 2; % Nombre maximum de cas à comparer

% Cas à comparer:
% ! Toujours commencer par file11 ou/et file12 et ainsi de suite...
% ! Commenter les fichiers selon l'envie de tracer ou non
% exemple: mettre % devant file11, si seul les spectres sont demandés

file11 = 'TS'; %Fichier TS
file12 = 'exact-TS-flux-spectra.data'; %Fichier exact-TS...
%vs
file21 = 'TS_20raf'; %vs
file22 = 'exact-TS-20raf.data';
%vs
%file31 =
%file32 =
%vs (rajouter si plus)

% -----
% Propriétés des plots
% Coupure à gauche
limxw = -1.5e-3;
%facteur de hauteur par rapport au plus haut pic (plot spectres)
%(Ne s'applique qu'au dernier plot tracé... (à développer))
hfact = 0.3;
% Plages de valeurs en nm (Intervalles souhaités):
lambda_range={[80 220], [200 460],[610 680],[740 800]};
%lambda_range={[300 450]};
%lambda_range={[-inf inf]};
%Propriétés figure (1 actif - 0 non actif)
prop = 1.0; %Faire un plot dans une fenêtre carrée.
savefig = 0.0; %Activer sauvegarde figure.
% Légende personnalisée:
custom_leg = {'P-tgt: plasma -a', 'P-tgt: plasma'};
% Lignes personnalisées:
linestyles = {'-'; ':'};
% Position de la légende
loc=['northwest'; 'northeast'];
% Couleurs personnalisées:
%rose: [0.9730 0.6490 0.8003]
%aléatoire: [rand(1) rand(1) rand(1)]
%vert clair: 0.7252 0.8343 0.0188
colors = {[0.7252 0.8343 0.0188]; 'k'};

% ----- Comparaison avec les données du NIST -----
% cette partie permet d'identifier les raies
Compare_to_NIST = 1.0;
NIST_color = 'r'; % Couleur du plot
%Chosir soit une atome soit une molécule: choix = 1 l'autre 0.
a_atm = 1; a_molecule = 0;

```

```

%marche si a_atm = 1:
    atom = 'carbon'; % Nom de l'atome (nom en lettres, en anglais)
    %scale_f=50; %Facteur d'ajustement d'intensité (ne marche que pour les atomes)
%marche si a_molecule = 1:
    %Regarder la fonction NIST_spectra pour plus d'infos sur le fichier
    %d'espèces, qu'il faut ajouter au répertoire courant
    s = 4; %ligne de l'espèce choisie dans le fichier

% -----
% ----- Annotations (ne fonctionne pas bien, voir dans le script pourquoi) -----
% Annotations (fichier_externe.txt nécessaire) avec dedans:
% Première ligne, les annotations (exemple les espèces: O N C ...)
% Lignes suivantes, colonnes dans l'ordre: colone1 pour O, colone2 pour N ...

% Chaque mots du fichier sont séparés par un délimiter de type?
    delimiter = ' ';
%Chargement fichier externe
    identified_NIST_lines = 'RaiesNist.txt';

% -----
% Ne plus remplir % Ne plus remplir %
for k= 1:nb
%Chargement
if exist(sprintf('file%i1',k)) == 1
    TS = importdata(eval(sprintf('file%i1',k)), ' ',4);
    %Vecteurs pour TS (absorption emission)
    s = TS.data(:,1);
    s = s - abs(s(end));
    divq1 = TS.data(:,3);
end
if exist(sprintf('file%i2',k)) == 1
    ExS = importdata(eval(sprintf('file%i2',k)), ' ',4);
    %Vecteurs pour ExS (spectre)
    Wv1 = ExS.data(:,1);
    q_lambda = ExS.data(:,2);
    %conversion W/m2 to MW/cm2
    q_lambda = q_lambda.*1e-10*1000000*1e-9;
end

% -----
% Plots %
%Le plot de la divergence du flux de rayonnement
if exist(sprintf('file%i1',k)) == 1
    fig=figure(1);
    plot(s,divq1, 'linestyle',linestyles{k}, 'color',colors{k}, 'linewidth',2, 'markersize',10)
    hold on
    if k==nb
        %Les labels, limites d'axes, légendes
        xlabel('x (m)', 'FontSize',10)
        ylabel('\nabla.q_{rad} (W/m^2)', 'FontSize',10) %,'rot',0)
        axis tight;
        axis ([limxw inf -inf inf])
        grid on; box on;
        legend(custom_leg, 'location', loc(1,:))
        if prop == 1
            axis square
        end
        if savefig == 1.0
            formatspec = strcat('Raddiv_compare.png'); print(gcf,formatspec, '-dpng', '-r300');
        end
    end
end

% -----
%Le plot des spectres
if exist(sprintf('file%i2',k)) == 1 %Existance de fichiers spectres
    nb_fig=length(lambda_range);
    for i=1:nb_fig
        figure(i+1);
        h(k) = plot(Wv1,q_lambda, 'linestyle',linestyles{k}, 'color',colors{k});
        hold on
        %Labels, axis limits
        xlabel('Longueur d''onde (nm)', 'FontSize',10)
    end
end

```

```

ylabel('q_lambda (W/cm^2 . nm)', 'FontSize', 10);
axis tight;

%Trouve le pic maximum et rajoute un tiers (à développer)
num = [lambda_range{1,i}];
axis_up = max(q_lambda(find(Wv1 > num(1,1) & Wv1 < num(1,2))));
axis_up = axis_up+(axis_up*(hfact));
axis([lambda_range{1,i} -inf axis_up]);
box on; %grid on

if k==nb % Opérations de dernière itération
%----- Comparaison NIST -----
if Compare_to_NIST == 1
% NIST range
NIST_range = [min(cell2mat(lambda_range)) max(cell2mat(lambda_range))];
% Get spectra from NIST
[lambda, y] = NIST_spectra(NIST_range, atom, s, a_atm, a_molecule);
if a_atm == 1
if exist('scale_f')
%rescale if a_atm=1
y=y./scale_f;
else
y(1:length(y)) = max(q_lambda)/10;
end
elseif a_molecule == 1
y(1:length(y)) = max(q_lambda)/10;
end
hold on
%Plot data
for j=1:length(lambda)
hNIST = plot([lambda(j) lambda(j)], [0 y(j)], 'color', NIST_color);
% Détection de pics automatique (à développer)
% idée: faire un petit intervalle autour de chaque pic
% du NIST, puis trouver dans cette intervalle la valeur
% max du plot obtenu par simulation. Si cette valeur
% dépasse un certain seuil alors c'est vraiment un pic.
end
end
% Legende NIST (ne s'affiche pas, à développer)
if Compare_to_NIST == 1
if a_atm == 1
% same color group of lines
legend([h hNIST], {custom_leg{1,1}, custom_leg{1,2}},
strcat('NIST:', atom), 'location', loc(2,:));
end
end
%----- Fin NIST -----
% Legendes
if Compare_to_NIST ~= 1
legend(custom_leg, 'location', loc(2,:))
end

%
% Annotations (à développer) (fichier .txt nécessaire)
% Raies_NIST = importdata(identified_NIST_lines, delimiter, 1);
% for l=1:1 %length(Raies_NIST.textdata)
% arrow = Raies_NIST.textdata{1,1};
% for m =1:length(Raies_NIST.data(:,1))
% ar_x = Raies_NIST.data(m,1);
% quiver(ar_x, max(q_lambda-20), 0, -30, 'color', 'k')
% end
% end

%Save figure
if savefig == 1.0
if Compare_to_NIST ~= 1
formatspec = strcat('SpFlx_range:', sprintf('%s', lambda_range{1,i}), '.png');
print(gcf, formatspec, '-dpng', '-r300');
end
end
end %fin de condition: dernière itération
end % fin de boucle pour toutes les plages de données
end %fin de condition existence de fichiers spectres
end % fin de boucle pour tout les fichiers a comparer

```

```

function [lambda, y] = NIST_spectra(lambda_range, atom,s, a_atm, a_molecule)
% -----
% Function MatLab
% -----
% Gets spectra from NIST
% Plots spectra (look end of file)

%To use this function as a script instead:
% 1. delete all the strings: '%input_var:'
% 2. delete first line and last line of this file
% 3. uncomment the two following words:

%close all;
%clear all;

%-----
% Start of modifiable zone by user.

%Select atom or molecule: your choice = 1 the other 0.
%input_var: a_atm = 0;
%input_var: a_molecule =1;

%Range in nm
%input_var: lambda_range=[50 1200];

% ----Atoms-----
%select atom (strong lines)
%input_var: atom = 'oxygen';

% ---- Molecules ----
%1. Go on this page
% http://www.nist.gov/pml/data/molspec.cfm

%2. On the right of the page, find the molecule you want,
% select the isotopes, tick ASCII output, and click "get data".
% Copy the link of the page that appears.

%3. Create a file "Species.txt"
%4. Write a line for each molecule (in this format: 'molecule = copied_url')
% Example (careful: put spaces on sides of the first '='):
% CO = http://...(link)...

% ----LOAD URL----
%5. Once your "Species.txt" is ready, select the row of the molecule you want.
% Type C{1} to see all molecules available (only works if a_molecule=1):

%input_var s=2; %s is the row

% End of modifiable zone.
%-----
%--BUILD URL----
if a_atm == 1
    url = strcat('http://physics.nist.gov/PhysRefData/Handbook/Tables/',atom,'table2', '_a.htm');
elseif a_molecule == 1
    fileID = fopen('Species.txt');
    C = textscan(fileID,'%s %s %s');
    fclose(fileID);
    url = char(C{3}{s});
    disp(strcat('molecule selected:', ' ',C{1}{s}))
end

% ---- html page process ----
html = webread(url);
% Use regular expressions to remove undesired markup.
txt = regexprep(html, '<script.*?/script>', '');
txt = regexprep(txt, '<style.*?/style>', '');
txt = regexprep(txt, '<.*?>', '');
txt = strsplit(txt,sprintf('\n')).';

% ---- Get the info ----
%Separe whenever atom or molecule
if a_atm == 1
    Name= atom;

```

```

Columns= txt(7);
data= char(txt(11:size(txt,1)-2));
cor = 0; %To adjust indexes if no value found
for i=1:size(data,1)
    D = strsplit(data(i,:), ' ');
    %Checks if value is in 3rd or 4th column
    if length(D) <= 3 %There won't be any information for this length
        cor = cor+1;
    elseif isnan(str2double(D(3))) == 0 %checks if 3rd column is numeric
        freq(i-cor,:) = D(3);
        Intensity(i-cor,:) = D(2);
    else
        freq(i-cor,:) = D(4);
        Intensity(i-cor,:) = D(2);
    end
end
freq = str2double((freq));

elseif a_molecule == 1
Name= txt(4);
Columns= txt(9);
data= char(txt(8:size(txt,1)-2));
for i=1:size(data,1)
    D = strsplit(data(i,:), ' ');
    freq(i,:) = D(2);
    if isnan(str2double(freq(i,:)))
        freq(i,:) = D(3); %depends on file
    end
end
freq = str2double((freq));
end

%---- Convert ----
if a_atm == 1
    % ---- Convert Angstrom to nm ----
    lambda = freq./10;
elseif a_molecule == 1
    % ---- Convert MHz to nm ----
    c = 299792458; %speed of light
    lambda = (c./freq)*1000; %conversion (nm)
end

% ---- prepare for bar plot ----
if a_atm == 1
    %Find duplicate indexes
    u=unique(lambda);
    n=histc(lambda,u);
    index=find(ismember(lambda,u(n>1)));
    if isempty(index) == 0
        display('duplicates are present')
    end
    %Prepare Intensity
    y=str2double((Intensity));
    %Delete duplicates
    lambda = unique(lambda);
elseif a_molecule == 1
    % ---- prepare for plot ----
    lambda = unique(lambda);
    y=ones(size(lambda,1),1); % For normalized plot
end
end
end

```



```

% -----
% Script MatLab
% -----
close all; clear all;
%Convertit le fichier réaction Eilmer3 en fichier réaction CFD++.
%Le nombre d'espèces identiques coté produits ou coté réactifs ne doit pas dépasser 3 par
équation de réaction.
%ATTENTION dans CFD++ la fraction massique de la dernière espèce ne doit pas être nulle:
%    Il faut modifier l'ordre dans le fichier de réaction Eilmer3:
%    1. Mettre en dernier de la liste une espèce dont la fraction massique est non nulle en
condition initiale.
%    2. Il faut aussi mettre en ordre les efficacités des espèces pour chaque réaction du
coup.
%file to convert
filename = 'Park2001_NEQ_20sp_ready2cvrt.lua';
%new file name
new_file_name = 'reaconverted.rea';
%pressure dependency?
press_dep = '0.0';

%-----
%line of species in source file
l_s = 12;
% R gaz parfait
R_gp = 8314.0;
%import
Re3 = importdata(filename);

%data
species = Re3(l_s,:);
species = regexprep(species, ' ', '');
species = regexprep(species, '--', '-');
species = regexprep(species, ', ', '');
species = regexprep(species, '\n', '');
species = regexprep(species, '_minus', '-');
species = regexprep(species, '_plus', '+');
species = char(species);
length_spc = [length(strsplit(species, '\n'))];
k=0;
prms=[''];
effs=[''];
nb_ps=[''];
nb_rs=[''];
iCs = [''];
schar = '%s? ? ? ? %s';

for i=1:size(Re3,1)
A = findstr(char(Re3(i,:)), 'reaction{'); %Reactions
B = findstr(char(Re3(i,:)), 'fr={'); %Parametres
C = findstr(char(Re3(i,:)), 'efficacities={'); %Efficacities

%Traitement des equations
if isempty(A) == 0
k = k+1; %nombre d'equations
R=char(Re3(i+1,:));
R=regexprep(R, '<=>', '<->'); R=regexprep(R, ' ', '');
R=regexprep(R, 'e-', 'e'); R=regexprep(R, ', ', ''); R = strsplit(R, {' + ', ' '});

R(1)= [];
lR = length(R);
for j = 1:lR
if j < lR
%2 especes d'affile identiques
if strcmp(R(j),R(j+1)) == 1
R(j) = {sprintf('2.0 %s',char(R(j)))};
R(j+1) = [];
lR = lR-1;
%2 especes d'affile non identiques
else
if strcmp(R(j),{'<->'}) == 1
indx_ar = j;
else
R(j) = {sprintf('1.0 %s',char(R(j)))};

```

```

        end
    end
elseif j == 1R
    %Controle derniere valeur
    R(j) = {sprintf('1.0 %s',char(R(j)))};
else
    break
end
end
EQ = R(1);
for j=2:1R
    if j ~= indx_ar
        if j-1 ~= indx_ar
            EQ = {sprintf('%s + %s',char(EQ),char(R(j)))};
        else
            EQ = {sprintf('%s %s',char(EQ),char(R(j)))};
        end
    else
        EQ = {sprintf('%s %s',char(EQ),char(R(j)))};
    end
end
EQS(k) = EQ; %Ensemble des equations: char(EQ)

%Nombre reactifs
nb_r = {indx_ar-1};
nb_rs = [nb_rs nb_r];

%Nombre produits
nb_p = {1R - indx_ar};
nb_ps = [nb_ps nb_p];
end

% Traitement des parametres
% Mettre les bons parametres s du modÈle Park:
%" 2 premiÈres cases: forward trans and vib, exponents"
%" 2 derniÈres cases: backward trans and vib, exponents"
if isempty(findstr(char(Re3(i,:)), '-- Dissociation reactions')) == 0
    schar = '%s0.5 0.5 1.0 0.0 %s';
elseif isempty(findstr(char(Re3(i,:)), '-- Neutral exchange reactions')) == 0
    schar = '%s1.0 0.0 1.0 0.0 %s';
elseif isempty(findstr(char(Re3(i,:)), '-- Associative ionization reactions')) == 0
    schar = '%s1.0 0.0 1.0 0.0 %s';
elseif isempty(findstr(char(Re3(i,:)), '-- Electron-impact ionization reactions')) == 0
    schar = '%s0.5 0.5 0.5 0.5 %s';
elseif isempty(findstr(char(Re3(i,:)), '-- Electron impact dissociation')) == 0
    schar = '%s? ? ? ? ?'; %forward T = T_vib^(s) * T_e^(s-1)
end
if isempty(B) == 0
    P=char(Re3(i,:));
    P=regexpr(P, ',', '');
    P=regexpr(P, 'fr={', '');
    P = strsplit(P, ', ');
    P = char(P);
    prm = [ ''];
    for l = 2:4 %Premiers parametres
        egl_i = findstr(P(l,:), '=');
        %Conversion des parametres
        if l == 2
            % P(2,:): A est le facteur frequence en (m3/mole)N^?1 . s?1 -> (m3/kmole)N^?1 . s?1
            prm_l = num2str(str2double(P(l,egl_i+1:end))/1000);
            prm_l = regexpr(prm_l, '+', '');
        elseif l == 4
            % P(4,:): T_a est la temperature d'activation -> convertie en Energie d'activation
            prm_l = P(l,egl_i+1:end);
            prm_l = num2str(str2double(P(l,egl_i+1:end))*R_gp);
            prm_l = regexpr(prm_l, '+', '');
        else
            prm_l = P(l,egl_i+1:end);
        end
        prm_l = regexpr(prm_l, ' ', '');
        prm = sprintf('%s%s ',prm,prm_l);
    end
    prm = {sprintf(schar,prm,press_dep)};
end

```

```

    prms = [prms; prm]; %Ensemble des parametres
end
% Traitement des efficacités
if isempty(C) == 0
    E = char(Re3(i,:));
    E=regexpr(E,'efficacités={','');
    E=regexpr(E,'}','','');
    E = strsplit(E,','');
    E = char(E);

    eff = [''];
    for m = 1:length(E);
        egl_i = findstr(E(m,:), '=');
        eff = sprintf('%s%s ',eff,strcat(E(m,egl_i+1:end)));
    end
    iC = {k};
    eff = strsplit(eff, ' ');
    eff(end) = [];
    effs = [effs; eff];
    iCs = [iCs iC];
end
end
o=1;
%Write file
fid = fopen(new_file_name,'wt');
fprintf(fid,'#\nVersion3.0\n#\n# Reaction File & Species File\n#\n#\n# Total Number of
Species\n#');
fprintf(fid,'\n%2i',length_spc);
fprintf(fid,'\n#\n#');
fprintf(fid,'%s',species);
fprintf(fid,'\n#\n# Total Number of Reactions\n#\n# Format is number of reactions\n# followed by
(for each reaction):\n#\n# number reactants\n# number products\n# reaction\n# modified
reaction\n# rates\n# efficacités in groups of 5');
fprintf(fid,'\n%2i',k);
fprintf(fid,'\n#\n#\n#');
for n=1:k
    fprintf(fid,'%i reactants\n',cell2mat(nb_rs(n)));
    fprintf(fid,'%i products\n',cell2mat(nb_ps(n)));
    fprintf(fid,'%s\n',char(EQS(n)));
    fprintf(fid,'%s\n',char(EQS(n)));
    fprintf(fid,'%s\n',char(prms(n)));
    if n == cell2mat(iCs(o)) %Detecte les equations qui ont des "efficacités"
        fprintf(fid,'%s\n',char(effs(o)));
        r=5;
        for p = 1:length_spc
            if p == r
                fprintf(fid,strcat(char(effs(o,p)),'\n'));
                r=r+5;
            elseif p == length_spc && r ~= p
                fprintf(fid,strcat(char(effs(o,p)),'\n'));
            else
                fprintf(fid,'%s ',char(effs(o,p)));
            end
        end
        if o < length(iCs)
            o=o+1;
        end
    else
        q=5;
        for p = 1:length_spc
            if p == q
                fprintf(fid,'1.0\n');
                q=q+5;
            elseif p == length_spc && q ~= p
                fprintf(fid,'1.0\n');
            else
                fprintf(fid,'1.0 ');
            end
        end
    end
end
end
fprintf(fid,'#');
fclose(fid);

```

```

% -----
% Script MatLab: Grid convergence 1
% -----
close all; clear all;
% Description:
% Ce script a pour but de récolter des données utiles pour une Etude de convergence des profils
Eilmer3:
% 1. Le maximum de profil sélectionne. Les valeurs trouvées sont ensuite écrites dans
Mx_Data.txt.
% 2. Le flux de chaleur intégrée sur toute la surface. Les valeurs trouvées sont ensuite
écrites dans HF_int.txt.

% Le fichier output "( ).txt" peut être utilisée par Nasagrdrv.m pour l'extrapolation de
Richardson.

% Fichiers nécessaires pour ce script:
% 1. Fichiers issus de e3post.py--job=x2 --output-file=PF --add-molef --slice-list=...
% 2. Fichiers heatfluxes: e3post.py --output-file=HF --job=x2 --heat-flux-list=...

% Guide:
% 1. Copier les fichiers issus des différents maillages dans le répertoire courant.
% 2. Définir les paramètres du script, indiquer les noms de fichier.
% 3. Run
% 4. Taper en commande ChPtext (si besoin) pour tester d'autres variables.

% Attention, si jamais les valeurs obtenues par ce script sont utilisées pour
une Etude de convergence, dans le cas de plusieurs super-blocs présents sur l'axe raffiné:
%(règles pour le nombre de mailles par maillages et par blocks de maillages):

% 1. Le nombre de mailles selon l'axe de raffinement souhaite doit respecter:
% Nb3/Nb2 = Nb2/Nb1 = ratio constant (avec 1,2, 3 les maillages).
% Le script find_nb_each_mesh.py trouve les bonnes valeurs.
%
% 2. Si le maillage est separé en blocks, selon l'axe de raffinement souhaite, le nombre par
blocks doit aussi respecter:
% nb_x1/nb_y1=nb_x2/nb_y2=nb_x3/nb_y3 et nb_y1/nb_z1=nb_y2/nb_z2=nb_y3/nb_z3
% (avec 1, 2,3 les blocks, x, y, z des maillages différents).
% Le script find_nb_each_blocks.m trouve les bonnes valeurs.

%Profile
data = 'T[0]'; title_dat = 'Température trans-rot';
% Propriétés figure
limxw = -0.00175; ylab = 'Ttr(K)';
savefig = 1; % 1 = oui , 0 = non
leg = {'nb = 40 ', 'nb = 60', 'nb = 90'}; loc='northeast';
% Un sélectionner le flux Etudié
select_HF = 1; %1 = HFc, 2 = HFd, 3 = HFr

for k= 1:3
%Entrer les noms: Maillage grossier k=1 -> maillage fin k=3
if k == 1
    filename1 = 'PF40'; filename2 = 'HF40';
end
if k == 2
    filename1 = 'PF60'; filename2 = 'HF60';
end
if k == 3
    filename1 = 'PF90'; filename2 = 'HF90';
end

%----- Ne plus remplir -----
HF = importdata(filename2, ' ',17); P = importdata(filename1, ' ',1);
%Recuperation et traitement des labels des données, definition des indices
Ptext = strsplit(P.textdata{1,:}); ChPtext = char(Ptext);
for i=3:length(ChPtext) %partir de 3 car les 2 premi@res colonnes sont inutiles.
    A = findstr(ChPtext(i,:), data);
    TF = isempty(A);
    if TF == 0
        indXT0 = i-2; indXT1 = i;
    end
end
end

%Vecteurs pour P

```

```

x = P.data(:,1);
if x<=0
    x = x + abs(max(x));
end
T0 = P.data(:,indxT0); T1 = P.data(:,indxT1);

%vecteurs pour HF
y = HF.data(:,1); HFc = HF.data(:,2); HFD = HF.data(:,3); HFr = HF.data(:,4);

%Integration
for l = 1:(length(y)-1)
    if k == 1
        [eI1(l)] = (y(l+1)-y(l))*HFc(l);
    end
    if k == 2
        [eI2(l)] = (y(l+1)-y(l))*HFc(l);
    end
    if k == 3
        [eI3(l)] = (y(l+1)-y(l))*HFc(l);
    end
end
if k == 1
    I1 = sum(eI1); T0_1=max(T0);
elseif k == 2
    I2 = sum(eI2); T0_2=max(T0);
else
    I3 = sum(eI3); T0_3=max(T0);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Les plots
figure(1)
plot(x,T0); hold on;
xlabel('x(m)', 'FontSize',10); ylabel(ylab, 'FontSize',10)
axis([limxw inf -inf inf])
axis square
if k == 3
    legend(leg, 'location',loc),grid on
    if savefig == 1
        formatspec = strcat(data, '_int.png'); print(gcf,formatspec, '-dpng', '-r300');
    end
end

%Les plots
figure(2)
hold on
if select_HF == 1
    plot(y,HFc)
    title_HF = 'Convective heat flux';
    ylabHF = 'HFc';
elseif select_HF == 2
    plot(y,abs(HFD))%,'r','linewidth',2,'markersize',10)
    title_HF = 'Duffusive heat flux';
    ylabHF = 'HFD';
elseif select_HF == 3
    plot(y,HFr, 'm','linewidth',2,'markersize',10)
    title_HF = 'Radiative heat flux';
    ylabHF = 'HFr';
end
if k == 3
    %Les labels, limites d'axes, legendes
    xlabel('y(mm)', 'FontSize',10); ylabel({ylabHF, '(W/m)'}, 'FontSize',10)
    axis tight; axis square;
    legend(leg, 'location',loc);grid on; box on;
    if savefig == 1
        formatspec = strcat(ylabHF, '_int.png'); print(gcf,formatspec, '-dpng', '-r300');
    end
end
end
end
I=[I3 I2 I1].';
save HF_Int.txt I -ASCII;
MxT0s = [T0_3 T0_2 T0_1].';
save Mx_Dat.txt MxT0s -ASCII;

```

```

% -----
% Script MatLab Nasagrdev
% -----
close all; clear all;

% This script is used to check grid convergence, based on Roache method.
% https://www.grc.nasa.gov/WWW/wind/valid/tutorial/spatconv.html
% Input: file created by integration.m script.

%Save ?
    savefig = 1;
%variable studied? (max T, Heat flux ...)
    %schar = 'HFc'
    schar = 'maxT'
%Normalised grid
    nm = [1.000000 1.500000 2.250000];
    plot_max_x = 2.5; % maximum dimension on x axis, for the plotting
%load files (comment the unwanted)
    load Mx_Dat.txt -ascii %max of data selected in integration file, for each grids
    %load HF_Int.txt -ascii %max of Integrate convective Heatflux on each grids
% select data (comment the unwanted)
    f = [Mx_Dat];
    %f = [HF_Int];
%labels
    HFylab = {'HFc(W)', 'HFc intégré sur la surface'};
    PFylab = {'Température trans-rot (K)'};
% -----
%Nasa verify
    r = 1.5; % grid refinement ratio
    f1 = f(1); % most refined grid
    f2 = f(2); % intermediate grid
    f3 = f(3); % coarsest grid
% order of convergence: theoretical order of
    p = log((f3 - f2)/(f2 - f1)) / log(r);
    fprintf('Theoretical order of convergence is: %f \n\n', p);
% Richardson extrapolation - use 2 finest grids to obtain estimate of parameter
% at zero grid spacing
    RE = f1 + (f1 - f2) / (r^p - 1);
    fprintf('Richardson extrapolated value of parameter at zero grid spacing: %f \n\n', RE)
% Safety factor
    Fs = 1.25; % because 3 grids were used
% Grid convergence index: grids 1 and 2
    GCI_12 = Fs * ((f1 - f2)/f1) / (r^p - 1) * 100;
% Grid convergence index: grids 2 and 3
    GCI_23 = Fs * ((f2 - f3)/f2) / (r^p - 1) * 100;
    fprintf('Grid convergence factors are \n');
    fprintf('(12):%f %% ', GCI_12)
    fprintf('and (23):%f %% \n\n', GCI_23)
% Check that solutions are in asymptotic range of convergence
    Rconv = GCI_23 / (r^p * GCI_12);
    if Rconv < 1.01 & Rconv > 0.99
        fprintf('The solutions are within the asymptotic range of convergence since 1 ~=%f
\n', Rconv)
    else
        fprintf('Range of convergence is not close enough to 1, at %f \n', Rconv)
    end
% Plot
    fig= figure(3);
    plot(nm, f, 'o-k'); hold on
    plot(0, RE, 's-r'); plot([0, nm(1)], [RE, f(1)], '--r')
    xlabel('Espacement de grille normalisÉ', 'FontSize', 10)
    if exist('Mx_Dat') == 1
        ylabel(PFylab, 'FontSize', 10) %12, 'rot', 0)
    elseif exist('HF_Int') == 1
        ylabel(HFylab, 'FontSize', 10) %12, 'rot', 0)
    else
    end
    axis tight; axis square;
    legend('Eilmer3', 'extrapolation de Richardson', 'location', southeast); grid on;
if savefig == 1
    formatspec = strcat(schar, 'NasaGCV.png'); print(gcf, formatspec, '-dpng', '-r300');
end

```

```

% -----
% Script MatLab: Find Number of Cells in each of the 2 Super Blocs,
% along a vertical or horizontal direction, in a 2D Mesh of Eilmer3
% -----
% Given 3 meshes X,Y,Z, refined differently because of the numbers of cells that each
% one have, in only one desired refining direction: Horizontal or Vertical.
%
% If along this direction, the meshes are separated in 3 blocks (1:east),(2:west):
% This script finds the number of cells per block (ex: x1 is the number of
% cells on the east block on the mesh X).
% It gives all the possible combinations.
%
% Find your equation system (end of script) using the number of cells that each of the meshes
% have, first.

close all;
clear all;

% East blocks (1) max number of cells.
max_y1 = 45;
max_z1 = 45;

k=0 %don't delete k=0
for x1 = 2:20 %For this given equation system, a range for x1 and x2 can be picked as wanted.
    for x2 = 2:20
        % -----
        % Equation system:
        if x1*x1~=40*x1
            x2=40-x1;
            y1=(3/2)*x1;
            y2=60-(3/2)*x1;
            z1=(9/4)*x1;
            z2=90-(9/4)*x1;

            % Don't modify here -----
            if round(y1) == y1 & round(y2) == y2 & round(z1) == z1 & round(z2) == z2 %nombre
entier
                if y1>=1 & y2>=1 & z1>=1 & z2>=1 %positif
                    if y1 <= max_y1 & z1 <= max_z1;
                        if y1 <= y2 & z1 <= z2
                            [x1 x2; y1 y2; z1 z2]
                                k=k+1;
                        end
                    end
                end
            end
        end
    end
end
% -----
end

end

% Type this in www.wolframalpha.com:
%
% solve [//math:x1/y1 = x2/y2//],[//math:y1/z1 =
y2/z2//],[//math:x1+x2=A//],[//math:y1+y2=B//],[//math:z1+z2=C//]
%
% with A the number of cells for the mesh the least refined, B ... , C the most refined (use
find_nb_each_mesh.py to get values),
% and find your equation system.
% In this example A = 40, B = 60, C = 90.

```

```

% -----
% Script MatLab: Find Number of Cells in each of the 3 Super Blocs,
% along a vertical or horizontal direction, in a 2D Mesh of Eilmer3
% -----
% Given 3 meshes X,Y,Z, refined differently because of the numbers of cells that each
% one have, in only one desired refining direction: Horizontal or Vertical.
%
% If along this direction, the meshes are separated in 3 blocks (1:east),(2:middle),(3:west):
% This script finds the number of cells per block (ex: x1 is the number of
% cells on the east block on the mesh X).
% It gives all the possible combinations.
%
% Find your equation system (end of script) using the number of cells that each of the meshes
% have, first.

close all;
clear all;

% East blocks (1) max number of cells.
max_y1 = 20;
max_z1 = 30;

k=0 %don't delete k=0
for x1 = 2:20 %For this given equation system, a range for x1 and x2 can be picked as wanted.
    for x2 = 2:20
        % -----
        % Equation system:
        if x1*x2*(x1+x2-40)~=0
            x3 = -x1-x2+40;
            y1 = (3*x1)/2;
            y2 = (3*x2)/2;
            y3 = -(3/2)*(x1+x2-40);
            z1 = (9*x1)/4;
            z2 = (9*x2)/4;
            z3 = (-9/4)*(x1+x2-40);
            % Don't modify here -----
            if round(x3) == x3 & round(y1) == y1 & round(y2) == y2 & round(y3) == y3 & round(z1)
== z1 & round(z2) == z2 & round(z3) == z3
                if x3>=1 & y1>=1 & y2>=1 & y3>=1 & z1>=1 & z2>=1 & z3>=1
                    if y1 <= max_y1 & z1 <= max_z1;
                        if y1 <= y2 & z1 <= z2
                            [x1 x2 x3; y1 y2 y3; z1 z2 z3]
                                k=k+1;
                        end
                    end
                end
            end
        end
    end
end
end

% -----
% Type this in www.wolframalpha.com:
%
% solve [//math:x1/y1 = x2/y2 = x3/y3//],[//math:y1/z1 = y2/z2 =
y3/z3//],[//math:x1+x2+x3=A//],[//math:y1+y2+y3=B//],[//math:z1+z2+z3=C//]
%
% with A the number of cells for the mesh the least refined, B ... , C the most refined (use
find_nb_each_mesh.py to get values),
% and find your equation system.
% In this example A = 40, B = 60, C = 90.

```


A.5 Script Python

```
## \Find number cells each mesh

from math import *
from cfpplib.grid.shock_layer_surface import *
from fractions import *
global x,y,z

#select range: number of cells of finest grid, here it's (100,140).
for k in range (100,140):
    for x in range(1,k):
        for y in range(1,x):
            for z in range(1,y):
                nb=Fraction(k,x)
                nb2=Fraction(x,y)
                nb3=Fraction(y,z)
                if nb == nb2 == nb3:
                    print k,x,y,z,nb3
```