Model Identification by Gradient Methods

Dr. Julien Billeter

Laboratoire d'Automatique
Ecole Polytechnique Fédérale de Lausanne (EPFL)

MLS-S03 | 2013-2014

Model Identification by Gradient methods

- Dynamic Models
  - Conservation of Mass (Concentration Measurements)
  - Conservation of Energy (Calorimetry)
  - Beer's Law (Spectroscopy)
- Integration of Dynamic Models
  - Euler's Method
  - Runge-Kutta's Methods (RK)
- Linear Regression (OLS) Problems
  - Calibration-free Calorimetry and Spectroscopy
- Gradient-based Nonlinear Regression (NLR) Methods
  - Steepest Descent Method (SD)
  - Newton-Raphson and Newton-Gauss Methods (NG)
  - Newton-Gauss Levenberg Marquardt Method (NGLM)
- References

Scalar, Vector and Matrix Notation

- **Scalars** a, ω, A, Ω
  (1×1) = number of $\dim\ 1$
  written in *lowercase/UPPERCASE italics*

- **Vectors** $\mathbf{a}, \boldsymbol{\omega}$
  $(n \times 1)$ = n-$\dim$ array (column vector)
  written in **lowercase boldface**

- **Matrices** $\mathbf{A}, \boldsymbol{\Omega}$
  $(n \times m)$ = array of dimensions n (rows) by m (columns)
  written in **UPPERCASE BOLDFACE**

Scalar, Vector and Matrix Operations

- Scalar multiplication $\alpha\mathbf{a},\ \alpha\mathbf{A}$
- Addition $\mathbf{a}+\mathbf{b},\ \mathbf{A}+\mathbf{B}$
- Multiplication $\mathbf{a}\ \mathbf{b},\ \mathbf{A}\ \mathbf{B}$
- Transposition $\mathbf{a}^{\mathrm{T}},\ \mathbf{A}^{\mathrm{T}}$
- Inverse (identity matrix) $\mathbf{A}\ \mathbf{A}^{-1}=\mathbf{A}^{-1}\mathbf{A}=\mathbf{I}$
- Rank and null space (kernel) $rank(\mathbf{A}),\ \mathbf{A}\ \mathbf{ker}(\mathbf{A})=\mathbf{0}$
- Rank-nullity theorem $dim(\mathbf{A})=rank(\mathbf{A})+nullity(\mathbf{A})$

Principal Component Analysis (PCA)

- Singular Value Decomposition (SVD) is a method to decompose a matrix $\mathbf{Y}$ into a product of orthonormal column ($\mathbf{U}$) and row ($\mathbf{V}^{\mathrm{T}}$) singular vectors weighted by singular values ($\mathbf{S}$).

$$\mathbf{Y} = \mathbf{U}\ \mathbf{S}\ \mathbf{V}^{\mathrm{T}} \qquad \text{with } \mathbf{S}^2 = \Lambda$$

- Principal Component Analysis (PCA) is a method to reduce the dimensionality of a matrix $\mathbf{Y}$ to its number of significant singular values.

$$\mathbf{Y} \approx \bar{\mathbf{Y}} = \bar{\mathbf{U}}\ \bar{\mathbf{S}}\ \bar{\mathbf{V}}^{\mathrm{T}} \qquad \text{with } \mathbf{Y} - \bar{\mathbf{Y}} = \text{noise}$$

Law of Conservation of Mass

- "Nothing is lost, nothing is created, everything is transformed"
  – Lavoisier (1743-1794)

$$\dot{m}(t) = \mathbf{1}_S^{\mathrm{T}} \dot{\mathbf{m}}(t) = 0 \rightarrow \mathbf{1}_S^{\mathrm{T}} \mathbf{M}_w \dot{\mathbf{n}}(t) = 0$$

$$\dot{\mathbf{n}}(t) = \mathbf{N}^{\mathrm{T}} V(t) \mathbf{r}(t) \pm \mathbf{W}_m \boldsymbol{\zeta}_m(t) + \mathbf{W}_{in} \mathbf{u}_{in}(t) - \frac{u_{out}(t)}{m(t)} \mathbf{n}(t), \qquad \mathbf{n}(0) = \mathbf{n}_0$$

$$\dot{\mathbf{c}}(t) = \mathbf{N}^{\mathrm{T}} \mathbf{r}(t) \pm \mathbf{W}_m \boldsymbol{\zeta}_c(t) + \mathbf{W}_{in} \frac{\mathbf{u}_{in}(t)}{V(t)} - \omega(t) \mathbf{c}(t), \qquad \mathbf{c}(0) = \mathbf{c}_0$$

$$\text{with} \quad \omega(t) = \frac{u_{out}(t)}{m(t)} + \frac{\dot{V}(t)}{V(t)} = \frac{\mathbf{1}_p^{\mathrm{T}} \mathbf{u}_{in}(t)}{m(t)} \pm \frac{\mathbf{1}_{p_m}^{\mathrm{T}} \boldsymbol{\zeta}_m(t)}{m(t)} - \frac{\dot{\rho}(t)}{\rho(t)}, \quad \dot{V}(t) = V(t) \left(\frac{\dot{m}(t)}{m(t)} - \frac{\dot{\rho}(t)}{\rho(t)} \right)$$

$$\text{and} \quad \dot{m}(t) = \mathbf{1}_p^{\mathrm{T}} \mathbf{u}_{in}(t) - u_{out}(t) \pm \mathbf{1}_{p_m}^{\mathrm{T}} \boldsymbol{\zeta}_m(t)$$

Law of Conservation of Energy

- "Any theory which demands the annihilation of *energy*, is necessarily erroneous" – Joule (1818-1889)

$$\dot{Q}(t) = 0 \rightarrow q_{acc}(t) = \mathbf{1}^{\mathrm{T}}\mathbf{q}(t)$$

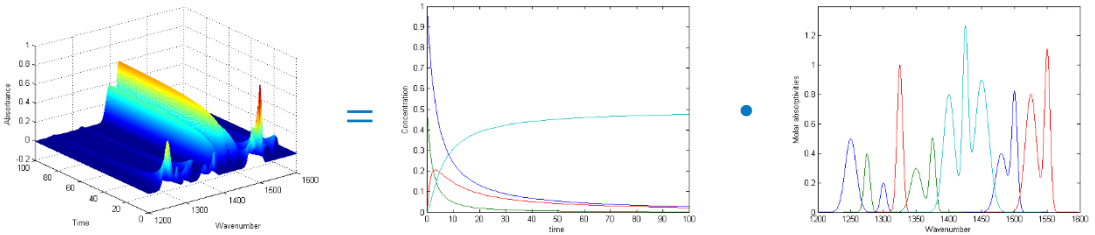
$$m(t)c_p(t)\dot{T}(t) = q_r \pm q_m + q_{ex} + q_{in} - q_{loss} + q_h - q_{out}, \quad T(0) = T_0$$

with
$$q_r(t) = V(t)(-\Delta\mathbf{h}_r^{\mathrm{T}})\mathbf{r}(t),$$
$$q_m(t) = (-\Delta\mathbf{h}_m^{\mathrm{T}})\boldsymbol{\zeta}_m(t),$$
$$q_{ex}(t) = UA\left(T_j - T(t)\right),$$
$$q_{in}(t) = \mathbf{c}_{p,in}^{\mathrm{T}}\mathbf{u}_{in}(t)\left(T_{in} - T(t)\right),$$
$$q_{out}(t) = c_p(t)u_{out}(t)T(t)$$

Beer's Law

- "The absorbance of a solution is proportional to the product of its concentration and the distance light travels through it"
  – Beer (1825-1863), Lambert (1728- 1777) and Bouguer (1698-1758)

$$\mathbf{Y} = \mathbf{C}\ \mathbf{A}$$



$\mathbf{Y}$ $\mathbf{C}$ $\mathbf{A}^{\mathrm{T}}$

with $\mathbf{Y}\ (nt \times nw)$,

$\mathbf{C} = [\mathbf{c}(t_1), \ldots, \mathbf{c}(t_{nt})]^{\mathrm{T}}\ \ (nt \times S)$

and $\mathbf{A} = [\ell\mathbf{a}(w_1), \ldots, \ell\mathbf{a}(w_{nw})]\ \ (S \times nw)$

Units conversion:

$$\mathrm{A} = -\log_{10}(\mathrm{T}),\ \ \mathrm{T} = \frac{\mathrm{I}}{\mathrm{I}_0}$$

Numerical Integration of ODE's

- Euler's method (implicit, explicit) was invented by the Swiss mathematician Euler (1707-1783)

$$\mathbf{y}_{i+1} = \mathbf{y}_i + h\dot{\mathbf{y}} + O(h^2)$$

h: integration stepsize

- Runge-Kutta's methods (RK2, RK4, explicit, implicit) were elaborated by Runge (1856-1927) and Kutta (1867-1944)

$$\mathbf{y}_{i+1} = \mathbf{y}_i + h\dot{\mathbf{y}}_{i+\frac{1}{2}} + O(h^3)$$

$$\text{with } \mathbf{y}_{i+\frac{1}{2}} = \mathbf{y}_i + \tfrac{h}{2}\dot{\mathbf{y}}(t_i, \mathbf{y}_i)$$

$$\dot{\mathbf{y}}_{i+\frac{1}{2}} = \dot{\mathbf{y}}(t_i + \tfrac{h}{2}, \mathbf{y}_{i+\frac{1}{2}})$$

$$\mathbf{y}_{i+1} = \mathbf{y}_i + \tfrac{1}{6}h(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) + O(h^5)$$

$$\text{with } \mathbf{k}_1 = \dot{\mathbf{y}}(t_i, \mathbf{y}_i),$$

$$\mathbf{k}_2 = \dot{\mathbf{y}}(t_i + \tfrac{h}{2}, \mathbf{y}_i + \tfrac{h}{2}\mathbf{k}_1)$$

$$\mathbf{k}_3 = \dot{\mathbf{y}}(t_i + \tfrac{h}{2}, \mathbf{y}_i + \tfrac{h}{2}\mathbf{k}_2),$$

$$\mathbf{k}_4 = \dot{\mathbf{y}}(t_i + h, \mathbf{y}_i + h\mathbf{k}_3)$$

Regression Problems

- A regression problem consists in minimizing the difference between measured output variables $\mathbf{y}(t)$ and modeled output variables $\hat{\mathbf{y}}(t, \mathbf{p}_f, \mathbf{p}_g)$ (the objective/cost function) by postulating a dynamic model $f(t, \mathbf{p}_f)$ and an output model $g(\mathbf{c}(t, \mathbf{p}_f), \mathbf{p}_g)$, and adjusting the parameters $\mathbf{p}_f$ (and $\mathbf{p}_g$).

$$\{\mathbf{p}_f, \mathbf{p}_g\}^* = \arg\left\{\min_{\mathbf{p}_f, \mathbf{p}_g} \ \phi\left(\mathbf{y}(t), \hat{\mathbf{y}}(t, \mathbf{p}_f, \mathbf{p}_g)\right)\right\}$$
$$\text{s.t.} \quad \dot{\hat{\mathbf{c}}}(t, \mathbf{p}_f) = f(t, \mathbf{p}_f)$$
$$\hat{\mathbf{y}}(t, \mathbf{p}_f, \mathbf{p}_g) = g\left(\mathbf{c}(t, \mathbf{p}_f), \mathbf{p}_g\right)$$

- In least-squares problems, ϕ is defined as the sum of squared residuals ($ssq = \mathbf{vec}(\mathbf{R})^{\mathrm{T}}\mathbf{vec}(\mathbf{R})$ with $\mathbf{R} = \mathbf{Y} - \hat{\mathbf{Y}}$) and the following matrices are defined:

$$\mathbf{Y} = [\mathbf{y}(t_1), \ldots, \mathbf{y}(t_{nt})]^{\mathrm{T}}, \ \hat{\mathbf{Y}} = [\hat{\mathbf{y}}(t_1, \mathbf{p}_f, \mathbf{p}_g), \ldots, \hat{\mathbf{y}}(t_{nt}, \mathbf{p}_f, \mathbf{p}_g)]^{\mathrm{T}}, \ \hat{\mathbf{C}} = [\mathbf{c}(t_1, \mathbf{p}_f), \ldots, \mathbf{c}(t_{nt}, \mathbf{p}_f)]^{\mathrm{T}}$$

Systems of Linear Equations

- A systems of linear equations can be written in matrix

$$\mathrm{S}: \begin{cases} a_{1,1}\ x_1 + \ldots + a_{1,n}\ x_n = y_1 \\ \quad \vdots \quad \ddots \quad \vdots \quad \vdots \\ a_{m,1}\ x_1 + \ldots + a_{m,n}\ x_n = y_m \end{cases} \quad \Rightarrow \quad \mathbf{A}\ \mathbf{x} = \mathbf{y}$$

with $\mathbf{A}\ (m \times n)$, $\mathbf{x}\ (n \times 1)$ the regressors and $\mathbf{y}\ (m \times 1)$ the regressands

- The number of solutions of S is:

∞	when $m < n$	underdetermined system	
1	$m = n$	determined system	$\Rightarrow \mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$
∞	$m > n$	overdetermined system	

Linear Regression (LR, OLS)

- For univariate data (data organized in a vector $\mathbf{y}$), a linear model relating the n independent variables (regressors, $\mathbf{x}$) to the $m > n$ dependent variables (regressands, $\mathbf{y}$) can be constructed as:

$$\mathbf{y} = \mathbf{A}\ \mathbf{x} \qquad \text{with } \mathbf{A}\ (m \times n),\ \mathbf{x}\ (n \times 1) \text{ and } \mathbf{y}\ (m \times 1)$$

$$\mathbf{x}^* = \arg\left\{\min_{\mathbf{x}} \mathbf{vec}(\mathbf{Ax} - \mathbf{y})^{\mathrm{T}}\, \mathbf{vec}(\mathbf{Ax} - \mathbf{y})\right\} = \mathbf{A}^{+}\mathbf{y} \quad \text{with } \mathbf{A}^{+} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}$$

The left pseudo-inverse $\mathbf{A}^{+}$ exists only if $rank(\mathbf{A}) = dim(\mathbf{A}) = n$

- For multivariate data (data organized in a matrix $\mathbf{Y}$), the linear model relating the $n \cdot w$ regressors $\mathbf{X}$ to the $m \cdot w$ regressands $\mathbf{Y}$ is built as:

$$\mathbf{Y} = \mathbf{A}\ \mathbf{X} \qquad \text{with } \mathbf{X}\ (n \times w) \text{ and } \mathbf{Y}\ (m \times w)$$

$$\mathbf{X}^* = \arg\left\{\min_{\mathbf{x}} \mathbf{vec}(\mathbf{AX} - \mathbf{Y})^{\mathrm{T}}\, \mathbf{vec}(\mathbf{AX} - \mathbf{Y})\right\} = \mathbf{A}^{+}\mathbf{Y}$$

Left or Right Pseudo-Inverse ?

- Left pseudo-inverse $\mathbf{A}^+ = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}$ $\quad rank(\mathbf{A}) = dim(\mathbf{A})$

$$\mathbf{Y} = \mathbf{A}\ \mathbf{X} \ \Rightarrow\ \mathbf{X}^* = \arg\left\{\min_{\mathbf{X}} \mathbf{vec}(\mathbf{AX}-\mathbf{Y})^{\mathrm{T}}\mathbf{vec}(\mathbf{AX}-\mathbf{Y})\right\} = \mathbf{A}^+\mathbf{Y}$$

Spectroscopy : $\mathbf{y} = \mathbf{C}\ \mathbf{a} \ \Rightarrow\ \mathbf{a}^* = \mathbf{C}^+\mathbf{Y}$

$\mathbf{Y} = \mathbf{C}\ \mathbf{A} \ \Rightarrow\ \mathbf{A}^* = \mathbf{C}^+\mathbf{Y}$ $\quad rank(\mathbf{C}) = S$

Calorimetry : $\mathbf{q}_r = \mathbf{R}_v(-\Delta\mathbf{h}_r) \Rightarrow -\Delta\mathbf{h}_r^* = \mathbf{R}_v^+\mathbf{q}_r$ $\quad rank(\mathbf{R}_v) = R$

- Right pseudo-inverse $\mathbf{X}^+ = \mathbf{X}^{\mathrm{T}}(\mathbf{X}\mathbf{X}^{\mathrm{T}})^{-1}$ $\quad rank(\mathbf{X}) = dim(\mathbf{X})$

$$\mathbf{Y} = \mathbf{A}\ \mathbf{X} \ \Rightarrow\ \mathbf{A}^* = \arg\left\{\min_{\mathbf{A}} \mathbf{vec}(\mathbf{AX}-\mathbf{Y})^{\mathrm{T}}\mathbf{vec}(\mathbf{AX}-\mathbf{Y})\right\} = \mathbf{Y}\mathbf{X}^+$$

Spectroscopy : $\mathbf{Y} = \mathbf{C}\ \mathbf{A} \ \Rightarrow\ \mathbf{C}^* = \mathbf{Y}\mathbf{A}^+$ $\quad rank(\mathbf{A}) = S$

with $\mathbf{Y}\ (nt \times nw)$, $\mathbf{C}\ (nt \times S)$, $\mathbf{A}\ (S \times nw)$, $\mathbf{q}_r (nt \times 1)$, $\mathbf{R}_v\ (nt \times R)$, $\Delta\mathbf{h}\ (R \times 1)$

Explicit vs Implicit Calibration

- In explicit calibration, a static calibration set is used to construct a calibration model from which concentrations are predicted for dynamic experiments.

$$\bar{\mathbf{Y}} = \bar{\mathbf{C}}\,\mathbf{A} \;\Rightarrow\; \hat{\mathbf{A}} = \bar{\mathbf{C}}^{+}\bar{\mathbf{Y}} \;\Rightarrow\; \hat{\tilde{\mathbf{C}}} = \tilde{\mathbf{Y}}\hat{\mathbf{A}}^{+}$$

- In implicit calibration (i.e. calibration free), dynamic experiments are used as an internal calibration set to eliminate the (static) linear counter-part $\mathbf{A}$.

$$\tilde{\mathbf{Y}} = \tilde{\mathbf{C}}\,\mathbf{A} \;\Rightarrow\; \hat{\mathbf{A}} = \hat{\tilde{\mathbf{C}}}^{+}\tilde{\mathbf{Y}} \;\Rightarrow\; \hat{\tilde{\mathbf{Y}}} = \tilde{\mathbf{C}}\,\hat{\tilde{\mathbf{C}}}^{+}\tilde{\mathbf{Y}}$$

The implicit calibration can even be used in case of rank-deficient data, i.e. when $\mathrm{rank}(\mathbf{C}) < S$

NONLINEAR (LEAST SQUARES) REGRESSION (NLR)

- Unlike linear regression problems, nonlinear regression problems are solved iteratively. Since linear parameters $\mathbf{p}_g$ can be estimated (eliminated) at each iteration, the optimization problem simplifies:

$$\mathbf{p}_f^* = \arg\left\{\min_{\mathbf{p}_f} \; ssq\right\}$$
$$\text{s.t.} \;\; \dot{\hat{\mathbf{c}}}(t, \mathbf{p}_f) = f(t, \mathbf{p}_f)$$

with $ssq = \mathbf{vec}\left(\mathbf{R}\right)^{\mathrm{T}} \mathbf{vec}\left(\mathbf{R}\right)$, $\mathbf{R} = \mathbf{Y} - \hat{\mathbf{Y}}(\mathbf{p}_f) = \mathbf{Y} - g_{lin}\left(\mathbf{C}(\mathbf{p}_f)\right) g_{lin}^{+}\left(\mathbf{C}(\mathbf{p}_f)\right)\mathbf{Y}$

- The (nested) linear regression problem is solved at each iteration as:

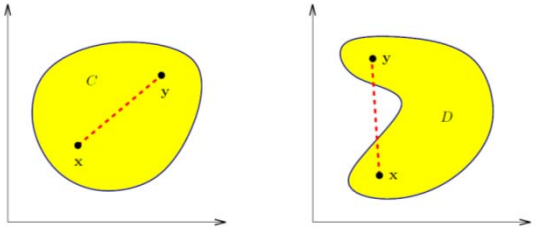
$$\hat{\mathbf{Y}}(\mathbf{p}_f) = g_{lin}\left(\mathbf{C}(\mathbf{p}_f), \mathbf{p}_g\right) = g_{lin}\left(\mathbf{C}(\mathbf{p}_f)\right)\mathbf{p}_g \;\Rightarrow\; \hat{\mathbf{p}}_g = g_{lin}^{+}\left(\mathbf{C}(\mathbf{p}_f)\right)\mathbf{Y}$$
$$\hat{\mathbf{Y}}(\mathbf{p}_f) = g_{lin}\left(\mathbf{C}(\mathbf{p}_f)\right) g_{lin}^{+}\left(\mathbf{C}(\mathbf{p}_f)\right)\mathbf{Y}$$

Convex Sets and Functions

- Convex Set: A set $\mathrm{C} \subset \mathbb{R}^n$ is said to be convex if for every points $\mathbf{x}, \mathbf{y} \in \mathrm{C}$, the points

$$\mathbf{z} = \lambda \mathbf{x} + (1-\lambda)\mathbf{y} \quad \forall \lambda \in [0,1],$$

are also in the set C.

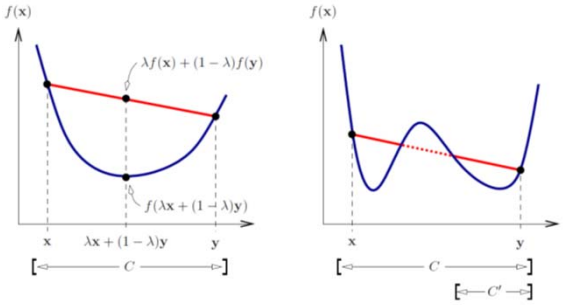


Courtesy of B. Chachuat

- Convex function: A function $f : \mathrm{C} \to \mathbb{R}$ defined on a convex set $\mathrm{C} \subset \mathbb{R}^n$ is said to be convex if

$$f\left(\lambda \mathbf{x} + (1-\lambda)\mathbf{y}\right) \leq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y}),$$

for each $\mathbf{x}, \mathbf{y} \in \mathrm{C}$ and $\lambda \in [0,1]$.



Courtesy of B. Chachuat

Necessary Conditions of Optimality (NCO)

- 1st order NCO: If $\mathbf{x}^*$ is a local minimum of $\phi : \mathrm{C} \to \mathbb{R}$, then

$$\nabla\phi(\mathbf{x}^*) = \mathbf{J}(\mathbf{x}^*) = 0$$

$$\mathbf{x}^* \text{ is a stationary point}$$

- 2nd order NCO: If $\mathbf{x}^*$ is a local minimum of $\phi : \mathrm{C} \to \mathbb{R}$, then

$$\nabla^2\phi(\mathbf{x}^*) = \mathbf{H}(\mathbf{x}^*) \text{ is positive semidefinite}$$

$$\mathbf{H}\mathbf{v}=\lambda\mathbf{v} \to (\mathbf{H} - \lambda\mathbf{I}_n)\mathbf{v} = \mathbf{0} \to p(\lambda) = \left|\mathbf{H} - \lambda\mathbf{I}_n\right| = 0 \to \lambda' s \geq 0$$

Note: 1st and 2nd order NCO form a set of sufficient conditions if $\phi(\mathbf{x})$ is a convex function defined on a convex set $\mathrm{C} \subset \mathbb{R}^n$.

Steepest (Gradient) Descent Method

- By definition, the gradient $\nabla\phi(\mathbf{x})$ points out the direction of the maximum of $\phi(\mathbf{x})$. Hence, a recurrence relation for finding the minimum is:

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \gamma \mathbf{J}(\mathbf{x}_i)^{\mathrm{T}} \mathbf{r}(\mathbf{x}_i)$$

with γ a stepsize parameter

$$\mathbf{J}(\mathbf{x}_i) = \frac{\mathbf{r}(\mathbf{x}_i + \mathrm{d}\mathbf{x}_i) - \mathbf{r}(\mathbf{x}_i)}{\mathrm{d}\mathbf{x}_i}$$

with $\mathrm{d}\mathbf{x}_i = (1+\varepsilon)\mathbf{x}_i$

- In simple algorithms, the stepsize parameter is fixed.
- In more sophisticated algorithms, the stepsize parameter is adapted at each iteration so that the step in the current search direction is maximum.

Newton-Raphson's Method (NR)

- The method of Newton (1642-1727) – Raphson (1648-1715) is an algorithm for finding iteratively the zeros of a system consisting of n equations and m unknowns:

$$\boldsymbol{\phi}(\mathbf{x}) = 0$$

For $m = n = 1$, one finds the well-known relation for $f(\mathrm{x}) = 0$,

$$x_{i+1} = x_i - \frac{\phi(x_i)}{\dot{\phi}(x_i)}$$

$m = n$, the unique solution is found as

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \nabla\boldsymbol{\phi}(\mathbf{x}_i)^{-1}\boldsymbol{\phi}(\mathbf{x}_i)$$

$m > n$, a solution in the least-squares sense is found as

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \nabla\boldsymbol{\phi}(\mathbf{x}_i)^{+}\boldsymbol{\phi}(\mathbf{x}_i)$$

Newton-Gauss Method (NG)

- The 1st order NCO directly provides a procedure for finding the minimum of a (regression) function, which can be solved using the Newton-Raphson method. This leads to the method of Newton (1642-1727) – Gauss (1777-1855):

$$\text{NCO:} \quad \nabla\phi(\mathbf{x}^*) = \mathbf{J}(\mathbf{x}^*) = 0$$

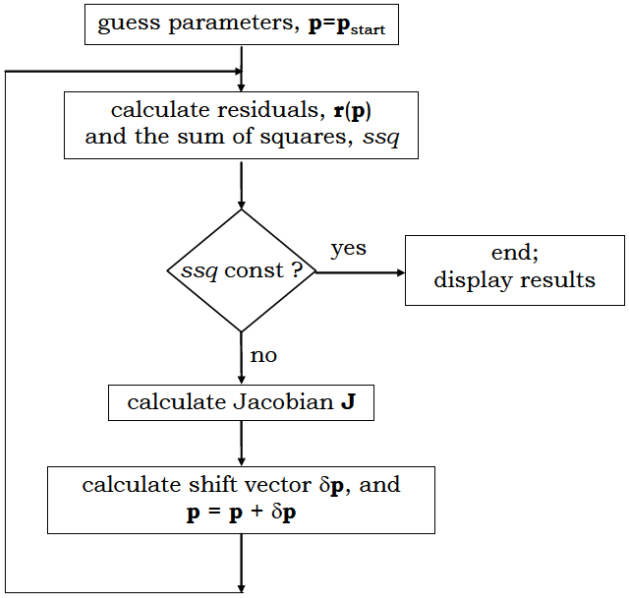
$$\text{NR:} \quad \mathbf{x}_{i+1} = \mathbf{x}_i - \nabla f(\mathbf{x}_i)^{-1} f(\mathbf{x}_i)$$

$$\text{with} \quad f(\mathbf{x}_i) = \nabla\phi(\mathbf{x}_i) = 2\mathbf{J}(\mathbf{x}_i)^{\mathrm{T}}\mathbf{r}(\mathbf{x}_i)$$

$$\nabla f(\mathbf{x}_i) = \nabla^2\phi(\mathbf{x}_i) \approx 2\mathbf{J}(\mathbf{x}_i)^{\mathrm{T}}\mathbf{J}(\mathbf{x}_i)$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \mathbf{H}(\mathbf{x}_i)^{-1}\mathbf{J}(\mathbf{x}_i)^{\mathrm{T}}\mathbf{r}(\mathbf{x}_i) \approx \mathbf{x}_i - \mathbf{J}(\mathbf{x}_i)^{+}\mathbf{r}(\mathbf{x}_i)$$

- Rel. convergence criterion $\left(\left|\frac{ssq_{i-1} - ssq_i}{ssq_{i-1}}\right| \leq \text{tol}\right)$



Courtesy of M. Maeder and Y.-M. Neuhold

NG Levenberg-Marquardt Method (NGLM)

- Levenberg (1919-1973) – Marquardt (1929-1997) modification allows interpolating between the Newton-Gauss method (NG) and the steepest descent method (SD):

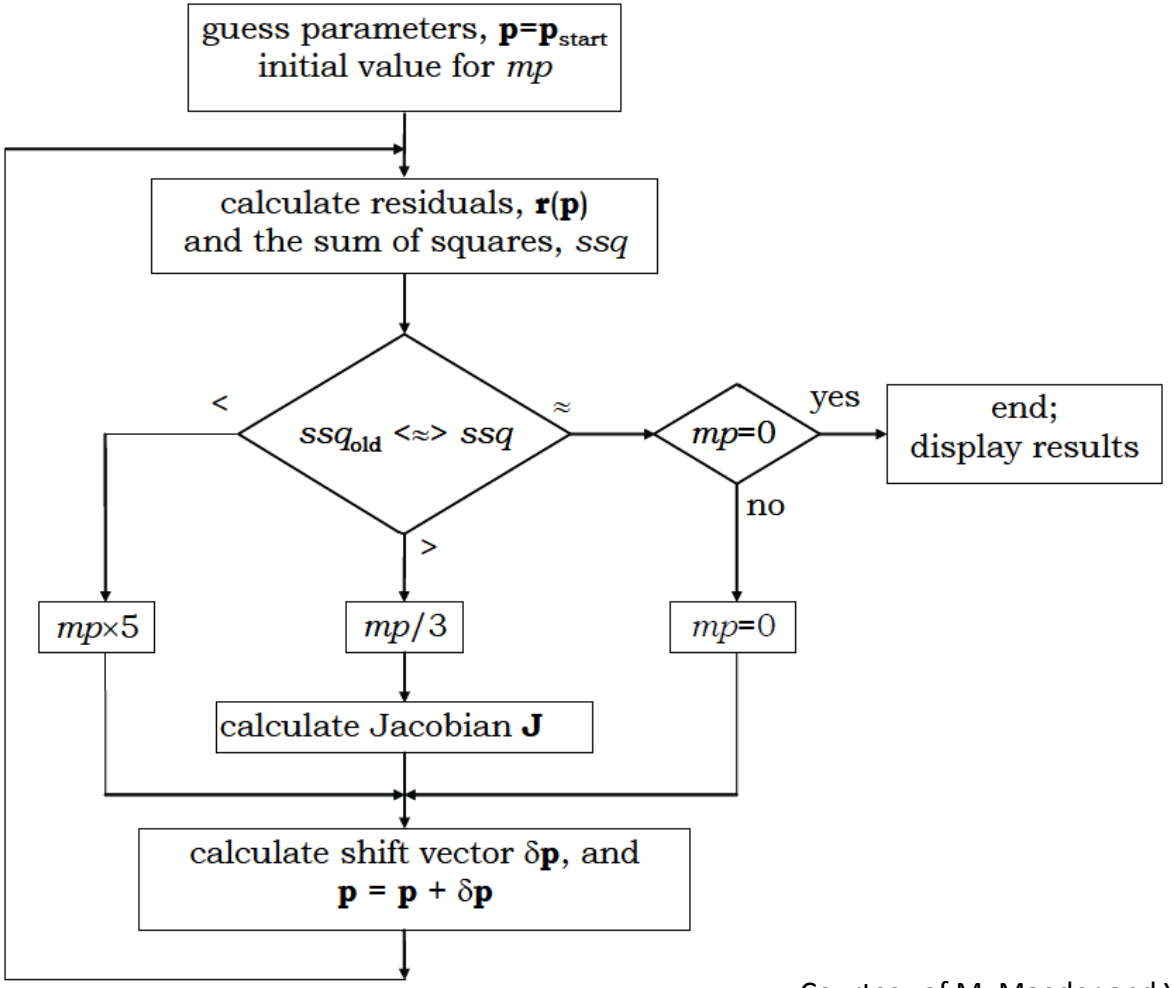
$$\mathbf{x}_{i+1} = \mathbf{x}_i - \left(\mathbf{H}(\mathbf{x}_i) + \lambda_i \mathbf{I}\right)^{-1} \mathbf{J}(\mathbf{x}_i)^{\mathrm{T}} \mathbf{r}(\mathbf{x}_i)$$

with $\mathbf{H}(\mathbf{x}_i) \approx \mathbf{J}(\mathbf{x}_i)^{\mathrm{T}} \mathbf{J}(\mathbf{x}_i)$ and $\lambda_i \geq 0$: Marquardt parameter (mp)

For $\lambda_i = 0 \Rightarrow$ NG ; for $\lambda_i \to \infty \Rightarrow$ SD (shorter stepsize)

The parameter λ_i is adapted at each iteration according to heuristic arguments to avoid divergence problems due to a bad choice of the initial guesses in the original NG method.

NGLM ALGORITHM



Courtesy of M. Maeder and Y.-M. Neuhold

Statistics Provided by Gradient Methods

- Degree of freedom $$df = \#\mathbf{Y} - \left(\#\mathbf{p}_f + \#\mathbf{p}_g\right)$$
- Residual variance $$\sigma_r^2 = \frac{ssq}{df}$$
- Variance/covariance (correlation) matrix $$\boldsymbol{\sigma}_{p_f}^2 = \sigma_r^2 \mathbf{H}^{-1}$$
- Correlation matrix $$corr(p_{f,i}, p_{f,j}) = \frac{\sigma_{p_f}^2(i,j)}{\sigma_{p_f}(i,i) \cdot \sigma_{p_f}(j,j)} \in [0,1]$$

REFERENCES

☺ M. Maeder, Y.-M. Neuhold, Practical Data Analysis in Chemistry, Elsevier, 2007

- M. Maeder, Y.-M. Neuhold, Chapter 7 of P. Gemperline (ed.) Practical Guide to Chemometrics, Taylor and Francis, 2006
- W.H. Press, W.T. Vetterling, S.A. Teukolsky, B.P. Flannery, Numerical Recipes in C++ – The art of Scientific Computing, 2nd Edition, Cambridge University Press, 2005
- B. Chachuat, G. François, Nonlinear Dynamic Optimization – From Theory to Practice, Lecture notes, McMaster University - EPFL, 2009

REFERENCES

☺ G. Puxty, M. Maeder, K. Hungerbühler, Chemom. Intell. Lab. Syst. 81 (2006), 149

- V.M. Taavitsainen, H. Haario, J. Chemom. 15 (2001), 215
- V.M. Taavitsainen, H. Haario et al, J. Chemom. 17 (2003), 140
- M. Maeder, A.D. Zuberbühler, Anal. Chem. 62 (1990), 2220
- J. Billeter, Chemometric Methods for Prediction of Uncertainties and Spectral Validation of Rank Deficient Mechanisms in Kinetic Hard-modelling of Spectroscopic Data, Doctoral dissertation n°18311, ETH Zurich, 2009