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**
  
  \((1 \times 1) = \text{number of dim 1}\)
  
  written in *lowercase/UPPERCASE italics*

- **Vectors**
  
  \((n \times 1) = n\text{-dim array (column vector)}\)
  
  written in *lowercase boldface*

- **Matrices**
  
  \((n \times m) = \text{array of dimensions } n \text{ (rows) by } m \text{ (columns)}\)
  
  written in *UPPERCASE BOLDFACE*
**Scalar, Vector and Matrix Operations**

- Scalar multiplication: $\alpha a$, $\alpha A$
- Addition: $a + b$, $A + B$
- Multiplication: $a b$, $A B$
- Transposition: $a^T$, $A^T$
- Inverse (identity matrix): $A A^{-1} = A^{-1}A = I$
- Rank and null space (kernel): $\text{rank}(A)$, $A \ker(A) = 0$
- Rank-nullity theorem: $\text{dim}(A) = \text{rank}(A) + \text{nullity}(A)$
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 ($\mathbf{V}^T$) singular vectors weighted by singular values ($\mathbf{S}$).

  \[
  \mathbf{Y} = \mathbf{U} \mathbf{S} \mathbf{V}^T \quad \text{with} \quad \mathbf{S}^2 = \mathbf{\Lambda}
  \]

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

  \[
  \mathbf{Y} \approx \overline{\mathbf{Y}} = \overline{\mathbf{U}} \overline{\mathbf{S}} \overline{\mathbf{V}}^T \quad \text{with} \quad \mathbf{Y} - \overline{\mathbf{Y}} = \text{noise}
  \]
LAW OF CONSERVATION OF MASS

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

\[ \dot{m}(t) = 1^T_S \dot{m}(t) = 0 \rightarrow 1^T_S M_w \dot{n}(t) = 0 \]

\[ \dot{n}(t) = N^T V(t) r(t) \pm W_m \xi_m(t) + W_{in} u_{in}(t) - \frac{u_{out}(t)}{m(t)} n(t), \quad n(0) = n_0 \]

\[ \dot{c}(t) = N^T r(t) \pm W_m \xi_c(t) + W_{in} \frac{u_{in}(t)}{V(t)} - \omega(t) c(t), \quad c(0) = c_0 \]

with \( \omega(t) = \frac{u_{out}(t)}{m(t)} + \frac{\dot{V}(t)}{V(t)} + \frac{1^T p u_{in}(t)}{m(t)} \pm \frac{1^T p_m \xi_m(t)}{m(t)} - \frac{\dot{\rho}(t)}{\rho(t)}, \quad \dot{V}(t) = V(t) \left( \frac{\dot{m}(t)}{m(t)} - \frac{\dot{\rho}(t)}{\rho(t)} \right) \)

and \( \dot{m}(t) = 1^T_p u_{in}(t) - u_{out}(t) \pm 1^T p_m \xi_m(t) \)
LAW OF CONSERVATION OF ENERGY

• “Any theory which demands the annihilation of energy, is necessarily erroneous” – Joule (1818-1889)

\[ \dot{Q}(t) = 0 \rightarrow q_{acc}(t) = 1^T q(t) \]

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

with

\[ q_r(t) = V(t)(-\Delta h^T_r)r(t), \]
\[ q_m(t) = (-\Delta h^T_m)\xi_m(t), \]
\[ q_{ex}(t) = UA(T_j - T(t)), \]
\[ q_{in}(t) = c_{p, in}^T u_{in}(t)(T_{in} - T(t)), \]
\[ q_{out}(t) = c_p(t)u_{out}(t)T(t) \]
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)

\[ Y = C \cdot A \]

with \( Y \) \((nt \times nw)\),

\[ C = [c(t_1), \ldots, c(t_{nt})]^T \quad (nt \times S') \]

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

Units conversion:

\[ A = -\log_{10}(T), \quad T = \frac{I}{I_0} \]
NUMERICAL INTEGRATION OF ODE’S

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

  \[ y_{i+1} = y_i + h\dot{y} + O(h^2) \]

  \( h \): integration stepsize

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

  \[ y_{i+1} = y_i + h\dot{y}_{i + \frac{1}{2}} + O(h^3) \]

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

  with \( y_{i + \frac{1}{2}} = y_i + \frac{h}{2}\dot{y}(t_i, y_i) \)

  \[ y_{i+1} = y_i + \frac{1}{6}h(\dot{k}_1 + 2\dot{k}_2 + 2\dot{k}_3 + \dot{k}_4) + O(h^5) \]

  with \( \dot{k}_1 = \dot{y}(t_i, y_i) \),

  \( \dot{k}_2 = \dot{y}(t_i + \frac{h}{2}, y_i + \frac{h}{2}\dot{k}_1) \),

  \( \dot{k}_3 = \dot{y}(t_i + \frac{h}{2}, y_i + \frac{h}{2}\dot{k}_2) \),

  \( \dot{k}_4 = \dot{y}(t_i + h, y_i + h\dot{k}_3) \)
**Regression Problems**

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

\[
\{p_f, p_g\}^* = \arg\left\{\min_{p_f, p_g} \phi\left(y(t), \hat{y}(t, p_f, p_g)\right)\right\}
\]

\[
\text{s.t. } \dot{c}(t, p_f) = f(t, p_f)
\]

\[
\hat{y}(t, p_f, p_g) = g\left(c(t, p_f), p_g\right)
\]

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

\[
Y = [y(t_1), \ldots, y(t_{nt})]^T, \quad \hat{Y} = [\hat{y}(t_1, p_f, p_g), \ldots, \hat{y}(t_{nt}, p_f, p_g)]^T, \quad \hat{C} = [c(t_1, p_f), \ldots, c(t_{nt}, p_f)]^T
\]
Systems of Linear Equations

- A system of linear equations can be written in matrix form as:

\[
\begin{align*}
S: & \begin{cases}
  a_{1,1} x_1 + \ldots + a_{1,n} x_n = y_1 \\
  \vdots & \vdots & \vdots & \vdots \\
  a_{m,1} x_1 + \ldots + a_{m,n} x_n = y_m
\end{cases} & \Rightarrow & A \mathbf{x} = \mathbf{y}
\end{align*}
\]

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

- The number of solutions of \( S \) is:

  \[
  \begin{align*}
  \infty & \quad \text{when} & m < n & \quad \text{underdetermined system} \\
  1 & \quad m = n & \quad \text{determined system} & \Rightarrow \mathbf{x} = A^{-1}\mathbf{y} \\
  \infty & \quad m > n & \quad \text{overdetermined system}
  \end{align*}
  \]
**Linear Regression (LR, OLS)**

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

\[
y = Ax \quad \text{with} \quad A \ (m \times n), \ x \ (n \times 1) \text{ and } y \ (m \times 1)
\]

\[
x^* = \arg\left\{ \min_x \text{vec}(Ax - y)^T \text{vec}(Ax - y) \right\} = A^+y \quad \text{with} \quad A^+ = (A^T A)^{-1} A^T
\]

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

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

\[
Y = AX \quad \text{with} \quad X \ (n \times w) \text{ and } Y \ (m \times w)
\]

\[
X^* = \arg\left\{ \min_x \text{vec}(AX - Y)^T \text{vec}(AX - Y) \right\} = A^+Y
\]
LEFT OR RIGHT PSEUDO-INVERSE?

- Left pseudo-inverse \( A^+ = (A^T A)^{-1} A^T \)
  \( \text{rank}(A) = \text{dim}(A) \)

  \( Y = A \ X \Rightarrow X^* = \arg \left\{ \min_x \text{vec}(AX - Y)^T \text{vec}(AX - Y) \right\} = A^+ Y \)

  Spectroscopy: \( y = C \ a \ \Rightarrow a^* = C^+ Y \)
  \( Y = C \ A \Rightarrow A^* = C^+ Y \)

  Calorimetry: \( q_r = R_v (-\Delta h_r) \Rightarrow -\Delta h^*_r = R^+_v q_r \)

- Right pseudo-inverse \( X^+ = X^T (XX^T)^{-1} \)
  \( \text{rank}(X) = \text{dim}(X) \)

  \( Y = A \ X \Rightarrow A^* = \arg \left\{ \min_A \text{vec}(AX - Y)^T \text{vec}(AX - Y) \right\} = YX^+ \)

  Spectroscopy: \( Y = C \ A \Rightarrow C^* = YA^+ \)

  \( \text{rank}(A) = S \)

with \( Y \ (nt \times nw) \), \( C \ (nt \times S) \), \( A \ (S \times nw) \), \( q_r \ (nt \times 1) \), \( R_v \ (nt \times R) \), \( \Delta h \ (R \times 1) \)
EXPLICIT VS IMPPLICIT CALIBRATION

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

\[
\bar{Y} = \bar{C} \ A \ \Rightarrow \ \hat{A} = \bar{C}^+ \bar{Y} \ \Rightarrow \ \hat{C} = \bar{Y} \hat{A}^+
\]

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

\[
\tilde{Y} = \tilde{C} \ A \ \Rightarrow \ \hat{A} = \tilde{C}^+ \tilde{Y} \ \Rightarrow \ \hat{Y} = \tilde{C} \ \tilde{C}^+ \tilde{Y}
\]

The implicit calibration can even be used in case of rank-deficient data, i.e. when \( \text{rank}(C) < S \).
NONLINEAR (LEAST SQUARES) REGRESSION (NLR)

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

$$p_f^* = \arg\min_{p_f} \{ ssq \}$$

s.t. $\hat{c}(t, p_f) = f(t, p_f)$

with $ssq = \text{vec}(R)^T \text{vec}(R)$, $R = Y - \hat{Y}(p_f) = Y - g_{lin}(C(p_f))g_{lin}^+(C(p_f))Y$

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

$$\hat{Y}(p_f) = g_{lin}(C(p_f), p_g) = g_{lin}(C(p_f))p_g \Rightarrow \hat{p}_g = g_{lin}^+(C(p_f))Y$$

$$\hat{Y}(p_f) = g_{lin}(C(p_f))g_{lin}^+(C(p_f))Y$$
**Convex Sets and Functions**

- **Convex Set:** A set $C \subset \mathbb{R}^n$ is said to be convex if for every points $x, y \in C$, the points $z = \lambda x + (1 - \lambda)y \ \forall \lambda \in [0,1]$, are also in the set $C$.

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

  $$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y),$$

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

---

*Courtesy of B. Chachuat*
NECESSARY CONDITIONS OF OPTIMALITY (NCO)

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

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

$x^*$ is a stationary point

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

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

$$Hv = \lambda v \rightarrow (H - \lambda I_n)v = 0 \rightarrow p(\lambda) = \begin{vmatrix} H - \lambda I_n \end{vmatrix} = 0 \rightarrow \lambda' s \geq 0$$

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

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

$$x_{i+1} = x_i - \gamma J(x_i)^T r(x_i)$$

with $\gamma$ a stepsize parameter

$$J(x_i) = \frac{r(x_i + d x_i) - r(x_i)}{d x_i}$$

with $d x_i = (1 + \varepsilon) x_i$

• 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:

$$\phi(x) = 0$$

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

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

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

$$x_{i+1} = x_i - \nabla \phi(x_i)^{-1} \phi(x_i)$$

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

$$x_{i+1} = x_i - \nabla \phi(x_i)^+ \phi(x_i)$$
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):

NCO: \( \nabla \phi(x^*) = J(x^*) = 0 \)

NR: \( x_{i+1} = x_i - \nabla f(x_i)^{-1} f(x_i) \)

with \( f(x_i) = \nabla \phi(x_i) = 2J(x_i)^T r(x_i) \)

\( \nabla f(x_i) = \nabla^2 \phi(x_i) \approx 2J(x_i)^T J(x_i) \)

\( x_{i+1} = x_i - [H(x_i)]^{-1} J(x_i)^T r(x_i) \approx x_i - [J(x_i)]^+ r(x_i) \)

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

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):

\[
x_{i+1} = x_i - \left( H(x_i) + \lambda_i I \right)^{-1} J(x_i)^T r(x_i)
\]

with \( H(x_i) \approx J(x_i)^T J(x_i) \) and \( \lambda_i \geq 0 \): Marquardt parameter (mp)

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

The parameter \( \lambda_i \) 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

guess parameters, $p = p_{start}$
initial value for $mp$

calculate residuals, $r(p)$
and the sum of squares, $ssq$

$sq_{old} <\approx ssq <\approx mp = 0$

end; display results

$mp <\approx 5$

$mp/3$

$mp=0$

calculate Jacobian $J$

calculate shift vector $\delta p$, and
$p = p + \delta p$
STATISTICS PROVIDED BY GRADIENT METHODS

- Degree of freedom
  \[ df = \# Y - \left( \# p_f + \# p_g \right) \]

- Residual variance
  \[ \sigma_r^2 = \frac{ssq}{df} \]

- Variance/covariance (correlation) matrix
  \[ \sigma_{pf}^2 = \sigma_r^2 H^{-1} \]

- Correlation matrix
  \[ corr(p_f,i, p_f,j) = \frac{\sigma_{pf}^2(i,j)}{\sigma_{pf}(i,i) \cdot \sigma_{pf}(j,j)} \in [0,1] \]
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


• B. Chachuat, G. François, Nonlinear Dynamic Optimization – From Theory to Practice, Lecture notes, McMaster University - EPFL, 2009
REFERENCES


• 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