# Data and Model Fitting



January 25, 2023

# Data and Model Fitting

## Ordinary Linear Least Square

- linear model assumes $y = \mathbf{x}^T\beta$ with model parameter $\beta$.
- ordinary linear least square fit solves,

$$\arg\min_{\beta} (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)$$

  , which is equivalently to solve $\mathbf{X}^T\mathbf{X}\beta = \mathbf{X}^T\mathbf{y}$
- $\mathbf{X}^T\mathbf{X}$ is at least positive semi-definite, appropriate linear system solvers are available in Mathwrist.

# Data and Model Fitting

## Generalized Ridge Regression

- alternatively and usually a better way of calibrating model parameter $\beta$ is to add a regularization term and solve,

$$\arg\min_{\beta} \left(\mathbf{y} - \mathbf{X}\beta\right)^T \left(\mathbf{y} - \mathbf{X}\beta\right) + \lambda\beta^T \mathbf{\Omega}\beta \tag{1}$$

  , where $\lambda > 0$ is a penalty factor and the regularization matrix $\mathbf{\Omega}$ is positive definite.
- formulation (1) is also to solve a linear system,

$$\left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{\Omega}\right)\beta = \mathbf{X}^T\mathbf{y}$$

- if $\mathbf{\Omega}$ is identity, (1) is the standard ridge regression, hence (1) sometimes is also called generalized ridge regression. Mathwrist provides a function linear_fit::grr() to solve (1).

# Data and Model Fitting

## Generalized Cross Validation (GCV)

- penalty factor $\lambda$ could be an experimental choice.
- GCV method computes the optimal $\lambda$ based on the data noise level.
- write observation of linear model as $\mathbf{y} = \mathbf{X}\beta + \epsilon$, GCV solves,

$$\arg\min_{\lambda} V(\lambda) = \frac{\|(\mathbf{I} - \mathbf{H}(\lambda))\mathbf{y}\|^2}{\mathrm{Tr}(\mathbf{I} - \mathbf{H}(\lambda))^2}$$

, where $\mathbf{H}(\lambda)$ is the unique symmetric influence matrix,

$$\mathbf{H}(\lambda) = \mathbf{X}\left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{\Omega}\right)^{-1}\mathbf{X}^T$$

- if parameter $\lambda \leq 0$ is passed to function linear_fit::grr(), we use GCV method to compute an optimal penalty factor.

# Data and Model Fitting

## Linearly Constrained Linear Least Square

- model parameter $\beta$ maybe imposed to general linear constraints and simple bounds.
- Mathwrist provides a function linear_fit::lsq() to solve the following linearly constrained linear least square problem,

$$\begin{array}{c} \arg\min_{\beta} \left(\mathbf{y} - \mathbf{X}\beta\right)^{T} \left(\mathbf{y} - \mathbf{X}\beta\right) \quad \text{s.t.} \\ \mathbf{b}_{l} \leq \mathbf{A}\beta \leq \mathbf{b}_{u} \text{ and} \\ \mathbf{l} \leq \beta \leq \mathbf{u} \end{array} \tag{2}$$

, which effectively is a convex quadratic programming (QP) problem.

# Data and Model Fitting

## Nonlinear Least Square

Given a nonlinear function $y = h(\mathbf{x}; \beta)$ with model parameter $\beta$ and $m$ number of observations $(y_i, \mathbf{x}_i)$, $i = 0, \cdots, m-1$, calibrate $\beta$ by minimizing the $l_2$ norm of residual vector,

$$\arg\min_{\beta} \psi(\beta) = \frac{1}{2}\|\mathbf{r}(\beta)\|_2^2 \tag{3}$$

, where the $i$-th element $r_i(\beta) = h(\mathbf{x}_i; \beta) - y_i$. Let $\mathbf{J}(\beta)$ be the Jacobian matrix of the residual vector $\mathbf{r}(\beta)$. The gradient and Hessian of $\psi(\beta)$ are

$$\nabla\psi(\beta) = \sum_{i=0}^{m-1} r_i(\beta)\nabla r_i(\beta) = \mathbf{J}^T(\beta)\mathbf{r}(\beta) \tag{4}$$

$$\nabla^2\psi(\beta) = \mathbf{J}^T(\beta)\mathbf{J}(\beta) + \sum_{i=0}^{m-1} r_i(\beta)\nabla^2 r_i(\beta) \tag{5}$$

# Data and Model Fitting

## Nonlinear Least Square: Gauss-Newton

- approximate the true Hessian matrix in equation (5) by $\mathbf{J}(\beta)^T\mathbf{J}(\beta)$.
- use a line search algorithm and iteratively computes a Newton search direction $\mathbf{p}$ at each step,

$$\mathbf{J}(\beta)^T\mathbf{J}(\beta)\mathbf{p} = -\nabla\psi(\beta) = -\mathbf{J}^T(\beta)\mathbf{r}(\beta)$$

- if the Jacobian matrix $\mathbf{J}(\beta)$ has rank deficiency, it produces unstable model calibration.
- if the residual $\mathbf{r}(\beta)$ is naturally large or non-negligible at certain point of the calibration, ignoring the second term in equation (5) produces incorrect search direction $\mathbf{p}$.

# Data and Model Fitting

## Nonlinear Least Square: Modified Gauss-Newton

- at each iteration in the line search, compute SVD, $\mathbf{J}(\beta) = \mathbf{USV}^T$.
- the Newton direction $\mathbf{p}$ wrt the true Hessian solves

$$\left(\mathbf{S}^2\mathbf{V}^T + \mathbf{V}^T\mathbf{Q}(\beta)\right)\mathbf{p} = -\mathbf{S}\bar{\mathbf{r}}(\beta) \tag{6}$$

, where $\mathbf{Q}(\beta) = \sum_{i=0}^{m-1} r_i(\beta)\nabla^2 r_i(\beta)$, $\bar{\mathbf{r}}(\beta) = \mathbf{U}^T\mathbf{r}(\beta)$.

- let $\mathbf{S}_d$ be the leading submatrix of $d$ number of dominant singulars in $\mathbf{S}$. Accordingly, let $\mathbf{V}_d$ be the first $d$ columns of $\mathbf{V}$, the principle components.
- test whether $\sqrt{|\mathbf{r}(\beta)|_\infty}$ is small enough relative to the smallest singulars in $\mathbf{S}_d$. If so, we ignore $\mathbf{Q}(\beta)$ and write direction as $\mathbf{p} = \mathbf{V}_d\mathbf{p}_d$. Let $\bar{\mathbf{r}}_d(\beta)$ be the first $d$ elements of $\bar{\mathbf{r}}(\beta)$ and solve $\mathbf{S}\mathbf{p}_d = -\bar{\mathbf{r}}_d(\beta)$

# Data and Model Fitting

## Nonlinear Least Square: Modified Gauss-Newton (continued)

- if $\mathbf{Q}(\beta)$ cannot be ignored, we approximate it by finite difference and solve the direction $\mathbf{p}$ in the full space of $\mathbf{V}$, $\mathbf{p} = \mathbf{V}\bar{\mathbf{p}}$,

$$\left(\mathbf{S}^2 + \mathbf{V}^T\mathbf{Q}(\beta)\mathbf{V}\right)\bar{\mathbf{p}} = -\mathbf{S}\bar{\mathbf{r}}(\beta) \tag{7}$$

- the second order term $\mathbf{Q}(\beta)$ could be indefinite. We use modified Cholesky to solve equation (7). This is similar to the modified Newton method in unconstrained optimization.

# Data and Model Fitting

## Nonlinear Least Square: Levenberg-Marquardt

- a special case of the trust region algorithm, uses $\mathbf{J}^T(\beta)\mathbf{J}(\beta)$ to approximate the true Hessian (5).
- at each trust region iteration, solve a sub problem

$$\arg\min_p \mathbf{p}^T\mathbf{J}^T(\beta)\mathbf{r}(\beta) + \frac{1}{2}\mathbf{p}^T\mathbf{J}^T(\beta)\mathbf{J}(\beta)\mathbf{p}, \text{ s.t. } \|\mathbf{p}\| \leq \Delta_k \tag{8}$$

- $\mathbf{p}^*$ is a solution of the trust region subproblem (8) if and only if $\exists\lambda \geq 0$ such that
  1. $\left(\mathbf{J}^T(\beta)\mathbf{J}(\beta) + \lambda\mathbf{I}\right)$ is positive semidefinite and
  2. $\left(\mathbf{J}^T(\beta)\mathbf{J}(\beta) + \lambda\mathbf{I}\right)\mathbf{p}^* = -\mathbf{J}^T(\beta)\mathbf{r}(\beta)$ and
  3. $\lambda\left(\Delta - \|\mathbf{p}^*\|\right) = 0$

  The first condition is automatically satisfied here.

# Data and Model Fitting

## Nonlinear Least Square: Levenberg-Marquardt (continued)

- write $\mathbf{p}(\lambda)$ as a function of $\lambda$ computed from the second condition,

$$\mathbf{p}(\lambda) = -\left(\mathbf{J}^T(\beta)\mathbf{J}(\beta) + \lambda\mathbf{I}\right)^{-1}\mathbf{J}^T(\beta)\mathbf{r}(\beta) \tag{9}$$

- if $\|\mathbf{p}(\lambda = 0)\| < \Delta_k$, $\mathbf{p}(\lambda = 0)$ is an exact solution of trust region sub problem (8).
- otherwise, we can always find a $\lambda \in (0, \infty)$ such that $\|\mathbf{p}(\lambda)\| = \Delta_k$.
- perform QR decomposition $\mathbf{J}(\beta) = \mathbf{Q}\begin{pmatrix}\mathbf{R} \\ 0\end{pmatrix}$. Based on the idea in [1], section 10.2, we can economically obtain an upper triangular $\mathbf{R}_\lambda$ from $\mathbf{R}$ such that

$$\mathbf{R}_\lambda^T\mathbf{R}_\lambda = \left(\mathbf{J}^T(\beta)\mathbf{J}(\beta) + \lambda\mathbf{I}\right)$$

# Data and Model Fitting

### Nonlinear Least Square: Regularization and Constraints

- in practice, it is often desired to regularize the model parameters $\beta$ and perhaps impose additional constraints.
- we offer a general nonlinear least square fit method that solves

$$\arg\min_{\beta} \psi(\beta) = \frac{1}{2}\|\mathbf{r}(\beta)\|_2^2 + \lambda\beta^T\mathbf{\Omega}\beta \quad \text{s.t.} \tag{10}$$

$$\mathbf{b}_l \leq \mathbf{A}\beta \leq \mathbf{b}_u \quad \text{and}$$

$$\mathbf{l} \leq \beta \leq \mathbf{u} \tag{11}$$

- $\lambda$ in (10) could be input or computed from GCV.

# Data and Model Fitting

## Curve Fitting

Given a set of data points $(x_i, y_i)$ observed from a unknown function $y = \tilde{f}(x)$, $x_i \in [a, b]$ for $i = 0, \cdots, m-1$, we want to approximates $\tilde{f}(x)$ by a smooth curve $f(x;\theta)$ that is parameterized by $\theta$.

## $f(x;\theta)$ as Linear Combination of Basis

Let $\phi^T(x) = (\phi_0(x), \cdots, \phi_{n-1}(x))$ be a vector of $n$ basis functions of certain form. Let the curve approximation function $f(x;\theta) = \phi(x)^T\theta$ as a linear combination of the basis vector and coefficient vector $\theta^T = (\theta_0, \cdots, \theta_{n-1})$.

# Data and Model Fitting

## Curve Fitting: Choice of Basis

- B-spline polynomials
  - polynomial degree as user input, high degree not recommended, i.e. above 6.
  - knot points placement as user input, usually want data points uniformally distributed to knot point intervals.
- Chebyshev polynomial of the first kind
  - polynomial degree as user input.
  - suitable for the situation $\tilde{f}(x)$ is naturally smooth.

# Data and Model Fitting

## Curve Fitting: Formulation

- construct a basis matrix $\mathbf{\Phi}(x)$ where the $(i,j)$-th element of the matrix is $\Phi_{i,j}(x) = \phi_j(x_i)$. The sum of square of residuals (SSR) is

$$\mathbf{SSR} = (\mathbf{y} - \mathbf{\Phi}(x)\theta)^T (\mathbf{y} - \mathbf{\Phi}(x)\theta)$$

- solve a regularized least square problem,

$$\arg\min_{\theta} (\mathbf{y} - \mathbf{\Phi}(\mathbf{x})\theta)^T (\mathbf{y} - \mathbf{\Phi}(\mathbf{x})\theta) + \lambda\theta^T \mathbf{\Omega}\theta \tag{12}$$

  , where $\mathbf{\Omega}$ in the regularization term penalizes the roughness of curve $f(x)$, penalty factor $\lambda > 0$ could be a user input or computed by GCV.

- alternatively, minimize curve roughness and subject to fitting error constraints.

$$\arg\min_{\theta} \theta^T \mathbf{\Omega}\theta \text{ s.t.} -\epsilon < \mathbf{\Phi}(\mathbf{x})\theta - \mathbf{y} < \epsilon \tag{13}$$

# Data and Model Fitting

## Curve Fitting: Shape Constraints

- at given points $x_k \in [a, b], k = 0, \cdots, K$, additional curve shape constraints may be imposed to formulation (12) and (13).

$$l_k \le f^{(d)}(x_k; \theta) \le u_k, d = 0, 1, 2 \tag{14}$$

  , where $f^{(d)}(x_k; \theta)$ denotes the $d$-th derivative of $f(x)$ wrt $x$.

- shape constraints (14) effectively restrict the function value $f(x_k; \theta)$, slope $f'(x_k; \theta)$ or curvature $f''(x_k; \theta)$ to be bounded within a certain range.
- for example the classic natural cubic spline can be built by imposing $f''(a; \theta) = 0$ and $f''(b; \theta) = 0$.

# Data and Model Fitting

### Curve Fitting: Roughness Measure

- the regularization matrix $\mathbf{\Omega}$ in formulation (12) and (13) is to make the curve function $f(x;\theta)$ "smooth".
- if one chooses $\mathbf{\Omega}$ being the identity matrix, the roughness measure is to reduce the $l_2$ norm of basis coefficients $\|\theta\|_2^2$, which tends to produce a flat curve close to $f(x;\theta) = 0$ as we increase the penalty factor $\lambda \to \infty$.

# Data and Model Fitting

## Curve Fitting: Roughness Measure (derivative based)

- classic definition, i.e. [4] chapter XIV, of the roughness measure of curve function $f(x;\theta)$ over $[a, b]$ is

$$\mathbf{R}(\theta) = \int_a^b f''(x;\theta)^2 dx \tag{15}$$

- we offer 4 levels of derivative-based roughness matrix construction,

$$\mathbf{R}(\theta) = \int_a^b \left(f^{(d)}(x;\theta)\right)^2 dx, d = 0, \cdots, 3$$

- users make the choice on $d$, we internally carry out calculations to write the roughness measure in the form of $\mathbf{R}(\theta) = \theta^T \mathbf{\Omega} \theta$.

# Data and Model Fitting

## Curve Fitting: Roughness Measure (divided difference based)

- the classic roughness measure (15) favorites small magnitude of curvature, but does not have preference over the curvature sign change.
- we offer another set of regularization choices that penalize the divided difference of derivatives,

$$\mathbf{R}(\theta) = \sum_k \left( \frac{f^{(d)}(x_k;\theta) - f^{(d)}(x_{k+1};\theta)}{x_{k+1} - x_k} \right)^2, d = 1, 2$$

  , where the index $k$ traverses through the knot points for B-spline basis and predefined points for Chebyshev basis.
- users only need choose the level of $d$. We internally compute $\mathbf{R}(\theta) = \theta^T \boldsymbol{\Omega} \theta$.

# Data and Model Fitting

## Curve Fitting: Roughness Measure (micro leverage)

- further, users can partition $[a, b]$ into sub intervals $(x_0, x_1, \cdots, x_{k+1})$ and apply different roughness weights in different sub interval.

$$\mathbf{R}(\theta) = \sum_{i=0}^{k} w_i \int_{x_i}^{x_{i+1}} \left(f^{(d)}(x;\theta)\right)^2 dx$$

- the total roughness measure $\mathbf{R}(\theta)$ then is a weighted sum of local roughness. The sub interval weights $\mathbf{w} = \{w_i\}$ play the role as micro leverage factors.
- users set the leverage factors $\mathbf{w}$ through a piecewise constant function $w(x)$.

# Data and Model Fitting

## Curve Fitting: nonlinear model

- design a mathematical model $g(x;\theta) = \phi(x)^T\theta$ as a smooth curve.
- dependent variable $y$ is connected to independent variable $x$ by a known nonlinear mapping function through the model, i.e. $y = f(g(x;\theta), x)$.
- let $\mathbf{r}(\theta)$ be a vector-valued residual functions, where the $i$-th element $r_i(\theta) = f(g(x_i;\theta), x_i) - y_i$ computes the residual error for observation $(x_i, y_i)$ given $\theta$.

# Data and Model Fitting

## Curve Fitting: nonlinear model (continued)

- formulation (12) changes to the following regularized nonlinear least square problem,

$$\arg\min_{\theta} |\mathbf{r}(\theta)|_2^2 + \lambda\theta^T \mathbf{\Omega}\theta \tag{16}$$

- accordingly, formulation (13) now becomes to an nonlinear programming problem,

$$\arg\min_{\theta} \theta^T \mathbf{\Omega}\theta \text{ s.t.}$$

$$-\epsilon_i < r_i(\theta) < \epsilon_i \forall i = 0, \cdots, m-1 \tag{17}$$

- additional curve shape constraint (14) may be imposed to both formulation (16) and (17).

# Data and Model Fitting

## Surface Fitting

Given $N$ number of data points $(x_k, y_k, z_k)$, $k = 0, \cdots, N-1$, which are observed from an unknown 2-dimensional function $z = \tilde{f}(x, y)$ defined in domain $[a, b] \times [c, d]$, we want to approximate $\tilde{f}(x, y)$ by a smooth surface function $f(x, y; \mathbf{\Theta})$.

## $f(x, y; \mathbf{\Theta})$ as Tensor Product of Basis

written as the tensor product of two sets of basis functions $\phi(x)$ and $\psi(y)$,

$$f(x, y; \mathbf{\Theta}) = \phi(x)^T \mathbf{\Theta} \psi(y) \tag{18}$$

, where $\mathbf{\Theta}$ is the coefficient matrix of the tensor product. The objective of surface fitting is to recover $\mathbf{\Theta}$ from observed data points.

## Surface Fitting: Choice of Basis

- basis functions $\phi(x)$ and $\psi(y)$ are of the same type.
- class SmoothSplineSurface uses B-spline as basis.
- class SmoothBilinearSurface uses piecewise linear functions as basis.
- class SmoothChebyshevSurface uses Chebyshev polynomial (first kind) as basis.

# Data and Model Fitting

## Surface Fitting: Formulation

- regularized least square fit,

$$\arg\min_{\mathbf{\Theta}} \sum_{k=0}^{N-1} (z_k - f(\mathbf{\Theta}; x_k, y_k))^2 + \lambda \mathbf{R}(\mathbf{\Theta})) \tag{19}$$

  , where $\mathbf{R}(\mathbf{\Theta})$ is some choice of roughness regularization. $\lambda > 0$ is the roughness penalty factor from user input or computed by GCV method.

- alternatively, directly minimize the roughness measure subject to bounded fitting error constraints.

$$\arg\min_{\mathbf{\Theta}} \mathbf{R}(\mathbf{\Theta}) \qquad s.t. \qquad -\epsilon < f(\mathbf{\Theta}; x_k, y_k) - z_k < \epsilon, \forall k \tag{20}$$

# Data and Model Fitting

## Surface Fitting: Shape Constraints

In both formulation (19) and (20), it is possible to further impose constraints to given points $(x_k, y_k)$. The supported constraint types are:

- Function value $f(x_k, y_k, \mathbf{\Theta})$ is bounded;
- Partial delta $\frac{\partial}{\partial x} f(x_k, y_k, \mathbf{\Theta})$ or $\frac{\partial}{\partial y} f(x_k, y_k, \mathbf{\Theta})$ is bounded;
- Gamma $\frac{\partial^2}{\partial^2 x} f(x_k, y_k, \mathbf{\Theta})$ or $\frac{\partial^2}{\partial^2 y} f(x_k, y_k, \mathbf{\Theta})$ is bounded;
- Cross gamma $\frac{\partial^2}{\partial x \partial y} f(x_k, y_k, \mathbf{\Theta})$ is bounded;

### Surface Fitting: Shape Constraints (continued)

Let $\delta^{(r,s)}(f(x, y; \mathbf{\Theta}))$ be the derivative operator relevant to the supported constraint types, i.e. $\delta^{(0,0)}$ for function value, $\delta^{(1,0)}$ for partial delta $\frac{\partial}{\partial x}$. The optimization problem (19) and (20) may be subject to additional surface shape constraints,

$$l_k \leq \delta^{(r_k,s_k)}(f(x_k, y_k; \mathbf{\Theta})) \leq u_k, \forall k \tag{21}$$

# Data and Model Fitting

## Surface Fitting: Roughness Measure

- Frobenius norm, equivalent to the $l_2$ norm in curve fitting.
- Dirichlet energy, defined as the square integral of the gradient norm,

$$\mathbf{R}(\boldsymbol{\Theta}) = \int_a^b \int_c^d \|\nabla f(x, y; \boldsymbol{\Theta})\|^2 dydx$$

- Thin-plate energy, a rotation-invariant measure defined as $\mathbf{R}(\boldsymbol{\Theta}) =$

$$\int_a^b \int_c^d \left(\nabla_{xx} f(x, y; \boldsymbol{\Theta})^2 + 2\nabla_{xy} f(x, y; \boldsymbol{\Theta})^2 + \nabla_{yy} f(x, y; \boldsymbol{\Theta})^2\right) dydx$$

# Data and Model Fitting

## Surface Fitting: Roughness Measure (micro leverage)

- Users can partition the whole surface domain $[a, b] \times [c, d]$ into sub areas $[x_i, x_{i+1}] \times [y_j, y_{j+1}]$, $i = 0, \cdots, k$, $j = 0, \cdots, l$ and supply a 2-d piecewise constant function $w(x, y)$ to surface fitting.
- Internally, we compute the total roughness measure as the weighted sum of roughness over those sub surface areas.

$$\mathbf{R}(\mathbf{\Theta}; a, b, c, d) = \sum_{i=0}^{k} \sum_{j=0}^{l} w_{i,j} \mathbf{R}(\mathbf{\Theta}; x_i, x_{i+1}, y_j, y_{j+1})$$

, where $w_{i,j} = w(x, y), \forall (x, y) \in [x_i, x_{i+1}] \times [y_j, y_{j+1}]$.

# Data and Model Fitting

### Surface Fitting: nonlinear model

- design a model $g(x, y; \mathbf{\Theta})$ as a smooth surface.
- a known 2-d nonlinear mapping function $z = f(g(x, y; \mathbf{\Theta}), x, y)$ connects independent variables $(x, y)$ to function value $z$ through the model.
- let $\mathbf{r}(\mathbf{\Theta})$ be the vector of residual functions where the $i$-th element $r_i(\mathbf{\Theta}) = f(g(x_i, y_i; \mathbf{\Theta}), x_i, y_i) - z_i$ is the residual error for observation $(x_i, y_i, z_i)$.

# Data and Model Fitting

## Surface Fitting: nonlinear model (continued)

Users can choose to calibrate surface model parameter $\mathbf{\Theta}$ by

- a regularized nonlinear least square fit,

$$\arg\min_{\mathbf{\Theta}} |\mathbf{r}(\mathbf{\Theta})|_2^2 + \lambda \mathbf{R}(\mathbf{\Theta}) \tag{22}$$

- or by solving an nonlinear programming problem,

$$\arg\min_{\mathbf{\Theta}} \mathbf{R}(\mathbf{\Theta}) \qquad \text{s.t.}$$

$$-\epsilon_i < r_i(\mathbf{\Theta}) < \epsilon_i \quad \forall i = 0, \cdots, m-1 \tag{23}$$

Again, in both formulation (22) and (23), additional shape constraints (21) can be imposed.

# References I


[1] Jorge Nocedal and Stephen J. Wright: Numerical Optimization, Springer, 1999

[2] Philip E. Gill, Walter Murray and Margaret H. Wright: Practical Optimization, Academic Press, 1981

[3] Trevor Hastie, Robert Tibshirani and Jerome Friedman: The Elements of Statistical Learning, Springer, 2001

[4] Carl de Boor: A Practical Guide to Splines, Revised Edition, Applied Mathematical Sciences Volume 27, Springer, 2001

[5] John P. Boyd: Chebyshev and Fourier Spectral Methods, second edition (revised), Dover Publications, 2001