Lecture 11:

Variational Methods II: Discrete Aspects

Contents

When Should We Discretise?

Strategy 1: Discretise the Euler-Lagrange Equation

- Variational restoration of a 1-D signal $f : [a, b] \to \mathbb{R}$ searches a minimiser $u(x)$ of

$$E(u) := \int_a^b (u - f)^2 \, dx + \alpha \int_a^b \Psi\left(u_x^2\right) \, dx$$

- Lecture 10: Minimiser satisfies necessarily to the Euler–Lagrange equation

$$0 \;=\; u - f - \alpha \, \partial_x \left(\Psi'(u_x^2)\, u_x\right)$$

with reflecting (homogeneous Neumann) boundary conditions:

$$u_x = 0 \qquad \text{for } x = a \text{ or } x = b.$$

When Should We Discretise? (2)

How Do We Discretise the Euler-Lagrange Equation?

- Consider a grid size $h := \frac{b-a}{N}$ and grid points $x_i := (i - \frac{1}{2})\, h$ with $i = 1,...,N$. Let u_i denote an approximation to $u(x_i)$.
- For some inner pixel $(i = 2, ..., N-1)$ we obtain

$$\begin{aligned} 0 \;=\; u_i - f_i \quad &- \quad \alpha\, \Psi'\left(\frac{(u_{i+1}-u_i)^2}{h^2}\right)\frac{u_{i+1}-u_i}{h^2} \\ &+ \quad \alpha \Psi'\left(\frac{(u_i-u_{i-1})^2}{h^2}\right)\frac{u_i-u_{i-1}}{h^2} \end{aligned}$$

- This formula may even be extended to the boundary pixels $i = 1$ and $i = N$: Taking into account the reflecting boundary conditions by means of dummy pixels $u_0 := u_1$ and $u_{N+1} := u_N$ yields

$$0 \;=\; u_1 - f_1 \quad - \quad \alpha\, \Psi'\left(\frac{(u_2-u_1)^2}{h^2}\right)\frac{u_2-u_1}{h^2}$$

$$0 \;=\; u_N - f_N \quad + \quad \alpha\, \Psi'\left(\frac{(u_N-u_{N-1})^2}{h^2}\right)\frac{u_N-u_{N-1}}{h^2}$$

When Should We Discretise? (3)

- In matrix–vector notation, the entire system reads

$$0 \;=\; \mathbf{u} - \mathbf{f} - \alpha A(\mathbf{u})\, \mathbf{u}$$

where $\mathbf{u} = (u_1, ..., u_N)^\top$, and the $N \times N$ matrix $A(\mathbf{u}) = (a_{kl}(\mathbf{u}))$ satisfies

$$a_{kl} \;:=\; \begin{cases} \frac{\Psi'_{(k+l)/2}}{h^2} & (l \in \mathcal{N}(k)), \\ -\sum\limits_{n \in \mathcal{N}(k)} \frac{\Psi'_{(k+n)/2}}{h^2} & (l = k), \\ 0 & (\text{else}). \end{cases},$$

$\mathcal{N}(k)$ are the two neighbours of pixel k (boundary pixels have one neighbour), and

$$\Psi'_{(k+l)/2} := \Psi'\left(\frac{(u_k-u_l)^2}{h^2}\right).$$

- Thus, we have to solve the nonlinear system of equations

$$(I - \alpha A(\mathbf{u}))\, \mathbf{u} \;=\; \mathbf{f}$$

with a symmetric matrix $A(\mathbf{u})$.

When Should We Discretise? (4)

Strategy 2: Discretise the Energy Functional

- Let us directly discretise the energy functional

$$E(u) := \int_a^b (u-f)^2\,dx + \alpha \int_a^b \Psi\left(u_x^2\right)\,dx.$$

- Restoration of a discrete signal $\mathbf{f} = (f_1, ..., f_N)^\top$ searches for a minimiser $\mathbf{u} = (u_1, ..., u_N)^\top$ of

$$E(\mathbf{u}) := \sum_{i=1}^{N}(u_i - f_i)^2 + \alpha \sum_{k=1}^{N-1} \Psi\left(\frac{(u_{k+1}-u_k)^2}{h^2}\right)$$

- In a minimiser $\mathbf{u} = (u_1, ..., u_N)^\top$, we necessarily have a vanishing gradient of E:

$$\frac{\partial E}{\partial u_i} = 0 \qquad \text{for } i = 1,...,N$$

In short: $\nabla_{\mathbf{u}} E = 0$ with $\nabla_{\mathbf{u}} := (\partial_{u_1}, ..., \partial_{u_N})^\top$.

When Should We Discretise? (5)

- Setting $\frac{\partial E}{\partial u_i} = 0$ for $i = 1,...,N$ gives

$$\begin{aligned}
0 &= u_1 - f_1 &&- \alpha\,\Psi'\left(\frac{(u_2-u_1)^2}{h^2}\right)\frac{u_2-u_1}{h^2} \\
0 &= u_i - f_i &&- \alpha\Psi'\left(\frac{(u_{i+1}-u_i)^2}{h^2}\right)\frac{u_{i+1}-u_i}{h^2} \\
&&&+ \alpha\Psi'\left(\frac{(u_i-u_{i-1})^2}{h^2}\right)\frac{u_i-u_{i-1}}{h^2} \\
&&&\quad (i = 2, ..., N-1), \\
0 &= u_N - f_N &&+ \alpha\Psi'\left(\frac{(u_N-u_{N-1})^2}{h^2}\right)\frac{u_N-u_{N-1}}{h^2}
\end{aligned}$$

- same nonlinear system as before, but without using the Euler-Lagrange equation
- Discretising the energy functional instead of the Euler-Lagrange equation can be particularly useful if you have difficulties with discretising the boundary conditions of if you want to prove stability results.

Numerical Methods for the Elliptic Problem

- We have to solve the nonlinear system

$$(I - \alpha A(\mathbf{u}))\,\mathbf{u} \;=\; \mathbf{f}.$$

- Iterative method:
  - Let $\mathbf{u}^k$ be the result of the k-th iteration.
  - Initialise with $\mathbf{u}^0 := \mathbf{f}$.
  - Replace the nonlinear system by a sequence of linear problems:

$$(I - \alpha A(\mathbf{u}^k))\,\mathbf{u}^{k+1} \;=\; \mathbf{f} \qquad (k = 1, 2, ...)$$

- This method is called *Kačanov method* or *method with time-lagged diffusivity*.
- Formally it may be regarded as a fixed point iteration of

$$\mathbf{u} \;=\; (I - \alpha A(\mathbf{u}))^{-1}\mathbf{f}.$$

How Are the Linear Systems Solved?

- The 1-D case is simple:
  It leads to tridiagonal systems that can be solved in linear complexity with the Thomas algorithm from Lecture 6.
- For dimensions ≥ 2, it is significantly more difficult to find efficient algorithms.
- $A(\mathbf{u})$ remains sparse: In standard pixel ordering one obtains
  - a pentadiagonal system in 2-D,
  - a heptadiagonal system in 3-D.

  However, the bandwidth gets large: many zero entries between the nonvanishing entries.
- direct algorithms fill in zeros within band
  $\Longrightarrow$ prohibitive storage and computational effort
- Let us now study some classical iterative algorithms.

Numerical Methods for the Elliptic Problem (3)

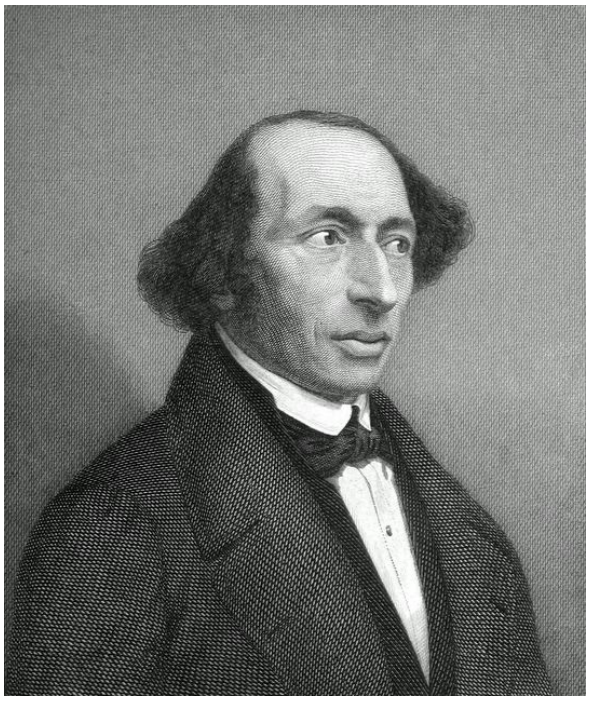
Classical Iterative Algorithms for Linear Systems

- *Given:* large linear system $B\mathbf{x} = \mathbf{c}$ with a sparse system matrix $B \in \mathbb{R}^{N\times N}$.
- *Goal:* find iterative method that hardly requires additional storage and that converges reasonably fast
- *General Strategy:*
  - splitting of B such that
    $$B = S + T$$
    and S is a matrix that can be inverted easily, e.g. a diagonal or triangular matrix
  - Now $B\mathbf{x} = \mathbf{c}$ can be rewritten as
    $$S\mathbf{x} = -T\mathbf{x} + \mathbf{c}.$$
  - This is solved iteratively by choosing some initialisation $\mathbf{x}^0$ and using
    $$S\mathbf{x}^{k+1} = -T\mathbf{x}^k + \mathbf{c} \qquad (k = 1, 2, ...).$$

Numerical Methods for the Elliptic Problem (4)

Example 1: Jacobi Method (Gesamtschrittverfahren)

- Let $B = D + L + R$ where D is a diagonal matrix, L a strictly lower triangular matrix and R a strictly upper triangular matrix.
- In the Jacobi method one chooses $S := D$ and $T := L + R$. This gives
  $$D\mathbf{x}^{k+1} = -(L + R)\mathbf{x}^k + \mathbf{c} \qquad (k = 1, 2, ...)$$
- This diagonal system can be solved directly:
  $$x_i^{(k+1)} = \frac{1}{b_{i,i}}\Big(- \sum_{j,j\neq i} b_{i,j}\, x_j^k + c_i\Big) \qquad (i = 1, ..., N)$$
- The Jacobi method is relatively slow, but well-suited for parallel computing.



Carl Gustav Jacob Jacobi (1804-1851) was a very hard-working mathematician. Not only many mathematical concept are named after him, but also a lunar crater. Images taken from `http://www.portrait.kaar.at/Stahlstiche%203/image22.html` and `http://de.wikipedia.org/wiki/Bild:Carl_Jacobi.jpg`.

Example 2: Gauß–Seidel Method (Einzelschrittverfahren)

- Based on the splitting $S := D + L$ and $T := R$.
- Requires to solve the lower triangular system

$$(D + L)\mathbf{x}^{k+1} = -R\mathbf{x}^k + \mathbf{c} \qquad (k = 1, 2, ...).$$

- This can be implemented by a simple forward substitution:

$$x_i^{k+1} = \frac{1}{b_{i,i}}\Big(-\sum_{j=1}^{i-1} b_{i,j}\, x_j^{k+1} - \sum_{j=i+1}^{n} b_{i,j}\, x_j^k + c_i \Big) \qquad (i = 1, ..., N).$$

- Thus, new values are used immediately once they are computed.
- Typically the Gauß–Seidel method requires about half as many iterations to reach the same accuracy as the Jacobi method.
- well-suited for sequential computing

Left: Carl–Friedrich Gauß (1777–1855) was not only one of the greatest mathematicians of all times, but contributed also to astronomy, geodesy, and physics. Image taken from `http://de.wikipedia.org/wiki/Bild:Carl_Friedrich_Gauss.jpg`. **Right:** Philipp Ludwig Ritter von Seidel (1821–1896) was born in Zweibrücken and studied under Carl Gustav Jacob Jacobi. Image from `http://www.badw.de/bilder/BAdW_Portraets_Web/Mitglieder/Seidel_063.jpg`

Example 3: Successice Overrelaxation (SOR) Method (Relaxationsverfahren)

- variant of the Gauß-Seidel method that attempts to achieve an accelerated convergence by extrapolation
- Let $\tilde{x}_i^{k+1}$ denote the solution of one Gauß-Seidel step:

$$\tilde{x}_i^{k+1} = \frac{1}{b_{i,i}}\Big(- \sum_{j=1}^{i-1} b_{i,j}\, x_j^{k+1} - \sum_{j=i+1}^{n} b_{i,j}\, x_j^{k} + c_i \Big)$$

  Then the SOR method replaces it by its extrapolation

$$x_i^{k+1} = x_i^k + \omega\,(\tilde{x}_i^{k+1} - x_i^k)$$

  with some so-called relaxation parameter $\omega \in (1, 2)$.
- can be shown to correspond to the splitting $S := \frac{1}{\omega} D + L$ and $T := \left(1 - \frac{1}{\omega}\right) D + R$.
- For large systems of equations, one often experiences a speed up by one order of magnitude, if an appropriate value for ω is chosen.
- very good compromise between ease of implementation and convergence speed

When Do These Methods Converge ?

- If the system matrix B is *strictly diagonally dominant*, i.e.

$$|b_{i,i}| > \sum_{j,j\neq i} |b_{i,j}| \qquad \forall\ i$$

  then one can show that the methods of Jacobi and Gauß–Seidel converge.

- For positive definite (or negative definite) matrices, the SOR method converges for $\omega \in (0, 2)$. In particular, this implies convergence of the Gauß–Seidel method.

- For the regularisation methods we consider, these requirements are satisfied.

More Advanced Numerical Methods

- faster convergence, but more difficult to implement
- two important classes:
  - preconditioned conjugate gradient methods (PCG)
  - multigrid methods

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

The Gradient Descent Method

Discrete Case

- A minimiser of a discrete energy function $E(\mathbf{u})$ with $\mathbf{u} \in \mathbb{R}^N$ satisfies

$$\nabla_{\mathbf{u}} E = 0.$$

- Since $\nabla_{\mathbf{u}} E$ points in the direction of steepest ascent, one can find a minimiser as steady state ($t \to \infty$) of an evolution equation that slides downhill:

$$\partial_t \mathbf{u} = -\gamma\, \nabla_{\mathbf{u}} E$$

  with an arbitrary speed factor $\gamma > 0$.

- This method is called *gradient descent* or *steepest descent*.

- For strictly convex functions, the minimiser is unique and the method converges globally (i.e. for arbitrary initialisations).

Continuous Case

- A minimising function of a continuous energy functional

$$E(u) := \int_\Omega F(x_1, x_2, u, u_{x_1}, u_{x_2})\, d\mathbf{x}$$

  satisfies also $\nabla_u E = 0$ when we interprete the terms in the Euler-Lagrange equation as

$$\nabla_u E \;=\; F_u - \partial_{x_1} F_{u_{x_1}} - \partial_{x_2} F_{u_{x_2}}.$$

- One can show that this is correct in the sense of so-called Gâteaux derivatives, a generalisation of directional derivatives to function spaces.

- A corresponding gradient descent is given by

$$\partial_t u = -\gamma\, \nabla_u E$$

- strict convexity yields unique minimiser and global convergence

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

Numerical Methods for the Parabolic Problem

An Alternative Approach for Minimising the Energy Functional

- The minimiser of the energy functional

$$E_f(u) := \int_\Omega \Big((u-f)^2 + \;\alpha\, \Psi(|\nabla u|^2) \Big)\, d\mathbf{x}$$

  satisfies necessarily the Euler–Lagrange equation

$$0 \;=\; -\mathrm{div}\,(\Psi'(|\nabla u|^2)\, \nabla u) \;+\; \frac{u-f}{\alpha}$$

  It is an elliptic partial differential equation.

- The corresponding gradient descend searches for the steady state ($t \to \infty$) of the parabolic PDE

$$\frac{\partial u}{\partial t} \;=\; \mathrm{div}\,(\Psi'(|\nabla u|^2)\, \nabla u) \;-\; \frac{u-f}{\alpha}$$

Numerical Methods for the Parabolic Problem (2)

- This is a *diffusion–reaction equation*. In contrast to pure diffusion equations, the reaction term (bias term, similarity term, fidelity term) $\frac{u-f}{\alpha}$ creates nonflat steady states.
- The "time" t is now a pure numerical parameter. The goal is to reach the nontrivial steady state $t \to \infty$ as quickly as possible.
- So let us now investigate numerical schemes for solving

$$\frac{\partial u}{\partial t} = \sum_{l=1}^{m} \partial_{x_l} \left(\Psi'(|\nabla u|^2)\, u_{x_l} \right) - \frac{u-f}{\alpha}.$$

Numerical Methods for the Parabolic Problem (3)

Modified Explicit Scheme

- Explicit approximation (implicit in bias term):

$$\frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\tau} = \sum_{l=1}^{m} A_l^k\, \mathbf{u}^k + \frac{1}{\alpha}(\mathbf{f} - \mathbf{u}^{k+1}).$$

- can be solved directly (explicitly) for unknown $\mathbf{u}^{k+1}$:

$$\mathbf{u}^{k+1} = \frac{\alpha}{\alpha+\tau} \left(I + \tau \sum_{l=1}^{m} A_l^k \right) \mathbf{u}^k + \frac{\tau}{\alpha+\tau}\mathbf{f}.$$

- typical stability limit $(m = 2, \quad h_1 = h_2 = 1, \quad 0 \le \Psi'(s^2) \le 1)$:

$$\tau \le 0.25$$

- same stability limit as explicit scheme for unbiased diffusion

Numerical Methods for the Parabolic Problem (4)

Semi-Implicit Scheme

- Semi-implicit approximation:

$$\frac{\mathbf{u}^{k+1}-\mathbf{u}^k}{\tau} = \sum_{l=1}^{m} A_l^k \mathbf{u}^{k+1} + \frac{1}{\alpha}(\mathbf{f}-\mathbf{u}^{k+1}).$$

- absolutely stable, but requires to solve the linear system

$$\left(I - \frac{\alpha\tau}{\alpha+\tau}\sum_{l=1}^{m} A_l^k\right)\mathbf{u}^{k+1} = \frac{\alpha\mathbf{u}^k+\tau\mathbf{f}}{\alpha+\tau}.$$

- appropriate solvers include Jacobi, Gauß-Seidel, SOR.

Numerical Methods for the Parabolic Problem (5)

AOS Scheme

- replaces semi-implicit scheme

$$\mathbf{u}^{k+1} = \left(I - \frac{\alpha\tau}{\alpha+\tau}\sum_{l=1}^{m} A_l^k\right)^{-1} \frac{\alpha\mathbf{u}^k+\tau\mathbf{f}}{\alpha+\tau}$$

  by the absolutely stable additive operator splitting

$$\mathbf{u}^{k+1} = \frac{1}{m}\sum_{l=1}^{m}\left(I - m\frac{\alpha\tau}{\alpha+\tau}A_l^k\right)^{-1}\frac{\alpha\mathbf{u}^k+\tau\mathbf{f}}{\alpha+\tau}.$$

- leads to easily solvable tridiagonal systems (Thomas algorithm, Lecture 6)
- drawback: for $t \to \infty$, one must reduce time step size to avoid splitting errors

Summary

- For variational methods, one can either discretise the Euler-Lagrange equation or the energy functional.
- Nonquadratic regularisers lead to nonlinear systems of equations. Quadratic regularisers create linear systems of equations.
- In the elliptic setting they can be solved using Kačanov's method, i.e.
  - an outer fixed point iteration for the nonlinearity
  - inner iterations for solving the linear systems (e.g. Jacobi, Gauß-Seidel, SOR)
- Gradient descent leads to the parabolic setting where one regards the Euler-Langrange equation as steady state of a diffusion–reaction equation.
- This equation can be solved with the usual numerical methods for diffusion filtering: (modified) explicit, semi-implicit or AOS schemes

References

- D. M. Young: *Iterative Solution of Large Linear Systems*. Dover, New York, 2003.
  (the most detailed monograph on classical iterative methods for linear systems of equations)
- Y. Saad: *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, Second Edition, 2003. (first edition can be downloaded from `http://www-users.cs.umn.edu/~saad/books.html`)
  (one of the best books on conjugate gradient methods)
- W. L. Briggs, V. E. Henson, S. F. McCormick: *A Multigrid Tutorial*. SIAM, Philadelphia, Second Edition, 2000.
  (excellent introduction to multigrid methods)
- J. Weickert, J. Heers, C. Schnörr, K. J. Zuiderveld, O. Scherzer, H. S. Stiehl: Fast parallel algorithms for a broad class of nonlinear variational diffusion approaches. *Real-Time Imaging*, Vol. 7, No. 1, 31–45, Feb. 2001.
  `(http://www.mia.uni-saarland.de/weickert/publications.shtml)`
  (describes both elliptic and parabolic approaches).

Assignment C3 – Classroom Work

Problem 1 (Stencils and Discrete Diffusion Properties)

Show that edge-enhancing anisotropic diffusion filtering may be written as

$$\partial_t u \;=\; \text{div}\left(g(\nabla u_\sigma \nabla u_\sigma^T)\nabla u\right) \;. \tag{1}$$

Assume that g can be represented by a power series with sufficiently large radius of convergence. Investigate the eigenvectors and eigenvalues of powers $(\nabla u_\sigma \nabla u_\sigma^T)^k$ for $k \in \mathbb{N}$. What are the eigenvectors and eigenvalues of $g(\nabla u_\sigma \nabla u_\sigma^T)$?

What is the effect of using anisotropic diffusion (1) without presmoothing, i. e. with $\sigma = 0$? Is the resulting filter already known from the lecture?

Problem 2 (Discrete Variational Methods)

For a real matrix $D \in \mathbb{R}^{M\times N}$, $M \le N$, and a vector $\mathbf{u} := (u_1, \dots, u_N)^T \in \mathbb{R}^N$, let

$$(D\mathbf{u})_k \;:=\; \sum_{j=1}^{N} D_{kj} u_j \qquad k = 1, \dots, M$$

denote the k-th component of the vector product $D\mathbf{u}$. For example, D can contain the coefficients for a pointwise derivative approximation with finite differences.

Consider the discrete energy function

$$E(\mathbf{u}) \;=\; \sum_{i=1}^{N}(u_i - f_i)^2 + \alpha \sum_{k=1}^{M} \Psi\left(((D\mathbf{u})_k)^2\right) \;.$$

(a) Calculate the gradient $\nabla_{\mathbf{u}} E$ in order to obtain a necessary condition for a minimiser $\mathbf{u}$ of E.

(b) Consider the corresponding method with time-lagged diffusivity to calculate a minimiser. Prove that the matrices of the corresponding linear system appearing there are positive definite and thus can be solved with successive overrelaxation (SOR).

Hint: Write the gradient calculated in part (a) as

$$\frac{1}{2}\nabla_{\mathbf{u}} E \;=\; \mathbf{u} - \mathbf{f} - \alpha A(\mathbf{u})\mathbf{u}$$

where A is the product of three matrices.