# NUMERICAL PROGRAMMING 2 (CSE)
# SUMMER TERM 2015

CAROLINE LASSER

Contents



*Date*: updated July 13, 2015.

## References

[DB] Peter Deuflhard, Folkmar Bornemann, Scientific computing with ordinary differential equations, Springer, 2002

[DW] Peter Deuflhard, Martin Weiser, Adaptive numerical solution of pdes, DeGruyter, 2012

[HW] Ernst Hairer, Gerhard Wanner, Solving ordinary differential equations II, 2nd edition, Springer, 1996

[Is] Arieh Iserles, A first course in the numerical analysis of differential equations, 2nd edition, Cambridge University Press, 2009

[LR] Ben Leimkuhler, Sebastian Reich, Simulating Hamiltonian dynamics, Cambridge University Press, 2004

[LeV] Randall Leveque, Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems, SIAM, 2007

[Le] Randall LeVeque, Finite-volume methods for hyperbolic problems, Cambridge University Press, 2002

[Lu] Christian Lubich, From quantum to classical molecular dynamics: reduced models and numerical analysis. European Math. Soc., 2008

[Tr] John Trangenstein, Numerical solution of hyperbolic conservation laws, Cambridge University Press, 2008

# 1. Hamiltonian dynamics

## 1.1. **The Störmer–Verlet method (13.04.)**

1.1.1. *Harmonic oscillator.* Rewrite the harmonic oscillator

$$\ddot{q} = -q, \qquad q(0) = q_0, \qquad \dot{q}(0) = p_0$$

in Hamiltonian form. Prove energy conservation, and observe that `ode45` performs poorly in that respect.

1.1.2. *Energy conservation.* Prove energy conservation for a Hamiltonian system

$$\dot{z} = \Omega \nabla H(z), \qquad z(0) = z_0.$$

1.1.3. *Störmer–Verlet method.* Check that Störmer–Verlet performs nicely for the harmonic oscillator's energy.

1.1.4. *Hamiltonian flows.* Define the flow $\phi_{t,H} : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ for a Hamiltonian system.

1.1.5. *Splitting methods.* Check that one time step of the Störmer–Verlet method equals

$$\phi_{h/2,T} \circ \phi_{h,V} \circ \phi_{h/2,T}.$$

1.1.6. *Recommended reading.* [LR, Chapter 3 & 4]

## 1.2. **Symplecticity (15.04.)**

1.2.1. *Trotter splitting.* Use Taylor expansion to prove that the Trotter splitting

$$\phi_{t,f_1} \circ \phi_{t,f_2}$$

is a first order method for $\dot{y} = f(y)$ with $f = f_1 + f_2$.

1.2.2. *Flow map of the harmonic oscillator.* Check that the harmonic oscillator's flow map is the multiplication by a rotation matrix $R_t$. Check that the rotation matrix is symplectic, that is,

$$R_t^T \Omega R_t = \Omega, \qquad t \in \mathbb{R}.$$

1.2.3. *Variational equation.* Derive the variational equation for the Jacobian matrix $D_z\phi_{t,H}(z) \in \mathbb{R}^{2d\times 2d}$ of a Hamiltonian flow and prove that it is a symplectic matrix.

1.2.4. *Exponential accuracy.* Discuss exponential approximation properties of the Störmer–Verlet map.

1.2.5. *Recommended reading.* [LR, Chapter 4 & 5]

## 1.3. **Higher order methods (20.04.)**

1.3.1. *The reduced Kepler problem.* The effective energy of a planet with distance $r > 0$ to the sun, that is fixed at the origin, is given by

$$H(r, p) = \tfrac{1}{2}p^2 + \tfrac{l}{2r^2} - \tfrac{m}{r}.$$

1.3.2. *Composition methods.* One chooses suitable weights $w_1, \dots, w_s \in \mathbb{R}$ so that the composition method

$$\widetilde{\psi}_h := \psi_{w_s h} \circ \cdots \circ \psi_{w_1 h}$$

is a higher order method than its buildung block $\psi_h$.

1.3.3. *Consistency.* A composition method has at least the order of its building block, if the consistency condition $w_1 + \cdots + w_s = 1$ holds.

1.3.4. *Symmetry.* A method $\psi_h$ is called symmetric, if it coincides with its adjoint

$$\psi_h^* := (\psi_{-h})^{-1}.$$

The adjoint of the explicit Euler method is the implicit Euler method. The Störmer–Verlet method is symmetric. A composition method with symmetric building block is symmetric, if $w_j = w_{s+1-j}$ for all $j = 1, \dots, s$.

1.3.5. *Recommended reading.* [LR, Chapter 6]

## 2. Partial differential equations

### 2.1. **Examples (22.04.)**

2.1.1. *The setup.* We consider partial differential equations on some domain $\Omega \subset \mathbb{R}^d$ for $d = 1$ or $d = 2$ subject to boundary (and initial conditions) so that we have a well-defined unique solution. Typical boundary conditions are Dirichlet, Neumann and Robin.

2.1.2. *Poisson equation.* For given right handside $f : \Omega \to \mathbb{R}$ we solve

$$\Delta u = f \quad \text{on} \quad \Omega$$

subject to boundary conditions. The homogeneous Poisson equation ($f = 0$) is called Laplace equation.

2.1.3. *Diffusion equation.* For given initial condition $\phi : \Omega \to \mathbb{R}$ we solve

$$u_t = \Delta u \quad \text{on} \quad \Omega$$

subject to $u(\cdot, 0) = \phi$ on $\Omega$ and boundary conditions for $t \in ]0, T[$. Examine the case $\Omega = ]0, \pi[$ with homogeneous Dirichlet conditions (cooling of a stick) by a Fourier ansatz.

2.1.4. *Wave equation.* For given initial conditions $\phi, \psi : \Omega \to \mathbb{R}$ we solve

$$u_{tt} - \Delta u = 0 \quad \text{on} \quad \Omega$$

subject to $u(\cdot, 0) = \phi$ and $u_t(\cdot, 0) = \psi$ on $\Omega$ as well as boundary conditions for $t \in ]0, T[$. Examine the case $\Omega = ]0, \pi[$ with homogeneous Dirichlet conditions by a Fourier ansatz and detect the characteristics.

2.1.5. *Recommended reading.* [DW, Chapter 1–2]

2.2. **Galerkin methods (27.04.05.)** The lecture has been substituted by Giulio Trigila. He has covered [DW, Chapter 1.6 & 4.1.1–4.1.2].

2.2.1. *Recommended reading.* [DW, Chapter 4]

## 2.3. **Finite differences (04.05.)**

2.3.1. *First derivatives.* The one-sided differences

$$D_+u(x) = \tfrac{1}{h}(u(x+h) - u(x)) \quad \text{and} \quad D_-u(x) = \tfrac{1}{h}(u(x) - u(x-h))$$

are first order accuracte, while the centered difference

$$D_0u(x) = \tfrac{1}{2h}(u(x+h) - u(x-h))$$

is second order accurate. Be aware of cancellation for small $h$.

2.3.2. *Method of undetermined coefficients.* As an example, determine $a, b, c \in \mathbb{R}$ so that $au(x) + bu(x-h) + cu(x-2h)$ is a second order approximation to $u'(x)$. After Taylor expansion, we obtain $a = \frac{3}{2h}$, $b = -\frac{2}{h}$, $c = \frac{1}{2h}$ as the unique solution to a linear system.

2.3.3. *Second derivates.* Discuss the centered difference approximation

$$D_0^2u(x) = \tfrac{1}{h^2}(u(x+h) - 2u(x) + u(x-h))$$

as second order approximation of $u''(x)$.

2.3.4. *Third derivatives.* Discuss $D_+D_0^2u(x)$ and $D_0D_0^2u(x)$ as first and second order approximations to $u'''(x)$.

2.3.5. *Recommended reading.* [Is, II.8], [LeV, Ch. 1]

## 3. Elliptic equations

### 3.1. **Finite differences (06.05.)**

3.1.1. *The standard setup.* For the Poisson problem

$$-\Delta u = f \quad \text{in} \quad \Omega = ]0,1[^2, \quad u\,|_{\partial\Omega} = 0$$

we discretise the domain equidistantly with $(x_i, y_j) = (ih, jh)$, $1 \le i, j \le M$ where $hM = 1$, und use the discrete 5-point Laplacian

$$\Delta_h u_h(x_i, y_j) = \frac{1}{h^2}(u_{i+1,j} + u_{i-1,j} - 4u_{ij} + u_{i,j+1} + u_{i,j-1}), \qquad u_{ij} = u_h(x_i, y_j).$$

We obtain a $N \times N$ linear system $A_h u_h = f_h$ with $N = (M-1)^2$.

3.1.2. *Numbering.* Depending on the numbering of the grid points, one obtains different stiffness matrices $A_h$. Discuss natural (row-wise) and checkerboard numbering.

3.1.3. *Nonuniform grids.* Discuss the Pearson discretisation of

$$-(\sigma(x)u'(x))' = f \quad \text{on} \quad ]0,1[, \qquad u_{\partial\Omega} = 0$$

for an arbitrary grid. Symmetry of the stiffness matrix enforces an equidistant grid.

3.1.4. *Recommended reading.* [DW, Chapter 3]

3.2. **Elementary finite elements (13.05.)** We consider the Poisson problem with Dirichlet boundary conditions,

$$-u'' = f \quad \text{on} \quad \Omega = ]a, b[, \qquad u\,|_{\partial\Omega} = 0.$$

3.2.1. *Discrete weak solution.* We use ansatz functions $\varphi_1, \ldots, \varphi_N \in H_0^1(\Omega)$ and $V_h = \operatorname{span}\{\varphi_1, \ldots, \varphi_N\}$. The discrete weak solution $u_h \in V_h$ satisfies

$$a(u_h, v_h) = \langle f, v_h \rangle \quad \text{for all} \quad v_h \in V_h$$

for the bilinear form $a(u, v) = \int_a^b u'(x)v'(x)dx$. One determines $a_1, \ldots, a_N \in \mathbb{R}$ with

$$u_h = a_1\varphi_1 + \cdots + a_N\varphi_N$$

by solving the linear system $A_h a_h = f_h$ with $(A_h)_{ij} = a(\varphi_i, \varphi_j)$ and $(f_h)_j = \langle f, \varphi_j \rangle$.

3.2.2. *Linear finite elements.* Let $a = x_0 < x_1 < \ldots < x_N < x_{n+1} = b$ and consider the hat functions $\varphi_1, \ldots, \varphi_N$ that are piecewise linear and satisfy

$$\varphi_i(x_j) = \delta_{ij} \quad \text{for all} \quad i, j.$$

3.2.3. *Assembly.* The stiffness matrix can be assembled via

$$A_h = \sum_k P_k A_{I_k} P_k^T, \qquad A_{I_k} = \frac{1}{h_k} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix},$$

where $P_k \in \mathbb{R}^{N \times 2}$ contains a properly placed $2 \times 2$ identity matrix.

3.2.4. *Shape functions.* Using an affine transformation from each local interval $I_k$ to the reference interval $\bar{I} = ]0, 1[$, the hat functions can be written in terms of the two shape functions

$$\phi_1(\xi) = 1 - \xi, \qquad \phi_2(\xi) = \xi.$$

3.2.5. *Recommended reading.* [Is, II.9], [DW, Chapter 4.3]

## 3.3. **More finite elements (18.05.)**

3.3.1. *Conforming meshes.* Conforming triangulations of two-dimensional domains do not have hanging nodes.

3.3.2. *Linear finite elements.* Two-dimensional linear finite elements are constructed from the shape functions

$$\phi_1(\xi,\eta) = 1-\xi-\eta, \qquad \phi_2(\xi,\eta) = \xi, \qquad \phi_3(\xi,\eta) = \eta$$

that are defined on the reference triangle with vertices $(0,0),(1,0),(0,1)$.

3.3.3. *Approximation.* Let $\Omega \subset \mathbb{R}^d$, $d = 1, 2$, be convex and $L^2$-regular, and consider conforming meshes of maximal interval length or triangle diameter $h > 0$. Then, there is a constant $c > 0$ so that the linear finite element solution $u_h$ satisfies

$$a(u-u_h, u-u_h)^{1/2} \le c\,h\|f\|_{L^2}, \qquad \|u-u_h\|_{L^2} \le c\,h^2\|f\|_{L^2}.$$

3.3.4. *Quadratic finite elements.* Quadratic finite elements for $d = 1$ are constructed from the shape functions

$$\phi_1(\xi) = 2(\xi-\tfrac{1}{2})(\xi-1), \qquad \phi_2(\xi) = -4\xi(\xi-1), \qquad \phi_3(\xi) = 2\xi(\xi-\tfrac{1}{2})$$

that are defined on the reference interval $]0,1[$.

3.3.5. *Linear elliptic grid problem.* The finite difference or finite element discretisation of an elliptic PDE typically results in a large, sparse, symmetric positive definite linear system. Sparsity is often met in block-tridiagonal form.

3.3.6. *Recommended reading.* [DW, Chapter 4.3–4.4]

### 3.4. **Sparse linear systems and a two-grid method (20.05.)**

3.4.1. *Linear iterative methods.* A linear iterative method solves $Au = b$ by setting

$$u_{k+1} = u_k + B^{-1}(b - Au_k).$$

Decomposing a symmetric matrix as $A = L + D + L^T$, we consider $B = D$ (Jacobi), $B = \omega D$ with $0 < \omega < 1$ (damped Jacobi), $B = D + L$ (Gauß–Seidel) and $B = (D + L)D^{-1}(D + L^T)$ (symmetric Gauß–Seidel).

3.4.2. *Preconditioned conjugate gradient.* We consider the preconditioned conjugate gradient method with preconditioner $B = D$ (Jacobi), $B = (D + L)D^{-1}(D + L^T)$ (Gauß–Seidel) and $B = \tilde{L}\tilde{L}^T$ (incomplete Cholesky).

3.4.3. *Setup of a two-grid method.* Solve $A_h u = b_h$ as follows:

(1) Few fine grid iterations on $A_h u = b_h$ produce $u_h$.
(2) Compute the correction $E_h$:
    (a) Restrict the residual: $r_H = R(b_h - A_h u_h)$.
    (b) Solve or iterate for $A_H E_H = r_H$ with $A_H = RA_h I$.
    (c) Interpolate the correction: $E_h = IE_H$.
(3) Few fine grid iterations on $A_h u_h = b_h$ started in $u_h + E_h$.

3.4.4. *Recommended reading.* [DW, Chapter 5 & 7]

## 3.5. **Multigrid methods (27.05.)**

3.5.1. *Linear interpolation.* Let $d = 1$. Assuming Dirichlet boundary conditions, linear interpolation from a grid of $N$ interior nodes to a refined grid of $2N + 1$ interior nodes is encoded in a $(2N + 1) \times N$ matrix $I$ containing in each column the vector $\frac{1}{2}(1, 2, 1)^T$ at the appropriate position.

3.5.2. *Full reweighting.* Full reweighting sets for the restriction matrix

$$R = \tfrac{1}{2} I^T,$$

so that the coarse matrix $A_H = RA_hI$ can inherit symmetry from the fine stiffness matrix $A_h$.

3.5.3. *V-cycles and W-cycles.* A multigrid algorithm works one a sequence of (nested) grids in a recursive way. The V-cycle has one coarse solve, the W-cycle several of them, depending on the number of levels.

3.5.4. *Recommended reading.* [DW, Chapter 7]

### 3.6. **Spectral methods (08.06.)**

3.6.1. *Eigenfunction expansions.* If the ansatz functions of a Galerkin method are eigenfunctions of the elliptic operator, then the corresponding linear system is diagonal.

3.6.2. *Fourier–Galerkin method.* A Fourier–Galerkin method constructs

$$u_N \in \left\{ \sum_{k=-N}^{N} \gamma_k e^{ikx}, \gamma_{-k} = \overline{\gamma_k} \right\} = \operatorname{span}_{\mathbb{R}}\{\phi_1, \dots, \phi_{2N+1}\}$$

by solving the elliptic linear system $A_N a_N = f_N$ defined by the square matrix $(A_N)_{kl} = \langle \phi_k, A\phi_l \rangle$ and the right hand side vector $(f_N)_k = \langle \phi_k, f \rangle$.

3.6.3. *Fourier collocation method.* Fourier collocation constructs

$$u_N(x) = \sum_{k=-N/2}^{N/2-1} c_k e^{ikx}, \qquad c_{-k} = \overline{c_k},$$

so that $(Au_N)(x_j) = f(x_j)$ for all $x_j = j \cdot 2\pi/N$, $j = -N/2, \dots, N/2 - 1$.

3.6.4. *Discrete Fourier transform.* The FFT allows to compute the discrete Fourier transform

$$F_N : \mathbb{C}^N \to \mathbb{C}^N, \quad (F_N v)_k = \frac{1}{N} \sum_{j=-N/2}^{N/2-1} e^{-ikj2\pi/N} v_j$$

in $O(N \log N)$ flops.

3.6.5. *Trigonometric interpolation.* The trigonometric interpolant $I_N u$ of a $2\pi$-periodic function $u$ with $\partial^s u \in L^2(]-\pi, \pi[)$, $s \geq 1$, satisfies

$$\|I_N u - u\|_{L^2} \leq C N^{-s} \|\partial^s u\|_{L^2},$$

where the constant $C \geq 0$ only depends on $s$.

3.6.6. *Recommended reading.* [DW, Chapter 4.2], [Is, II.10], [Lu, III.1]

## 4. Stiff ordinary differential equations

### 4.1. **Stability notions (10.06.)**

4.1.1. *Stability function.* The stability function $R : \mathbb{C} \to \mathbb{C}$ of a method with constant step size $h > 0$ satisfies

$$y_1 = R(h\lambda)y_0,$$

when applied to $y'(t) = \lambda y(t)$ for $\lambda \in \mathbb{C}$.

4.1.2. *A-stability.* A method is called A-stable if the stability domain

$$S = \{z \in \mathbb{C}; |R(z)| \leq 1\}$$

satisfies $\{z \in \mathbb{C};\, \Re(z) \leq 0\} \subset S$. Explicit Runge–Kutta methods have a polynomial stability function and are thus not A-stable.

4.1.3. *L-stability.* A method is called L-stable if it is

$$\text{A-stable and } \lim_{z\to\infty} R(z) = 0.$$

The trapezoidal rule has $R(z) = (1 + \frac{z}{2})/(1 - \frac{z}{2})$ and is thus A-stable but not L-stable.

4.1.4. *A($\alpha$)-stability.* A method is called A($\alpha$)-stable, $0 \leq \alpha \leq \pi/2$, if

$$\{z \in \mathbb{C};\ |\arg(-z)| \leq \alpha\} \subset S.$$

4.1.5. *Recommended reading.* [HW, IV.2 & IV.3]

## 4.2. **Implicit methods (15.06.)**

4.2.1. *Linear multistep methods.* A linear $s$-step method for the ordinary differential equation $y' = f(t, y)$ is defined as

$$\sum_{m=0}^{s} a_m y_{n+m} = h \sum_{m=0}^{s} b_m f(t_{n+m}, y_{n+m})$$

for given $a, b \in \mathbb{R}^{s+1}$ with $a_s = 1$. The stability domain is defined as

$$\{z \in \mathbb{C}; |w_1(z)|, \dots, |w_s(z)| \leq 1\},$$

where $w_1(z), \dots, w_s(z)$ are the roots of the stability polynomial

$$w \mapsto \sum_{m=0}^{s} (a_m - z b_m) w^m.$$

4.2.2. *Backward differentiation formulas.* A BDF method

$$\sum_{m=0}^{s} a_m y_{n+m} = h b_s f(t_{n+s}, y_{n+s})$$

achieves order $s$ by an optimal choice of $a_0, \dots, a_{s-1}, b_s$. For $s = 1, 2$ BDF is A-stable. For $s = 3, 4, 5$ BDF is A($\alpha$)-stable with $\alpha$ ranging from $83^o$ to $17^o$.

4.2.3. *Newton iteration for implicit RK-methods.* A Newton iteration for solving the stage system of an implicit $s$-stage Runge–Kutta method requires to evaluate the Jacobian $f_y(\cdot, \cdot) \in \mathbb{R}^{d \times d}$ for $s$ different arguments per iteration step and is thus very costly.

4.2.4. *Simplified Newton iteration.* A simplified Newton iteration replaces the Jacobian of the stage function $F : \mathbb{R}^{ds} \to \mathbb{R}^{ds}$ by

$$M := \mathrm{Id}_{ds \times ds} - hA \otimes f_y(t_0, y_0),$$

thus achieving linear convergence. The LU decomposition of $M$ can be precomputed beforehand.

4.2.5. *Recommended reading.* [DB, 6.2, 7.3–7.4]

## 4.3. **Adapative BDF methods (17.06.)**

4.3.1. *Basic adaptive algorithm.* We are at time $t_j$ with a prediceted step size $h_j$ and order $s$.

(1) Compute predictor polynomials by interpolating in $\nu+1$ points for $\nu = s-1, s, s+1$ and extrapolate the corresponding predictor value $y^0_{\nu,j+1}$.
(2) Perform simplified Newton iterations started in the predictor value to obtain $y_{\nu,j+1}$.
(3) Compute the local error estimators $\varepsilon_{\nu,j+1}$.
(4) Based on the local error tolerance TOL and a safety factor $\rho < 1$ evaluate the predicted step size

$$h_{\nu,j+1} = \sqrt[\nu+1]{\frac{\rho \cdot \mathrm{TOL}}{\|\varepsilon_{\nu,j+1}\|}}\, h_j.$$

(5) If at least one estimator is below TOL, then choose the corresponding $\nu$ and set $y_{j+1} = y_{\nu,j+1}$. Choose the predicted step size $h_{j+1}$ among the $h_{\nu,j+1}$.
(6) If all estimators are above TOL, then repeat the step with one of the step sizes $h_{\nu,j+1}$.

4.3.2. *The predictor polynomial.* For BDF(1), that is, the implicit Euler method, the predictor value $y^0_{j+1}$ that results from extrapolating the interpolating polynomial to time $t_{j+1}$, satisfies the implicit midpoint rule

$$y^0_{j+1} = y_{j-1} + 2hf(t_{j+1}, \tfrac{1}{2}(y^0_{j+1} + y_{j-1})).$$

4.3.3. *Recommended reading.* [DB, 7.4]

## 5. Parabolic equations

### 5.1. **Order reduction & method of lines (22.06.)**

5.1.1. *Order reduction.* Depending on the sine coefficients $(a_k)_{k\geq 1}$ of the initial condition $u_0$ for the one-dimensional diffusion problem

$$u_t = u_{xx} \quad \text{on } ]0,\pi[$$

with homogeneous Dirichlet boundary condition, one can chose a worst case step size $\tau_k > 0$ so that the the local error of a L-stable time discretization satisfies

$$\|\varepsilon(\tau_k)\|_{L^2} = O(a_k)$$

regardless of the order of the method. In particular, the initial condition $u_0(x) = x(\pi - x)$ has $a_k = O(k^{-3})$ and

$$\|\varepsilon(\tau_k)\|_{L^2} = O(\tau_k^{3/2}),$$

that is an effective order $p^* = \frac{1}{2}$.

5.1.2. *Space-time discretization of parabolic PDEs.* We distinguish the method of lines, that discretizes first in space and then in time, from the method of time layers (or Rothe method), that discretizes first in time and then in space.

5.1.3. *Dynamic regridding.* Dynamic regridding works with a moving frame solution $v(t) = u(x(t), t)$ that is determined from the discretization of the coupled PDE system

$$M(v' - u_x x') = F(v), \qquad u_x^T v' = 0.$$

The crossing of nodes might result in step size restrictions.

5.1.4. *Recommended reading.* [DW, 9.1.3 & 9.2.1]

## 5.2. **Methods of lines and Rothe method (24.06.)**

5.2.1. *Centered finite differences for the Laplacian.* For a uniform mesh on the unit square $]0,1[^2$ of mesh size $h > 0$ the centered finite difference Laplacian $-\Delta_h$ has its eigenvalues in the interval $]0, 8/h^2[$.

5.2.2. *Method of lines for the heat equation with centered finite difference.* The eigenvalues of the explicit Euler evolution map $\mathrm{Id} + \tau\Delta_h$ are in modulus less than one, if and only if $\tau < h^2/4$. The implicit Euler method and the trapezoidal rule, however, are unconditionally stable.

5.2.3. *Crank–Nicolson scheme.* The combination of centered finite differences in space and the trapezoidal rule in time is the Crank–Nicolson method for parabolic problems.

5.2.4. *Method of lines for the heat equation with a Galerkin method.* A Galerkin method approximates the solution by $u_h(x,t) = \sum_{j=1}^{N} \alpha_j(t)\phi_j(x)$ resulting in the ordinary differential equation

$$M\alpha'(t) + A\alpha(t) = F(t),$$

where $M_{ij} = \langle \phi_i, \phi_j \rangle$ and $A_{ij} = b(\phi_i, \phi_j)$ define the mass and stiffness matrix, respectively, and $F_i = \langle f(t), \phi_i \rangle$ the load vector.

5.2.5. *Method of time layers.* The method of time layers discretizes first in time. For time adapativity one constructs time-discretizations of order $p$ and $p+1$,

$$u_\tau^{p+1} = u_\tau^p + \Delta u_\tau^p,$$

where the increments are multiplicatively determined by $\Delta u_\tau^0, \Delta u_\tau^1, \ldots$

5.2.6. *Recommended reading.* [DW, 9.2.]

## 6. Schrödinger equation

### 6.1. **Spectral methods (01.07.)**

6.1.1. *The setup.* We consider the Hilbert space $L^2(\mathbb{R}^d, \mathbb{C})$ and a linear, self-adjoint operator $H$ as for example the Schrödinger operator $H = -\Delta + V$. The corresponding time-dependent Schrödinger equation reads

$$i\partial_t \psi = H\psi, \qquad \psi(0) = \psi_0.$$

6.1.2. *Galerkin approximation.* We consider $\operatorname{span}\{\phi_0, \dots, \phi_K\} \subset L^2(\mathbb{R}^d, \mathbb{C})$ and construct the Galerkin approximation

$$\psi_K(t) = \sum_{j=0}^{K} c_k(t)\phi_j$$

by solving $iM_k\dot{c} = H_k c$.

6.1.3. *Hermite functions.* Let $A = \frac{1}{\sqrt{2}}(x + \nabla)$ and $A^\dagger = \frac{1}{\sqrt{2}}(x - \nabla)$. Then,

$$\phi_0(x) = \pi^{-d/4} \exp(-\tfrac{1}{2}|x|^2), \qquad x \in \mathbb{R}^d,$$

satisfies $A\phi_0 = 0$, and $(\phi_k)_{k\in\mathbb{N}^d}$ with

$$\phi_{k+e_j} = \frac{1}{\sqrt{k_j + 1}} A_j^\dagger \, \phi_k, \qquad k \in \mathbb{N}^d,$$

defines an orthonormal basis of $L^2(\mathbb{R}^d)$.

6.1.4. *Approximation theory.* The orthogonal projection $P_K f = \sum_{|k|<K} \langle \phi_k, f\rangle \phi_k$ satisfies for all $s \le K$

$$\|f - P_K f\|_{L^2} \le C(s) K^{-s/2} \|A^s f\|_{L^2}.$$

6.1.5. *Hyperbolic cross approximation.* The orthogonal projection with respect to the hyperbolic cross

$$\mathcal{K} = \{k \in \mathbb{N}^d \mid (1 + k_1) \cdots (1 + k_d) < K\}$$

satisfies

$$\|f - P_{\mathcal{K}} f\|_{L^2} \le C(s, d) K^{-s/2} \max_{\|\sigma\|_\infty \le s} \|A^s f\|_{L^2}.$$

The hyperbolic cross has at most $K(\log K)^{d-1}$ elements, which has to be compared with the $K^d$ elements of corresponding full index grid.

6.1.6. *Recommended reading.* [Lu, III.1]

## 6.2. **Chebyshev & Lanczos method (06.07.)**

6.2.1. *Method of lines for the Hermite–Galerkin approach.* Discretizing

$$i\partial_t \psi = H\psi, \qquad \psi(0) = \psi_0,$$

in space by a Hermite–Galerkin ansatz in $L^2(\mathbb{R}^d, \mathbb{C})$, we have to approximate $\exp(-itA)$, $t \in \mathbb{R}$, for a Hermitian matrix $A \in \mathbb{C}^{N\times N}$.

6.2.2. *Chebyshev polynomials.* The Chebyshev polynomials

$$T_k(x) = \cos(k\theta), \qquad \theta = \arccos x, \qquad x \in [-1, 1]$$

form a sequence of orthogonal polynomials with the respect to the measure $1/\sqrt{1-x^2}$. For all $\omega \in \mathbb{R}$ and $m \geq |\omega|$ the truncated Chebyshev series for $f(x) = e^{i\omega x}$ satisfies

$$\max_{x\in[-1,1]} |p_{m-1}(x) - e^{i\omega x}| \leq 4 \left(\exp(1 - (\tfrac{\omega}{2m})^2)\tfrac{|\omega|}{2m}\right)^m.$$

6.2.3. *Chebyshev method.* The Chebyshev method approximates

$$\exp(-i\Delta t A) \approx P_{m-1}(\Delta t A)$$

for $A = A^* \in \mathbb{C}^{N\times N}$ with $\sigma(A) \subset [a, b]$, where $P_{m-1}$ is the truncated Chebyshev series for $f(x) = e^{i\omega x}$ with $\omega = \Delta t(b-a)/2$.

6.2.4. *Krylov methods.* We approximate $\exp(-i\Delta t A)v$ by a Galerkin approximation on the Krylov space

$$\mathcal{K}_m(A, v) = \operatorname{span}\{v, Av, \ldots, A^{m-1}v\}, \qquad m \ll N,$$

and solve $i\dot{c}(t) = T_m c(t)$ with stiffness matrix $T_m \in \mathbb{C}^{m\times m}$, $(T_m)_{jk} = v_j^* A v_k$ for an orthonormal basis $v_1, \ldots, v_m$ of $\mathcal{K}_m(A, v)$.

6.2.5. *Hermitian Lanczos method.* One constructs the orthormal basis $v_1, \ldots, v_m$ by a Gram–Schmidt procedure such that the stiffness matrix $T_m$ is tridiagonal.

6.2.6. *Recommended reading.* [Lu, III.2]

## 6.3. **Splitting methods (08.07.)**

6.3.1. *Accuracy of the Lanczos method.* Let $A = A^* \in \mathbb{C}^{N\times N}$ with $\sigma(A) \subset [a,b]$ and $v \in \mathbb{C}^N$ with $\|v\|_2 = 1$. The Hermitian Lanczs approximation satisfies

$$\left\|V_m e^{-i\Delta t T_m} e_1 - e^{-i\Delta t A} v\right\|_2 \leq 8\left(\exp(1-(\tfrac{\omega}{2m})^2)\tfrac{|\omega|}{2m}\right)^m$$

for $m \geq \omega = \Delta t(b-2)/2$.

6.3.2. *Splitting in kinetic and potential energy.* We solve the Schrödinger equation $i\partial_t\psi = (T+V)\psi$, $\psi(0) = \psi_0$ by the Strang splitting scheme

$$\psi_{n+1} = S(\Delta t)\psi_n, \qquad S(\Delta t) = e^{-i\Delta t/2V} e^{-i\Delta t T} e^{-i\Delta t/2V},$$

that produces approximations $\psi_n \approx \psi(t_n,\cdot)$ on the uniform time grid $t_{n+1} = t_n + \Delta t$.

6.3.3. *Fourier collocation.* Assuming that $\psi(t,\cdot)$ is negligible outside an interval $[a,b]$, one performs a spatial change of coordinates $x \mapsto \alpha x + \beta$ from $[a,b]$ to $[-\pi,\pi]$ and approximates the solution $\psi(t,\cdot)$ as if it were periodic, $\psi(t,-\pi) = \psi(t,\pi)$ for all $t$. Let $x_j$, $j = -K/2,\dots,K/2-1$ be a uniform grid of $[-\pi,\pi]$ and $\psi_K(t,\cdot)$ the corresponding trigonometric interpolant. Then $u_j(t) = \psi_K(x_j,t)$ satisfies

$$i\dot{u} = \mathcal{F}_K^{-1} D_K \mathcal{F}_K + V_k u,$$

where $D_K = \mathrm{diag}(k^2)$ and $V_K = \mathrm{diag}(V(x_j))$ are diagonal $K\times K$ matrices.

6.3.4. *Fourier split-step algorithm.* Strang splitting with Fourier collocation is realized by the following one-step propagation scheme:

(1) Potential energy: $u_j = e^{-i\Delta t/2V(x_j)} u_j$ for all $j$.
(2) Kinetic energy: $u = \mathcal{F}_K u$, $u_k = e^{-i\Delta t k^2} u_k$ for all $k$, $u = \mathcal{F}_K^{-1} u$.
(3) Potential energy: $u_j = e^{-i\Delta t/2V(x_j)} u_j$ for all $j$.

6.3.5. *Recommended reading.* [Lu, III.3]

## 7. Hyperbolic equations

### 7.1. **Linear advection (13.07.)**

7.1.1. *Structural properties (Schrödinger equation, time discretization).* Strang splitting and the Lanczos propagator are unitary approximations to $e^{-i\Delta t H}$, while the Chebyshev propagator is not. Since

$$S(\Delta t)^{-1} = S(-\Delta t),$$

Strang splitting is time-reversible, while neither the Chebyshev nor the Lanczos method are time-reversible.

7.1.2. *Error bounds for splitting methods.* One can prove for the Strang splitting solution $\psi_n$ at time $t = t_n$

$$\|\psi_n - \psi(t)\|_{L^2} \leq C(\Delta t)^2 t,$$

where the constant $C > 0$ only depends on regularity properties of $T$ and $V$.

7.1.3. *The linear advection equation.* The solution of the linear advection equation on the real line

$$\partial_t u + c\,\partial_x u = 0 \quad \text{on} \quad \mathbb{R} \times [0, \infty), \qquad u(\cdot, 0) = u_0$$

is constant along the characteristic lines, that is,

$$u(x, t) = u_0(x - ct), \qquad (x, t) \in \mathbb{R} \times [0, \infty[.$$

7.1.4. *Conservation laws.* Linear advection is a conservation law

$$\partial_t u + \partial_x f = 0$$

with flux function $f = cu$.

7.1.5. *Explicit upwind method.* Let $c > 0$. The explicit upwind discretization with time step $\Delta t^{n+1/2}$, $n \geq 0$, and cell size $\Delta x_i$, $i = 0, \ldots, I$ approximates the cell average

$$u_i^n \approx \frac{1}{\Delta x_i} \int_{x_{i-1/2}}^{x_i+1/2} u(x, t^n) dx$$

by setting

$$u_i^{n+1} = u_i^n - c(u_i^n - u_{i-1}^n) \frac{\Delta t^{n+1/2}}{\Delta x_i}, \qquad 0 < i < I,$$

and uses the boundary condition at $a = x_{-1/2}$ for defining $u_0^n$.

7.1.6. *Recommended reading.* [Tr, Chapter 2]

## 7.2. **Nonlinear conservation laws (15.07.)**

7.2.1. *Recommended reading.* [Tr, Chapter 3]