Computational physics

Heiko Rieger and Adam Wysocki

Theoretical Physics
Saarland University

SS2017

Contents

Examples of ordinary differential equations (ODEs)

- Newton's law:

$$m\frac{d^2\mathbf{r}}{dt^2} = \mathbf{f}(\mathbf{r}(t), t) \tag{1}$$

  with mass m, position $\mathbf{r}$, time t and force $\mathbf{f}$.

- Predator-prey (or Lotka-Volterra) equations:

$$\frac{dx}{dt} = \alpha x - \beta xy \tag{2}$$

$$\frac{dy}{dt} = \delta xy - \gamma y \tag{3}$$

  with number of prey x, number of predator y and interaction parameters of the two species α, β, δ, γ.

- Static beam (or Euler-Bernoulli) equation:

$$\frac{d^2}{dx^2}\left(E(x)I(x)\frac{d^2w}{dx^2}\right) = q(x) \tag{4}$$

  with deflection of the beam $w(x)$, load $q(x)$, elastic modulus E and area moment of inertia I.

Reduction of *n*th-order ODE to 1st-order ODEs

Explicit ODE of order n

$$f\left(t, y, y', y'',\ \cdots,\ y^{(n-1)}\right) = y^{(n)} \tag{5}$$

can be reduced to a system of n first-order ODEs

$$\begin{aligned} \mathbf{y}' &= \mathbf{f}(t, \mathbf{y}) = \mathbf{f}(t, y_1, \cdots, y_n) && (6) \\ (y_1', \cdots, y_n') &= (y_2, \cdots, y_n, f(t, y_1, \cdots, y_n)) && (7) \end{aligned}$$

defining new functions $y_i = y^{(i-1)}$.

Initial value problem: $\mathbf{y}(t = 0) = \mathbf{y}_0$

Notation: $y' = \frac{dy}{dt}$, $y'' = \frac{d^2y}{dt^2}$, ... or $\dot{y} = \frac{dy}{dt}$, $\ddot{y} = \frac{d^2y}{dt^2}$, ...

Example

Newton's law:

$$m\frac{d^2x(t)}{dt^2} = f(x(t), t) \tag{8}$$

Harmonic oscillator:

$$mx'' = -kx \Rightarrow x'' = -\frac{k}{m}x \tag{9}$$

Define new functions: $y_1 := x$ and $y_2 := x'$

$$y_1' = x' = y_2 \tag{10}$$

$$y_2' = x'' = -\frac{k}{m}y_1 \tag{11}$$

Euler method

Grid points t_i and $y_i = y\left(t_i\right)$.

$$\frac{dy}{dt}(t_i) \approx \frac{y_{i+1} - y_i}{t_{i+1} - t_i} \approx f\left(y_i, t_i\right) \tag{12}$$

$$\Rightarrow y_{i+1} \approx y_i + \left(t_{i+1} - t_i\right) \cdot f_i \tag{13}$$

For equidistant spacing $t_{i+1} - t_i = \tau \ll 1$:

$$y_{i+1} = y_i + \tau f_i + \mathcal{O}\left(\tau^2\right) \tag{14}$$

Integrate from t_0 till $t_N = t_0 + N\tau = t_0 + T$.
Global truncation error is $N \cdot \mathcal{O}\left(\tau^2\right) \approx \frac{T}{\tau}\tau^2 = T\tau$
$\Rightarrow$ Euler method is very inaccurate.

Remark: Euler method is an explicit method: $y_{i+1} = F(y_i)$.

Picard method

Formal solution of $y' = f(y,t)$ is $y_{i+j} = y_i + \int_{t_i}^{t_{i+j}} dt\, f(y,t)$.

Using trapezoid rule we obtain:

$$y_{i+1} = y_i + \frac{\tau}{2}\left(f_i + f_{i+1}\right) + \mathcal{O}\left(\tau^3\right) \tag{15}$$

with $f_{i+1} = f\left(y_{i+1}, t_{i+1}\right)$.
Can be solved via fixed point iteration

$$y_{i+1}^{(k+1)} = y_i + \frac{\tau}{2}\left(f_i + f_{i+1}^{(k)}\right) \tag{16}$$

until $y_{i+1}^{(k+1)} = y_{i+1}^{(k)}$ using an initial guess $y_{i+1}^{(0)} = y_1$ obtained with Euler method. Convergence can be slow.

Remark: The Picard method is an implicit method: $G(y_i, y_{i+1}) = 0$. Implicit methods are usually more stable for solving stiff ODEs, e.g.: $y' = \lambda y$ with large $|Re(\lambda)|$.

Predictor corrector methods: Part 1

Use Euler method to predict y_{i+1}:

$$y_{i+1}^{(predict)} = y_i + \tau f_i\left(y_i, t_i\right) \tag{17}$$

and next Picard method to improve y_{i+1}:

$$y_{i+1}^{(correct)} = y_i + \frac{\tau}{2}\left\{f\left(y_i, t_i\right) + f\left(y_{i+1}^{(predict)}, t_{i+1}\right)\right\} \tag{18}$$

$$y_{i+1} = y_{i+1}^{(correct)} \tag{19}$$

Predictor corrector methods: Part 2

Increase the number of grid points: $y_{i+2} = y_i + \int_{t_i}^{t_{i+2}} dt\, f\left(y, t\right)$

Use linear extrapolation: $f_{i+2} = f_i + \frac{t_{i+2}-t_i}{t_{i+1}-t_i}(f_{i+1} - f_i) = 2f_{i+1} - f_i$ and trapezoid rule:

$$\Rightarrow \quad \int_{t_i}^{t_{i+2}} dt\, f\left(y, t\right) \approx \frac{2\tau}{2}\left(2f_{i+1} - f_i + f_i\right) = 2\tau f_{i+1} \qquad (20)$$

$$\Rightarrow \quad y_{i+2} = y_i + 2\tau f_{i+1} + \mathcal{O}\left(\tau^3\right) \qquad (21)$$

Initial values are y_0 and y_1 for $f_1 = f(y_1, t_1)$, use Taylor expansion $y_1 = y_0 + \tau f_0 + \frac{\tau^2}{2}\left(\frac{\partial f_0}{\partial t} + f_0 \frac{\partial f_0}{\partial y}\right) + \mathcal{O}\left(\tau^3\right)$.

Remark: Is an example for a multistep method, i.e., it uses the information from the past. Higher order methods can be obtained with a better quadrature, for example, Simpson's rule.

Derivation of Runge Kutta methods

Taylor series of $y(t+\tau)$ around $y(t)$:

$$y(t+\tau) = y(t) + \tau y'(t) + \frac{\tau^2}{2} y''(t) + \frac{\tau^3}{3!} y^{(3)}(t) + \ldots \tag{22}$$

$$\begin{aligned}
y' &= f(y,t) && (23)\\
y'' &= y' f_y + f_t && (24)\\
&= f f_y + f_t && (25)\\
y^{(3)} &= \left(y' f_y + f_t\right) f_y + f\left(y' f_{yy} + f_{yt}\right) + y' f_{ty} + f_{tt} && (26)\\
&= f f_y^2 + f_t f_y + f^2 f_{yy} + 2 f f_{yt} + f_{tt} && (27)
\end{aligned}$$

$$\begin{aligned}
\Rightarrow y(t+\tau) &= y + \tau f + \frac{\tau^2}{2}\left(f_t + f f_y\right) && (28)\\
&+ \frac{\tau^3}{6}\left(f_{tt} + 2 f f_{ty} + f^2 f_{yy} + f f_y^2 + f_t f_y\right) && (29)\\
&+ \mathcal{O}\left(\tau^4\right) && (30)
\end{aligned}$$

Derivation of Runge Kutta methods

In general a Runge Kutta method of order s can be written as:

$$y\left(t+\tau\right)=y\left(t\right)+\tau\sum_{i=1}^{s}b_i k_i+\mathcal{O}(\tau^{s+1}) \tag{31}$$

with

$$k_i=f\left(y+\tau\sum_{j=1}^{i-1}a_{ij}k_j, t+\tau\sum_{j=1}^{i-1}a_{ij}\right) \tag{32}$$

Comparison with the Taylor series yields an under-determined system of constraints on b_i and a_{ij}.

Classical Runge Kutta method (RK4)

$$y(t+\tau) = y(t) + \frac{\tau}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) + \mathcal{O}(\tau^5) \tag{33}$$

with

$$\begin{aligned}
k_1 &= f(t, y), && (34)\\
k_2 &= f(t + \frac{\tau}{2}, y + \frac{\tau}{2}k_1), && (35)\\
k_3 &= f(t + \frac{\tau}{2}, y + \frac{\tau}{2}k_2), && (36)\\
k_4 &= f(t + \tau, y + \tau k_3). && (37)
\end{aligned}$$

Boundary value problem

Consider, for example, a second-order ODE:

$$y'' = f(t, y, y'), \quad y(t_0) = y_0, \quad y(t_1) = y_1 \tag{38}$$

Solution via shooting method:

- Reformulate boundary to initial value problem:

  $$y'' = f(t, y, y'), \quad y(t_0) = y_0, \quad y'(t_0) = a \tag{39}$$

  with a unknown variable a.
- Denote the solution of Eq.(39) for fixed a as $y(t; a)$. If $F(a) = y(t_1; a) - y_1 = 0 \Rightarrow y(t; a)$ is a solution of the boundary value problem Eq.(38).
- In practice, integrate Eq.(39) with RK4 for different a and use Newton's method to find $F(a) = 0$.
- Newton's method: $a_{n+1} = a_n - \frac{F(a_n)}{F'(a_n)}$ with forward difference formula $F'(a_n) = \frac{F(a_n+\delta a) - F(a_n)}{\delta a}$ where δa is small.
  Remark: Method fails near a local extremum $F'(a) \approx 0$.

Literature

- Josef Stoer and Roland Bulirsch: *Introduction to Numerical Analysis*
- William H. Press, Saul Teukolsky, William T. Vetterling und Brian P. Flannery: *Numerical Recipes in C. The Art of Scientific Computing*

Partial differential equations (PDEs)

Linear PDEs:

- Poisson's equation: $\Delta\phi(\mathbf{r}) = -\frac{\rho(\mathbf{r})}{\varepsilon_0}$
- Diffusion equation: $\frac{\partial n(\mathbf{r},t)}{\partial t} - \nabla\cdot\left(D(\mathbf{r})\nabla n(\mathbf{r},t)\right) = S(\mathbf{r},t)$
- Wave equation: $\frac{1}{c^2}\frac{\partial^2 u(\mathbf{r},t)}{\partial t^2} - \Delta u(\mathbf{r},t) = R(\mathbf{r},t)$
- Schrödinger equation: $-\frac{\hbar}{\imath}\frac{\partial\psi(\mathbf{r},t)}{\partial t} = \hat{H}\psi(\mathbf{r},t)$

Nonlinear PDEs:

- Continuity equation: $\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{v}) = 0$
- Navier Stokes equations: $\frac{\partial\mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v} + \frac{1}{\rho}\nabla p - \eta\Delta\mathbf{v} = 0$

Again linear PDEs:

- Navier Stokes equations becomes linear for negligible inertia and large viscosity (swimming of microorganisms).
  Stokes equations: $\frac{1}{\rho}\nabla p - \eta\Delta\mathbf{v} = 0$

Finite differences

Using Taylor series expansion one obtains 1st order forward time difference:

$$\frac{\partial A(\mathbf{r}, t_k)}{\partial t} \approx \frac{A(\mathbf{r}, t_{k+1}) - A(\mathbf{r}, t_k)}{\tau}, \tag{40}$$

with $\tau = t_{k+1} - t_k$, or 1th order central difference:

$$\frac{\partial A(\mathbf{r}, t_k)}{\partial t} \approx \frac{A(\mathbf{r}, t_{k+1}) - A(\mathbf{r}, t_{k-1})}{2\tau} \tag{41}$$

Applying central difference to $\frac{\partial A(\mathbf{r}, t+\tau/2)}{\partial t}$ and $\frac{\partial A(\mathbf{r}, t-\tau/2)}{\partial t}$ results in 2nd order central difference:

$$\frac{\partial^2 A(\mathbf{r}, t_k)}{\partial t^2} \approx \frac{A(\mathbf{r}, t_{k+1}) - 2A(\mathbf{r}, t_k) + A(\mathbf{r}, t_{k-1})}{\tau^2}. \tag{42}$$

Similar expessions can be obtained for space derivatives: $\frac{\partial A(x, t_k)}{\partial x}$, $\frac{\partial^2 A(x, t_k)}{\partial x^2}$, $\frac{\partial^2 A(x, t_k)}{\partial x \partial y}$, etc.

Boundary conditions

PDE plus a boundary condition defines the physical problem.

Ω indicates the solution domain and $\partial\Omega$ the boundary of Ω.

- Dirichlet boundary condition:

$$A\left(\mathbf{r}\right)|_{r\in\partial\Omega} = g\left(\mathbf{r}\right) \tag{43}$$

  Examples: fluid velocity vanishes at the boundary, surface is held at a fixed temperature, etc.

- Neumann boundary condition:

$$\frac{\partial}{\partial \mathbf{n}} A\left(\mathbf{r}\right)|_{r\in\partial\Omega} = h\left(\mathbf{r}\right), \tag{44}$$

  with normal derivative $\frac{\partial}{\partial \mathbf{n}} A(\mathbf{r}) = \nabla A(\mathbf{r}) \cdot \mathbf{n}(\mathbf{r})$.
  Examples: prescribed heat flux from a surface, etc.

- Combinations of Dirichlet and Neumann boundary conditions.
- Periodic boundary conditions.

Nonstandard discretization

Different discretization for inhomogeneous grids or non-constant variables, e.g., $\nabla \cdot (D(\mathbf{r}) \nabla n(\mathbf{r}, t))$

- Consider 1D Poisson's equation with permittivity $\varepsilon(x)$:

$$\frac{d}{dx}\left(\varepsilon(x)\frac{d\phi}{dx}\right) = -\rho \text{ for } x \in [0, L] \tag{45}$$

- Approximate $\frac{d\phi}{dx}$ with a central difference at $x_{k+1/2}$ and $x_{k-1/2}$, followed by a central difference between this points:

$$\frac{d}{dx}\left(\varepsilon\frac{d\phi}{dx}\right) \approx \frac{\varepsilon_{k+1/2}(\phi_{k+1} - \phi_k) - \varepsilon_{k-1/2}(\phi_k - \phi_{k-1})}{h^2} \tag{46}$$

- Using interpolation to estimate $\varepsilon_{k+1/2}$ and $\varepsilon_{k-1/2}$ one obtains a discretized form of Eq.(45):

$$(\varepsilon_{k+1} + \varepsilon_k)\phi_{k+1} - 4\varepsilon_k\phi_k + (\varepsilon_{k-1} + \varepsilon_k)\phi_{k-1} = -2h^2\rho_k \tag{47}$$

Finite difference method

Discretized linear PDE, $Lu(\mathbf{r}, t) = f(\mathbf{r}, t)$, where L is a differential operator, transforms into a system of linear equations $\mathbf{Au} = \mathbf{b}$, which can be solved with methods of numerical linear algebra.

Consider, for example, Eq.(45) with Dirichlet boundary condition $\phi(0) = 0$ and $\phi(L) = 0$. The discretized version is

$$(\varepsilon_{k-1} + \varepsilon_k)\phi_{k-1} - 4\varepsilon_k\phi_k + (\varepsilon_{k+1} + \varepsilon_k)\phi_{k+1} = -2h^2\rho_k, \quad (48)$$

for $k = 1, \ldots, n$ with boundary conditions $\phi_0 = \phi_{n+1} = 0$.
Eq.(48) can be written in a matrix form $\mathbf{Au} = \mathbf{b}$ with, see next slide

Finite difference method

$$\mathbf{A} = -\begin{pmatrix} -4\varepsilon_1 & \varepsilon_2+\varepsilon_1 & & & \\ \varepsilon_1+\varepsilon_2 & -4\varepsilon_2 & \varepsilon_3+\varepsilon_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \varepsilon_{n-2}+\varepsilon_{n-1} & -4\varepsilon_{n-1} & \varepsilon_n+\varepsilon_{n-1} \\ & & & \varepsilon_{n-1}+\varepsilon_n & -4\varepsilon_n \end{pmatrix} \tag{49}$$

and

$$\mathbf{u} = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{n-1} \\ \phi_n \end{pmatrix} \quad \text{and} \quad \mathbf{b} = 2h^2 \begin{pmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_{n-1} \\ \rho_n \end{pmatrix} \tag{50}$$

The matrix $\mathbf{A}$ is tridiagonal and symmetric positive definite $\Rightarrow$ invertible

Relaxation methods

- Search for a stationary solution ($\frac{\partial n(\mathbf{r})}{\partial t} = 0$) of the diffusion equation: $-\frac{d}{dx}\left(D\left(x\right)\frac{dn(x)}{dx}\right) = S\left(x\right)$.
- Discretized form

$$n_i = \frac{D_{i+1/2}n_{i+1} + D_{i-1/2}n_{i-1} + h^2 S_i}{D_{i+1/2} + D_{i-1/2}} \quad (51)$$

- Make an initial guess $n_i^{(0)}$ ($i = 0, \ldots, n+1$) which fulfills the boundary conditions.
- Calculate $n_i = \frac{D_{i+1/2}n_{i+1}^{(0)} + D_{i-1/2}n_{i-1}^{(0)} + h^2 S_1}{D_{i+1/2} + D_{i-1/2}}$.
- Mix the solution n_i with the guess $n_i^{(k)}$ according to $n_i^{(k+1)} = (1-p)\, n_i^{(k)} + pn_i$ and iterate till convergence. Mixing parameter $p \in [0, 2]$ controls convergence.
- Check if boundary conditions are fulfilled.

Initial value problems

Higer order PDEs can be reduced to a system of first-order PDEs.

Example: Wave equation $\frac{1}{c^2}\frac{\partial^2 u(\mathbf{r},t)}{\partial t^2} - \Delta u(\mathbf{r},t) = R(\mathbf{r},t)$:

$$\frac{\partial u(\mathbf{r},t)}{\partial t} = v(\mathbf{r},t) \tag{52}$$

$$\frac{1}{c^2}\frac{\partial v(\mathbf{r},t)}{\partial t} = \Delta u(\mathbf{r},t) + R(\mathbf{r},t) \tag{53}$$

Has now a structure of a diffusion equation:

$$\frac{\partial n(\mathbf{r},t)}{\partial t} = D\Delta n(\mathbf{r},t) + S(\mathbf{r},t) \tag{54}$$

Explicit method: Forward-Time Central-Space

Discretized form of the diffusion equation in 1D using a forward difference at time t_k and a second-order central difference at position x_i is

$$\frac{n_i^{k+1} - n_i^k}{\tau} = D \frac{n_{i+1}^k - 2n_i^k + n_{i-1}^k}{h^2} + S_i^k \tag{55}$$

with $n(x_i,\, t_k) = n_i^k$.

Truncation error is $\mathcal{O}\left(h^2\right) + \mathcal{O}\left(\tau\right)$.
Caution: Method is numerically stable and convergent whenever $\frac{D\tau}{h^2} < \frac{1}{2}$ or $\tau < \frac{h^2}{2D}$ (Courant–Friedrichs–Lewy condition), i.e., the maximum allowed τ is the diffusion time across a cell of width h.

Implicit method: Crank–Nicolson method

Combine the forward time difference at position x_i and the average of the central space difference at time t_k and t_{k+1}.

$$\frac{n_i^{k+1} - n_i^k}{\tau} = \frac{D}{2h^2}\left[\left(n_{i+1}^{k+1} - 2n_i^{k+1} + n_{i-1}^{k+1}\right) + \left(n_{i+1}^{k} - 2n_i^{k} + n_{i-1}^{k}\right)\right] + \frac{1}{2}\left(S_i^{k+1} + S_i^k\right) \quad (56)$$

or

$$-rn_{i-1}^{k+1} + (1+2r)n_i^{k+1} - rn_{i+1}^{k+1} = rn_{i-1}^{k} + (1-2r)n_i^{k} + rn_{i+1}^{k} + \frac{\tau}{2}\left(S_i^{k+1} + S_i^k\right) = d_i^k \quad (57)$$

with $r = \frac{D\tau}{2h^2}$

Crank–Nicolson method: matrix form

For example with Dirichlet boundary condition $n_0^k = n_{n+1}^k = 0$

$$\begin{pmatrix} 1+2r & -r & & & \\ -r & 1+2r & -r & & \\ & \ddots & \ddots & \ddots & \\ & & -r & 1+2r & -r \\ & & & -r & 1+2r \end{pmatrix} \begin{pmatrix} n_1^{k+1} \\ n_2^{k+1} \\ \vdots \\ n_{n-1}^{k+1} \\ n_n^{k+1} \end{pmatrix} = \begin{pmatrix} d_1^k \\ d_2^k \\ \vdots \\ d_{n-1}^k \\ d_n^k \end{pmatrix} \tag{58}$$

The scheme is always numerically stable and convergent.
Truncation error is $\mathcal{O}\left(h^2\right) + \mathcal{O}\left(\tau^2\right)$.
The algebraic problem is tridiagonal and may be efficiently solved with the tridiagonal matrix algorithm.

Lax–Friedrichs method

Method for the solution of hyperbolic PDEs, e.g., wave equation.
Consider, for example, the advective equation $\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0$.
Simple Forward Time Centered Space scheme

$$\frac{u_i^{k+1} - u_i^k}{\tau} + a\frac{u_{i+1}^k - u_{i-1}^k}{2h} = 0 \qquad (59)$$

is unconditionally unstable.
Substitution of u_i^k by $\frac{1}{2}(u_{i+1}^k + u_{i-1}^k)$ leads to Lax–Friedrichs method

$$\frac{u_i^{k+1} - \frac{1}{2}(u_{i+1}^k + u_{i-1}^k)}{\tau} + a\frac{u_{i+1}^k - u_{i-1}^k}{2h}, = 0 \qquad (60)$$

which is stable for $\frac{|a|\tau}{h} < 1$ (Courant condition).
Back from discrete to continuous: $\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \frac{h^2}{2\tau}\frac{\partial^2 u}{\partial x^2}$
$\Rightarrow$ Lax scheme has a numerical dissipation or viscosity, i.e., short wavelengths are damped out.

Literature

- William F. Ames: *Numerical methods for partial differential equations*
- James William Thomas: *Numerical partial differential equations: finite difference methods*
- William H. Press, Saul Teukolsky, William T. Vetterling und Brian P. Flannery: *Numerical Recipes in C. The Art of Scientific Computing*