Computer Simulations in Physics

Differential equations

Janos Török

Department of Theoretical Physics

February 20, 2020

Finite difference: First order

- Forward difference

$$\Delta f(x) \equiv f(x+h) - f(x) = f_{k+1} - f_k$$

- Backward difference

$$\nabla f(x) \equiv f(x) - f(x-h) = f_k - f_{k-1}$$

- Central difference

$$\delta f(x) \equiv f(x+h/2) - f(x-h/2) = f_{k+\frac{1}{2}} - f_{k-\frac{1}{2}}$$

- Well, if you know it at $k + \frac{1}{2}$

Finite difference: Second order

- Forward difference

$$\Delta^2 f(x) \equiv f_{k+2} - 2f_{k+1} + f_k$$

- Backward difference

$$\nabla^2 f(x) \equiv f_k - 2f_{k-1} + f_{k-2}$$

- Central difference

$$\delta^2 f(x) \equiv f_{k+1} - 2f_k + f_{k-1}$$

Finite difference: General

- Forward difference

$$\Delta^n f(x) \equiv \sum_{i=0}^{n} (-1)^i \binom{n}{i} f_{k-i+n}$$

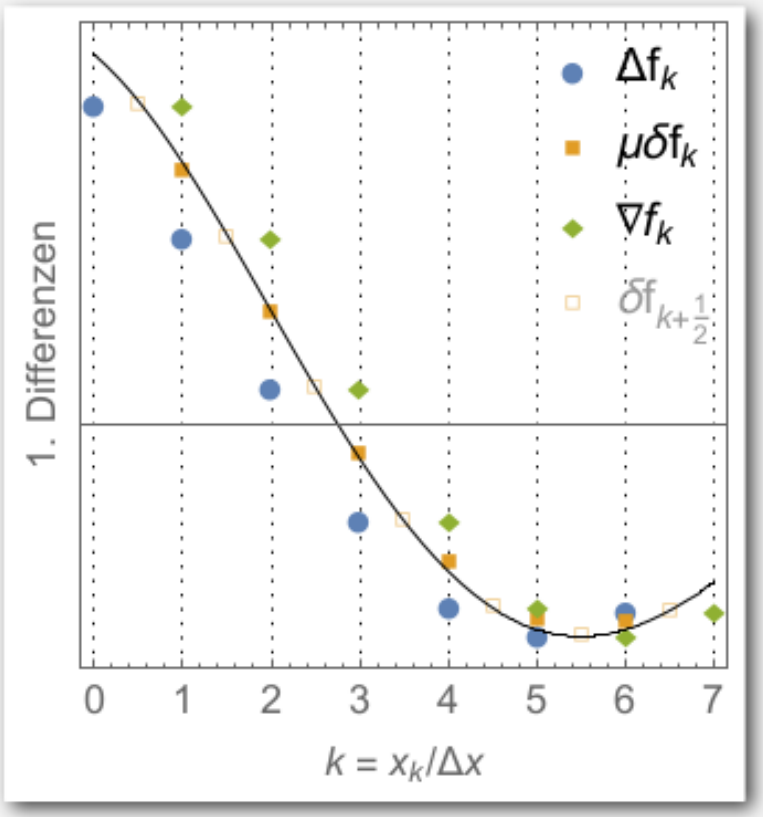
- Backward difference

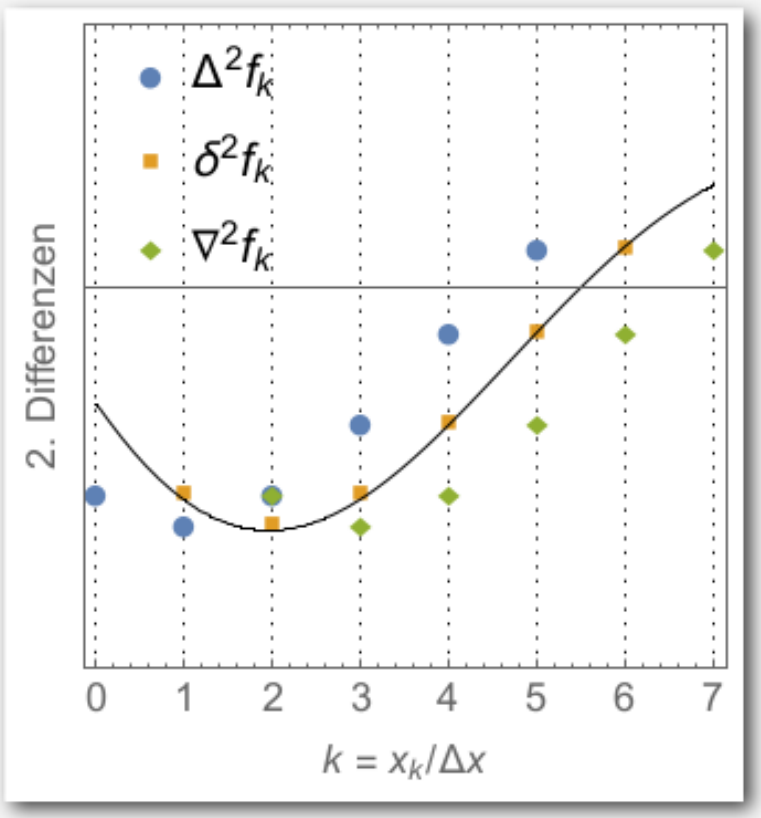
$$\nabla^n f(x) \equiv \sum_{i=0}^{n} (-1)^i \binom{n}{i} f_{k-i}$$

- Central difference

$$\delta^n f(x) \equiv \sum_{i=0}^{n} (-1)^i \binom{n}{i} f_{k-i+n/2}$$

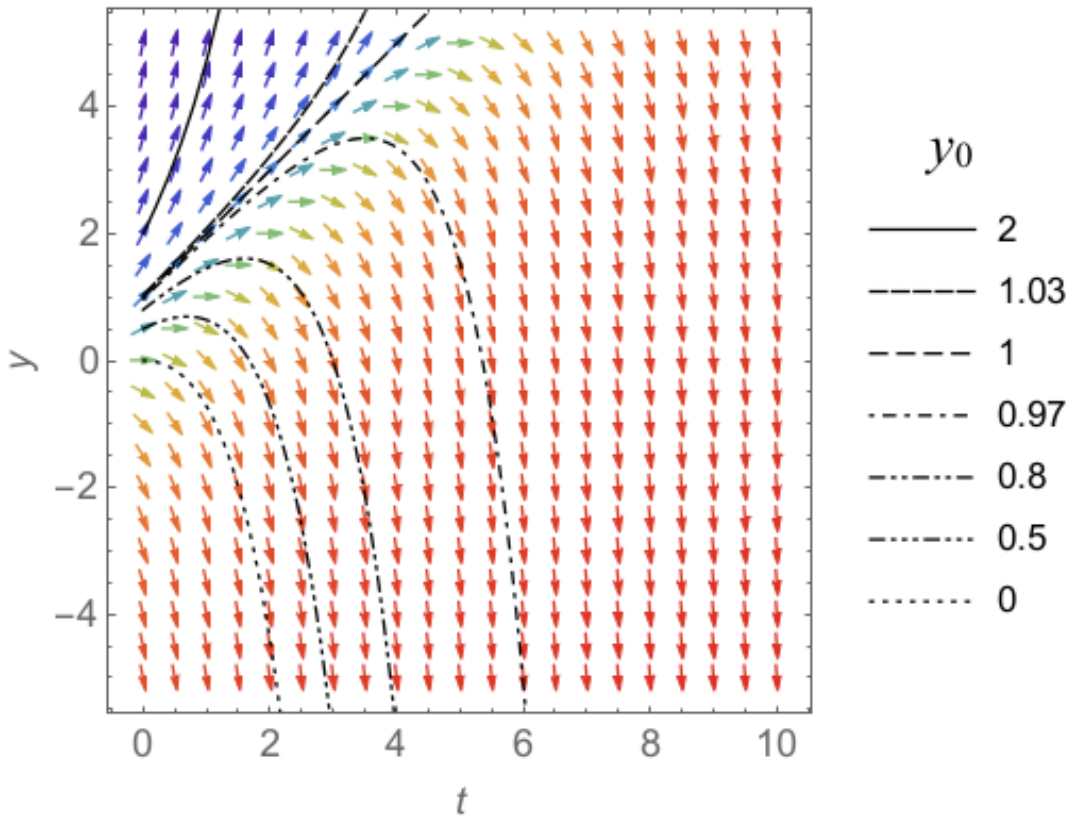
Finite difference: Comparison





First order differential equation

- Example: $y' = y - t$, boundary conditions: $y(0) = y_0$
- Solution: $y(t) = 1 + t + e^t(y_0 - 1)$
- Vector field



Euler method

- Forward difference, Euler method

$$y_{n+1} = y_n + \Delta t f_n + \mathcal{O}(\Delta t^2)$$

- Backward difference, Implicite Euler method

$$y_{n+1} = y_n + \Delta t f_{n+1} + \mathcal{O}(\Delta t^2)$$

- Example, linear function:

$$f(y) = a_0 + a_1 y$$

- Then: $f_{n+1} = f_n + a_1(y_{n+1} - y_n)$
- And the new position can be obtained as:

$$y_{n+1} = \frac{1}{1 - a_1 \Delta t}(y_n + a_0 \Delta t) + \mathcal{O}(\Delta t^2)$$

Euler method

- Second order differential equation:

$$\ddot{y} = f(\dot{y}(t), y(t), t)$$

- First velocity ($v = \dot{y}$)

$$v_{n+1} = v_n + \Delta t \; f_n + \mathcal{O}(\Delta t^2)$$

- Then position

$$y_{n+1} = y_n + \Delta t \; v_n + \mathcal{O}(\Delta t^3)$$

- Do not use it!

Implicit Euler method (backward)

- Second order differential equation:

$$\ddot{y} = f(\dot{y}(t), y(t), t)$$

- First velocity ($v = \dot{y}$)

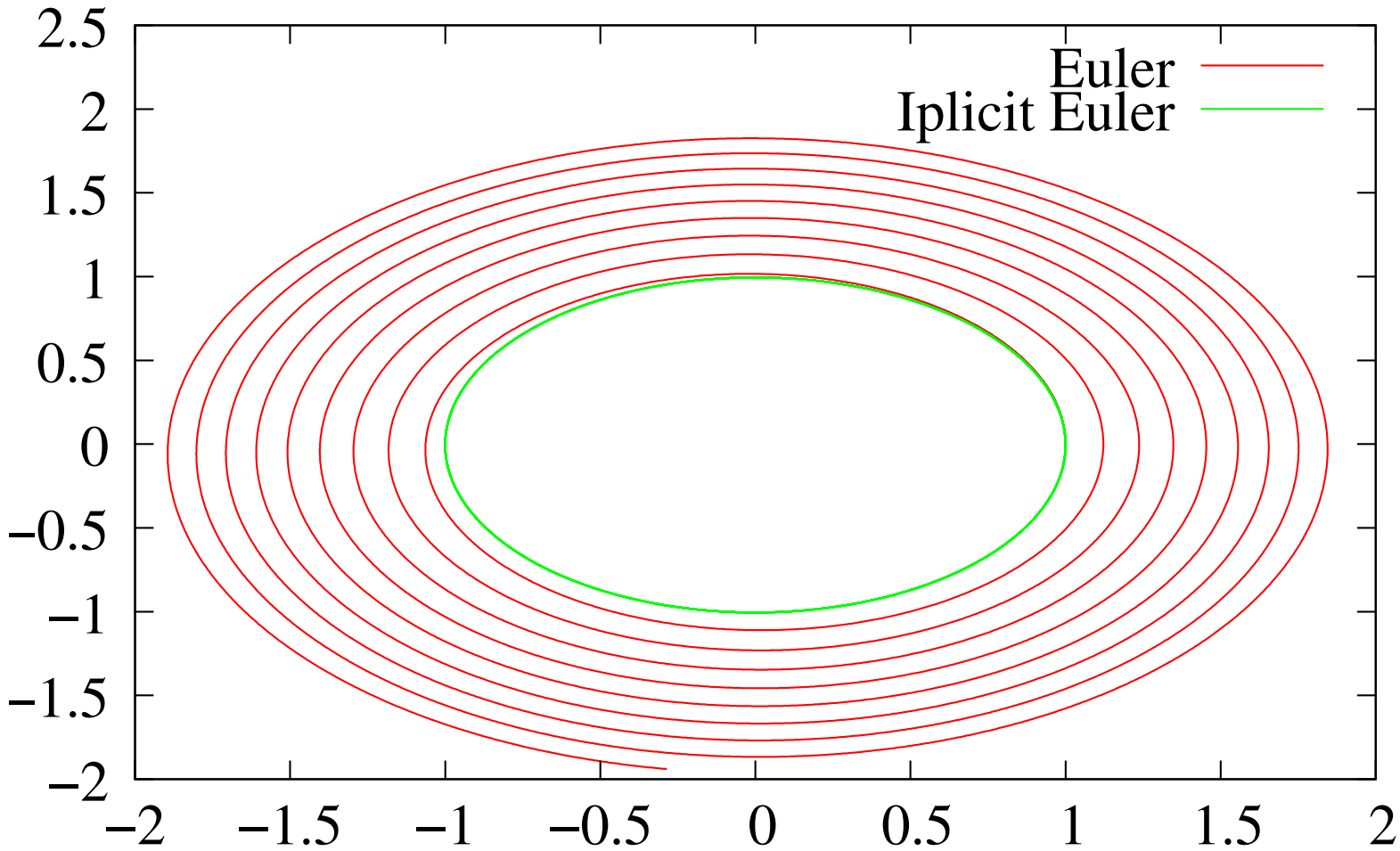
$$v_{n+1} = v_n + \Delta t \; f_n + \mathcal{O}(\Delta t^2)$$

- Then position

$$y_{n+1} = y_n + \Delta t \; v_{n+1} + \mathcal{O}(\Delta t^3)$$

- Surprisingly good!

Euler



Verlet method

- Second order differential equation:

$$\ddot{y} = f(y(t), t)$$

- From central difference

$$y_{n+1} = 2y_n - y_{n-1} + \Delta t^2 \; f_n + \mathcal{O}(\Delta t^4)$$

- Leapfrog

$$y_{n+1} = y_n + \Delta t \; v_{n+\frac{1}{2}}$$
$$v_{n+\frac{1}{2}} = v_{n+\frac{1}{2}} + \Delta t \; f_n$$

- None of them is used
- Velocity dependent forces are difficult to add

Velocity Verlet method

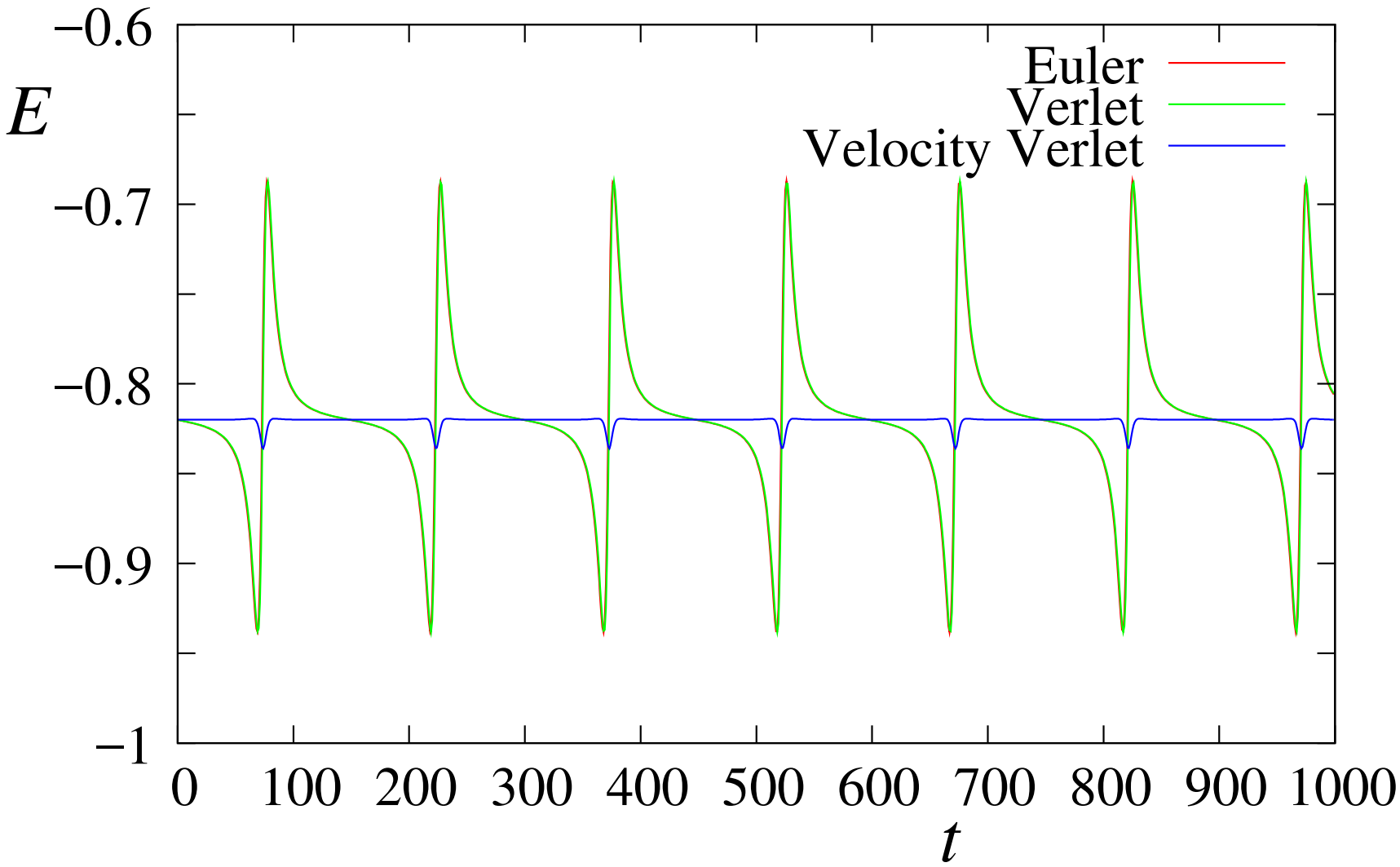
- The one actually used in all codes:

$$y_{n+1} = y_n + \Delta t\ v_n + \frac{1}{2}\Delta t^2 f_n$$

$$v_{n+1} = v_n + \frac{1}{2}\Delta t(f_n + f_{n+1})$$

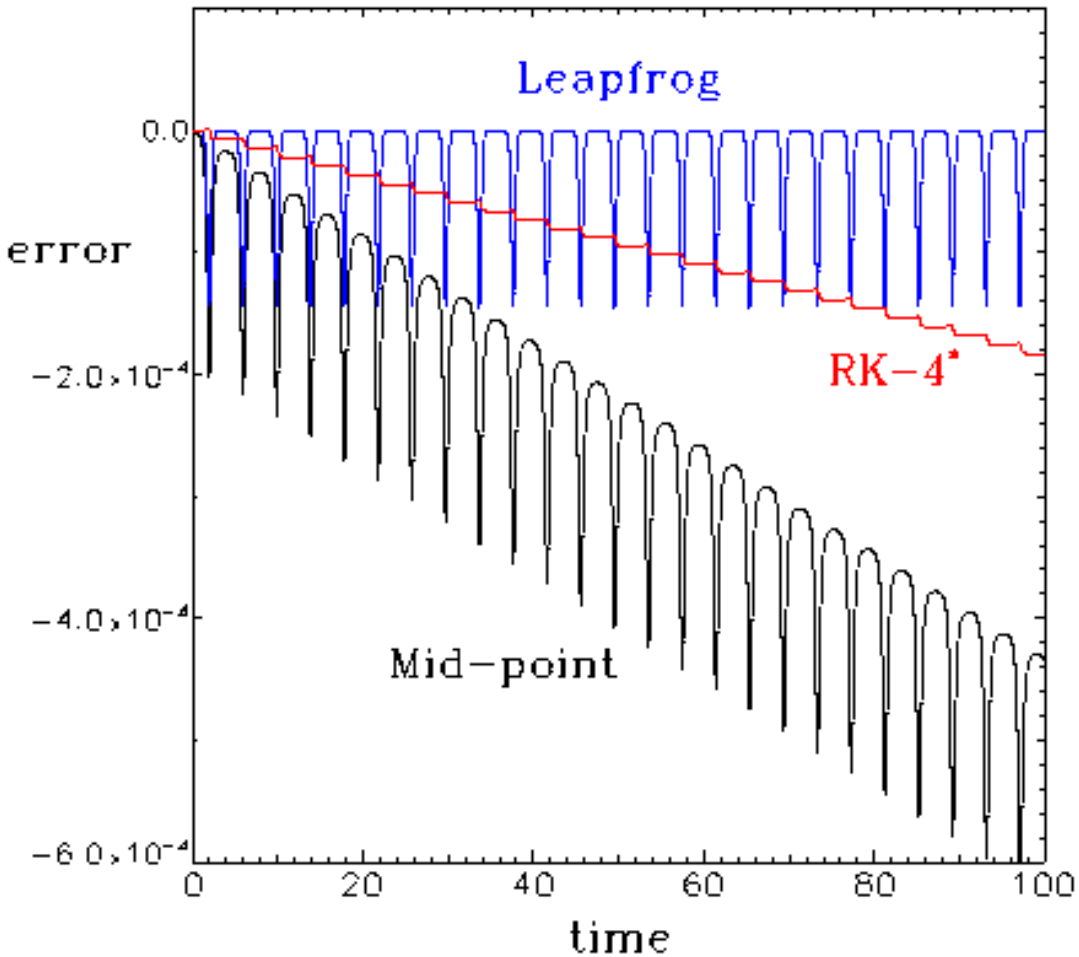
- Implementation
  1. $v_{n+1/2} = v_n + \frac{1}{2} f_n \Delta t$
  2. $y_{n+1} = y_n + \Delta t\ v_{n+1/2}$
  3. Calculate forces
  4. $v_{n+1} = v_{n+1/2} + \frac{1}{2} f_{n+1} \Delta t$
- Simplified implementation (1+4 at once):
  1. $y_{n+1} = y_n + v_n\ \Delta t + \frac{1}{2} f_n \Delta t^2$
  2. Calculate forces
  3. $v_{n+1} = v_n + \frac{1}{2}(f_n + f_{n+1})\Delta t$

Energy comparison



Energy conservation

- Runge-Kutta is not good, neither is the Euler method



Value of the timestep

- Example:

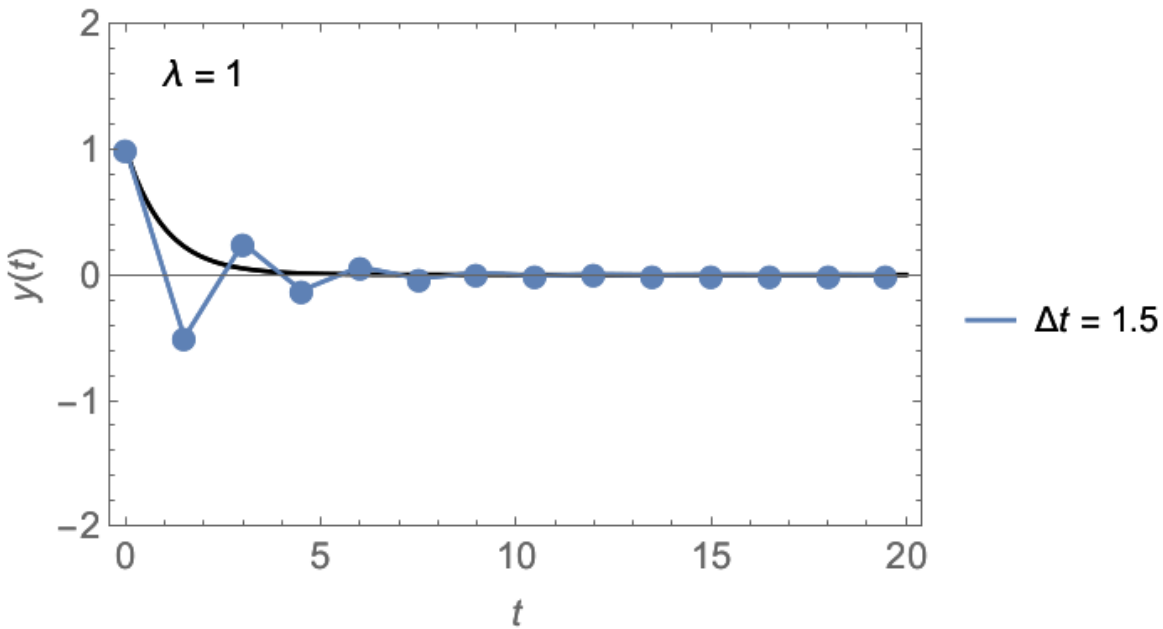
$$\dot{y}(t) = -\lambda y(t)$$

- Euler methor:

$$y_{n+1} = (1 - \lambda \Delta t) y_n$$

- Exact solution:

$$y(t) = y_0 \exp(-\lambda t)$$



- Solution is stable, but what are those oscillations?

Value of the timestep

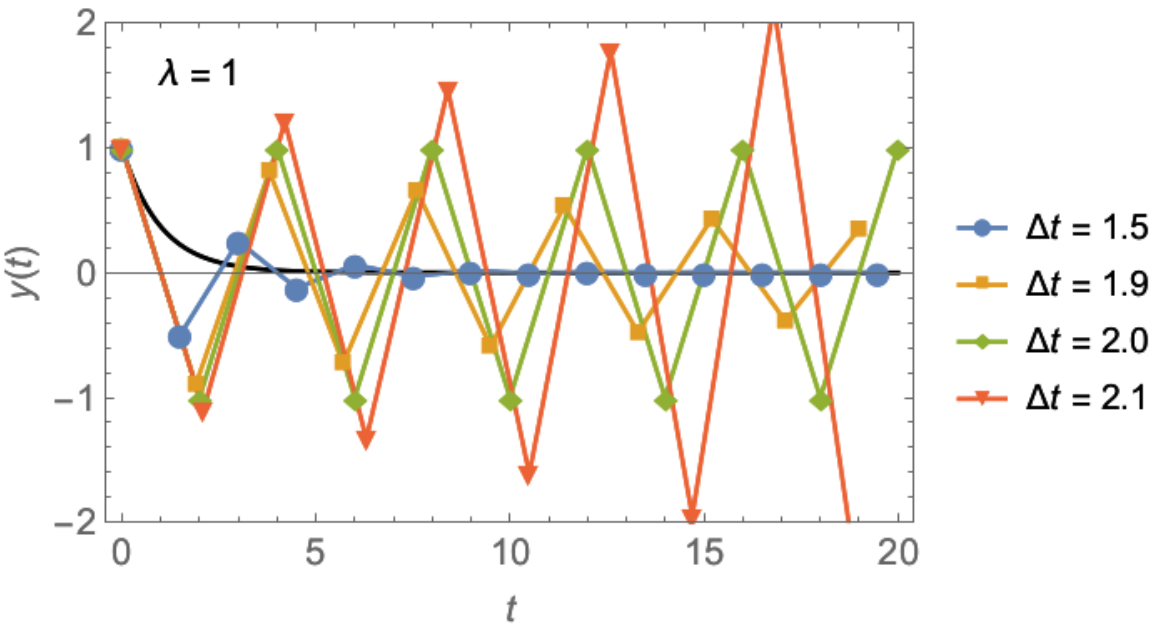
- Example:

$$\dot{y}(t) = -\lambda y(t)$$

- Euler methor:

$$y_{n+1} = (1 - \lambda \Delta t) y_n$$

- If $\lambda \Delta t > 2$ the oscillations increase



Stability

- Describe the iteration with an operator T, error is $\epsilon_n \ll \mathbf{y}_n$

$$\mathbf{y}_{n+1} = T[\mathbf{y}_n] \qquad \mathbf{y}_{n+1} + \epsilon_{n+1} = T[\mathbf{y}_n + \epsilon_n]$$

- First order expansion

$$\epsilon_{n+1} = T[\mathbf{y}_n + \epsilon_n] - T[\mathbf{y}_n] \simeq T'[\mathbf{y}_n]\epsilon_n \equiv G\ \epsilon_n$$

- This is stable for Eigenvalues with $|g_v| < 1$ of G
- Example: relaxation with Euler method

$$T[y_n] = (1 - \lambda\Delta t)y_n$$

- Error propagation matrix:

$$G = T'[y_n] = 1 - \lambda\Delta t$$

- The Eigenvalue: $g_1 = 1 - \lambda\Delta t$, which gives the condition:

$$0 < \lambda\Delta t < 2$$

Stability of the Harmonic oscillator with Euler

- Positon vector $\mathbf{y}(t) = (x(t), v(t))$
- Differential equation:

$$\mathbf{y}(t)' = \mathbf{L} \cdot \mathbf{y}(t)$$

$$\mathbf{L} = \begin{pmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{pmatrix}$$

- Euler method

$$\mathbf{y}_{n+1} = (\mathbf{I} + \mathbf{L}\Delta t)\mathbf{y}_n$$

- Error propagation matrix:

$$\mathbf{G} = \mathbf{I} + \mathbf{L}\Delta t$$

- Eigenvalues:

$$g_{1,2} = 1 \pm i\omega_0 \Delta t$$

- Unfortunately $|g_{1,2}| > 1$

Stability of the Harmonic oscillator with Velocity Verlet

- Positon vector $\mathbf{y}(t) = (x(t), v(t))$
- Differential equation:

$$\mathbf{L} = \begin{pmatrix} 1 - \frac{1}{2}\omega_0^2 \Delta t^2 & \Delta t \\ -\Delta t \omega_0^2 (1 + \frac{1}{4}\Delta t^2 \omega_0^2) & 1 - \frac{1}{2}\Delta t^2 \omega_0^2 \end{pmatrix}$$

- Eigenvalues:

$$|g_{1,2}| = 1$$

Stability of other methods

- The implicite Euler with relaxation:

$$T[y_n] = (1 + \lambda \Delta t)^{-1} y_n$$

- Which gives $0 < \lambda \Delta t$, always stable!
- Leapfrog:

$$y_{n+1} = -2\lambda \Delta t y_n + y_{n-1}$$

- The same applies for the error

$$\begin{pmatrix} \epsilon_{n+1} \\ \epsilon_n \end{pmatrix} = \begin{pmatrix} -2\lambda \Delta t & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \epsilon_n \\ \epsilon_{n-1} \end{pmatrix}$$

- Which gives:

$$g_{1,2} = -\lambda \Delta \pm \sqrt{1 + (\lambda \Delta t)^2}$$

- Unfortunately $|g_2| > 1$

Partial differential equations

$$c^2 \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial t^2} = f(x,t)$$

$$D \frac{\partial^2 u}{\partial x^2} - \frac{\partial u}{\partial t} = f(x,t)$$

$$\boldsymbol{\Delta} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$

$$\boldsymbol{\Delta} u(x,y,z,t) = -\rho(x,y,z,t)$$

$$\frac{\hbar^2}{2m} \boldsymbol{\Delta} u(x,y,z) + E - V(x,y,z) = 0$$

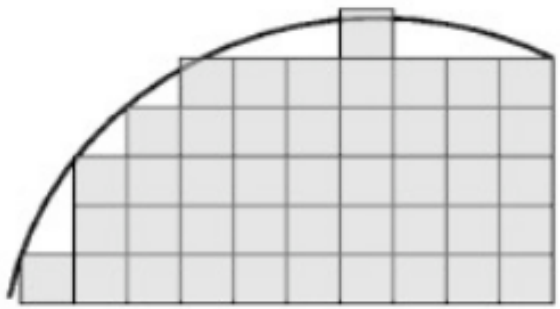
$$\frac{\hbar^2}{2m} \boldsymbol{\Delta} u + i\hbar \frac{\partial u}{\partial t} - V = 0$$

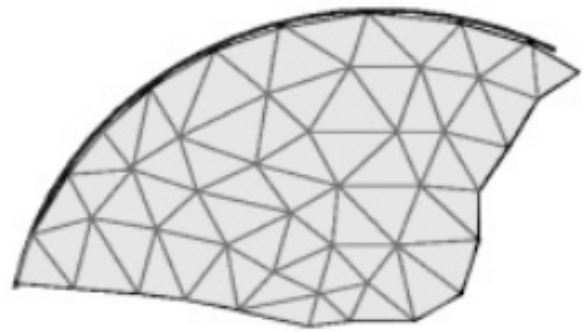
Partial differential equations: Problems

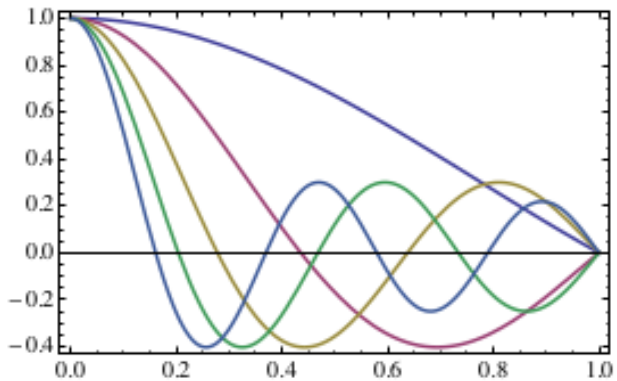
- Boundary problems:
  - Typically time independent systems
  - Values are given on a surface, and solution is search for inside the volume
  - e.g. Poisson problem
- Starting value problems:
  - Typically time dependent systems
  - Start conditions are known, time evolution is searched for
  - e.g. Newton equations

Discretization

- Discretization of
  - Derivative
  - Space (mesh)
  - Basis function







Discretization on a two-dimensional lattice

- Poisson equation:

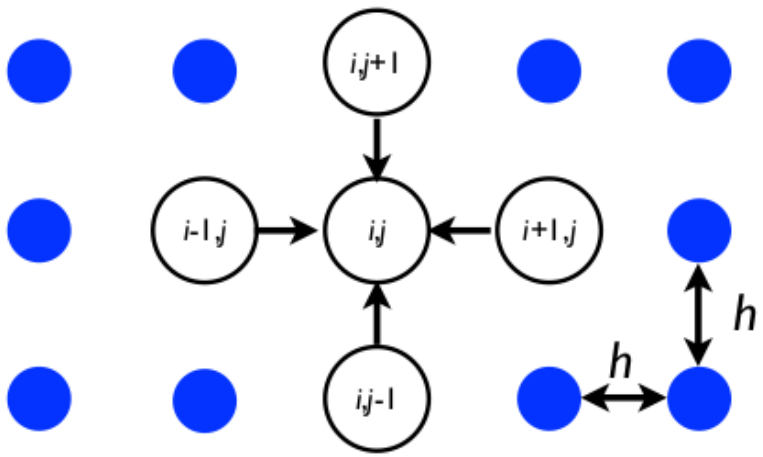
$$\frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} = -\rho(x,y)$$

- Central derivative

$$-\rho_{i,j} = \frac{1}{h^2}(u_{i+1,j} + u_{i-1,j} - 4u_{i,j} + u_{i,j+1} + u_{i,j-1})$$

$$u_{i,j} = \frac{1}{4}(h^2\rho_{i,j} + u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1})$$

- The new value of the grid does not depend on itself!



Matrix formulation

- 1d:

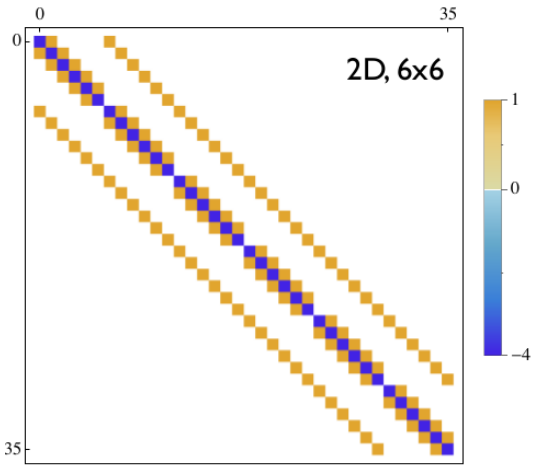
$$u_{i+1} - 2u_i + u_{i-1} = -\rho_i h^2$$

- System of linear equations, solve it!

$$A_{ij} = \begin{cases} 1 & \text{if } |i-1| = j \\ -2 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

- 2d: $(i = 0, \ldots N-1,\ j = 0, \ldots M-1)\ r = iM + j$

$$v_{r-M} + v_{r-1} - 4v_r + v_{r+1} + v_{r+M} = -\rho_i h^2$$



Matrix formulation

- Problem

$$A\mathbf{v} = \mathbf{b}$$

- Gauss elimination is not a good idea
- Sparce matrices
- Special sparce matric methods
- Iterative methods
- LU decomposition: $A = L - zI + U$
- Jacobi approximation

$$\begin{aligned}\vec{v}_{\text{new}} &= \frac{1}{z}\left[(\mathbf{L}+\mathbf{U})\vec{v}_{\text{old}} + h^2\vec{\rho}\right] \\ &= \vec{v}_{\text{old}} + \vec{r}_{\text{old}} + \frac{1}{z}h^2\vec{\rho}\end{aligned}$$

- Gauss-Seidel approximation

$$\begin{aligned}\vec{r}_{\text{GS}} &= \frac{1}{z}(\mathbf{L}\vec{v}_{\text{new}} + \mathbf{U}\vec{v}_{\text{old}}) - \vec{v}_{\text{old}} \\ \vec{v}_{\text{new}} &= \vec{v}_{\text{old}} + \vec{r}_{\text{GS}} + \frac{1}{z}h^2\vec{\rho}\end{aligned}$$

Partial differential equations: Boundary values

- Dirichlet problems:
  - Values of the function is known on the surface
- Neumann problem
  - Derivative of the function is known on the surface
- Cauchy problem
  - Alternatively derivative or the value of the function is known on the surface
- Periodic boundary
  - Same value, zero derivative on both sides
  - e.g. crystal potential

Partial differential equations: Boundary values

- Dirichlet problems:
  - Do not update boundary points
  - In matrix formulation move values to the constant part
- Neumann problem
  - Boundary points are inactive for the dynamics
  - The value is changed however if the corresponding inner grid changes its value to keep derivative constant
  - For matrix method new fictional points
  - Equation for derivative

Fourier transform

- The differential equation

$$D[y(t)] = f(t)$$

- The Fourier transform of the Green's function of $\mathcal{F}[D] = G(\omega)$
- The Fourier transform of $\mathcal{F}[f(t)] = F(\omega)$

$$y(t) = \mathcal{F}^{-1}[G(\omega)F(\omega)]$$

Order of update

- Random: pick a node randomly and update the value at that point
- Random sequential: at every step shuffle the order of the nodes, and update each in the given order
- Parallel: solution is done to a separate array simultaneously
  - Can be done in parallel
  - Stability problems may arise!
  - Example, Laplace equation 1d:

$$u_i = \frac{1}{2}(u_{i-1} + u_{i+1})$$

  - State $\mathbf{u} = \{1, -1, 1, -1, 1, -1, \dots\}$
  - After parallel update: $\mathbf{u} = \{-1, 1, -1, 1, -1, 1, \dots\}$

Units

- Computer stores only numbers
- We have to keep in mind the units
- Better to facilitate our life
- e.g. Damped harmonic oscillator

$$m\partial_t^2 x + \gamma \partial_t x + kx = 0$$

- Units/values:

$$m = m' \cdot [m], \quad x = x' \cdot [x], \quad t = t' \cdot [t]$$

  where [.] is the unit of the quantity
- SI units: kg, m, s

Units

- Parameters:

$$[m] = [m], \quad \gamma = \frac{[m]}{[t]}, \quad [k] = \frac{[m]}{[t]^2}$$

- Boundary conditions

$$[x_0] = [x], \quad [v_0] = \frac{[x]}{[t]}$$

- Possible choice

$$[m] = m, \quad [x] = x_0, \quad [t] = \sqrt{m/k}$$

- This gives

$$m' = 1, \quad x_0' = 1$$

Units

- Dimensionless equation:

$$\partial_{t'}^2 x' + \frac{\gamma}{\sqrt{km}} \partial_{t'} x' + x' = 0$$

- This gives us two control parameters:

$$\Gamma = \frac{\gamma}{\sqrt{km}}, \quad v_0' = \frac{v_0}{x_0} \sqrt{\frac{m}{k}}$$

Units: example

- Gravitational potential

$$V(r) = -\frac{\alpha m}{r}$$

- Parameters:

$$[m] = [m], \quad [\alpha] = \frac{[x]^3}{[t]^2}, \quad [x_0] = [x], \quad [v_0] = \frac{[x]}{[t]}$$

- Natural units

$$[m] = m, \quad [x] = x_0, \quad [t] = \sqrt{\frac{x_0^3}{\alpha}}$$

- Control parameter:

$$v_0' = v_0 \frac{[t]}{[x]} = v_0 \sqrt{\frac{x_0}{\alpha}}$$

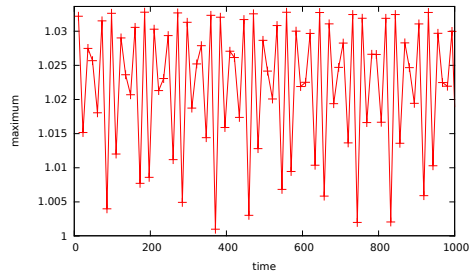
References

- https://www.cs.colorado.edu/~lizb/chaos/ode-notes.pdf
- http://faculty.olin.edu/bstorey/Notes/DiffEq.pdf

Practice

- **Oscillator:** Solve numerically the following differential equation, starting from $x(0) = 0$, $\dot{x}(0) = 1$ using different integrators:

$$\ddot{x} = -\gamma x$$



Measure the positions of the maxima:

- **Poisson equation:** Solve the following differential equation using iterative technique:

$$\nabla^2 u(x, y) = -\rho(x, y)$$

Choose $\delta x = \delta y = 1$, $L = 10$, $\rho(x, y) = 0$. The boundary is zero everywhere, except for $u(x, L) = 1$. Change only $\rho(4, 4) = a$, Solve the problem again. For plotting surfaces you can use gnuplot (splot "datafile"). Datafile format: "x y z" and one empty line between rows (before x changes)

- **Vector field:** Reproduce the figure at page 6