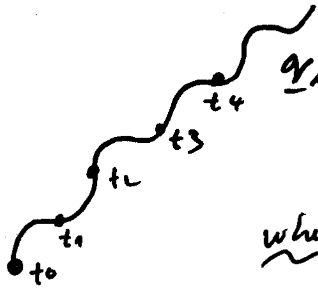
Ordinary differential equations:

- **Motivation:** Newton dynamics



$$m_i \ddot{\underline{q}}_i = \underline{F}_i \; ; \quad m_i: \text{ mass of particle } i$$

$\underline{F}_i$: force on particle i

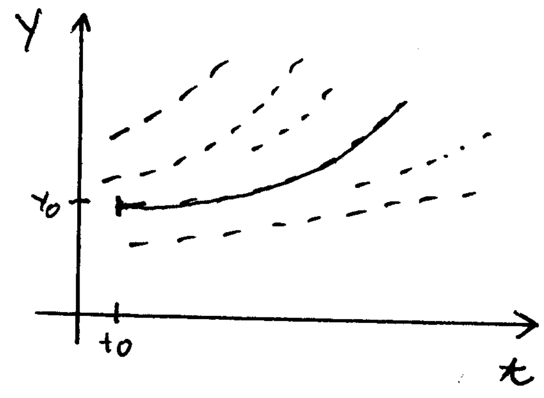
where: $\underline{F}_i = -\dfrac{\partial U}{\partial \underline{q}_i}$; U: potential energy

Determine trajectories $\underline{q}_i(t)$

$\Longrightarrow$ Molecular dynamics

- **Theoretical background:**

$$\frac{dy}{dt} = f(t, y)$$



- for each pair (t, y), there is a unique $\dfrac{dy}{dt}$

  $\Longrightarrow$ draw slopes

- find $y(t)$ which passes through initial value $y_0(t_0)$

- higher order ODE's:

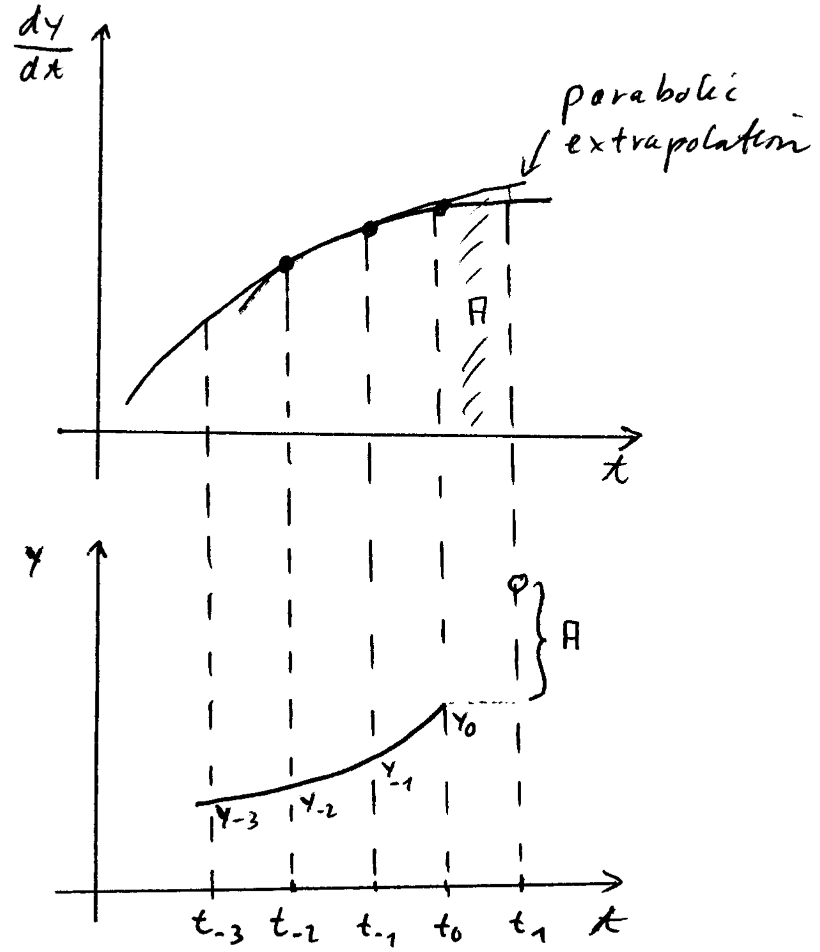
$$y^{(4)} = f(x, y(x), y'(x), y''(x), y^{(3)}(x))$$

set:

$$\left.\begin{aligned} y_1(x) &= y(x) \\ y_2(x) &= y'(x) \\ y_3(x) &= y''(x) \\ y_4(x) &= y^{(3)}(x) \end{aligned}\right. \quad \Longrightarrow \quad \left.\begin{aligned} y_1'(x) &= y_2(x) \\ y_2'(x) &= y_3(x) \\ y_3'(x) &= y_4(x) \\ y_4'(x) &= f(x, y_1, y_2, y_3, y_4) \end{aligned}\right\} \text{ system of first order ODE's !!!}$$

• Predictor-corrector methods:

Procedure:



$$\rightarrow y_{1p} - y_a = \int_{t_a}^{t_1} \frac{dy}{dt}\, dt = A$$

y_{1p}: predicted value of y_1

fit parabola through y'_{-2}, y'_{-1}, y'_0 and integrate

$\rightarrow$ Determine: $\quad y_1' = f(t_1, y_{1p})$

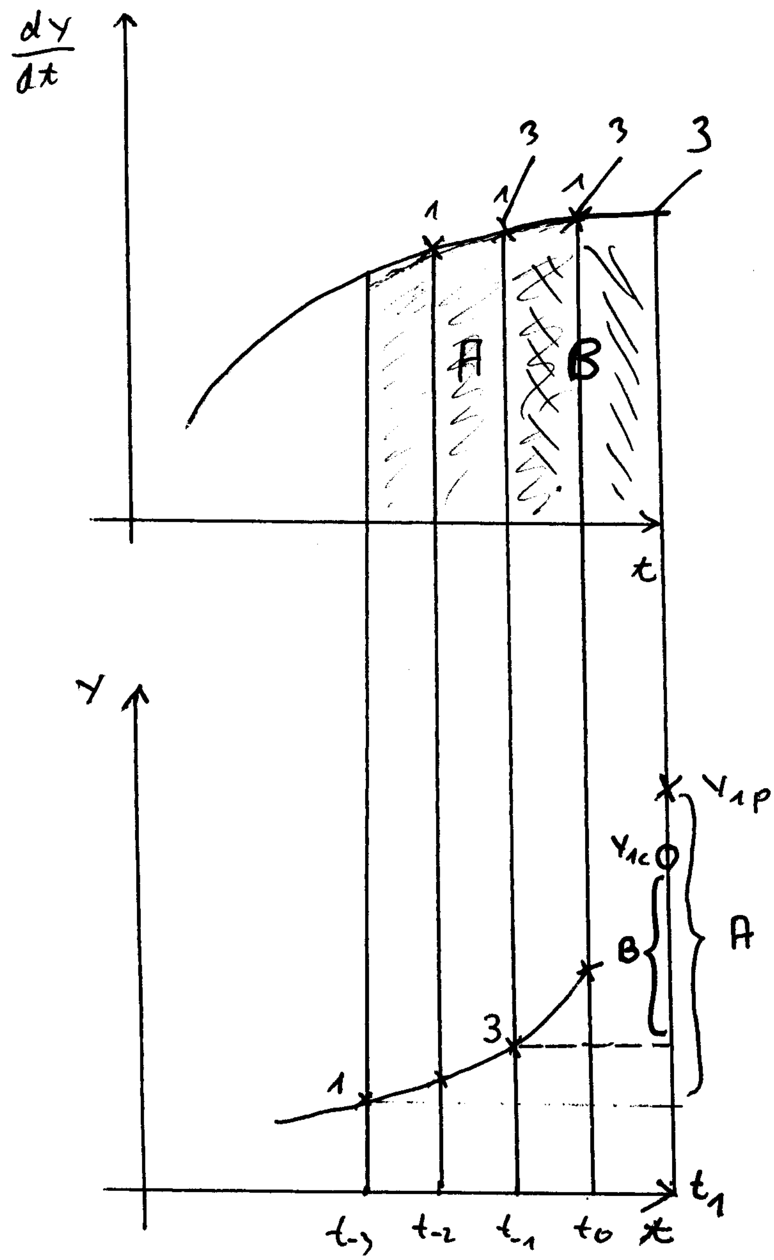
$\rightarrow$ Fit parabola through y_1', y_0', y'_{-1} and integrate

y_{1c}: corrected value of y_1

$\rightarrow$ Criteria: $\quad \dfrac{|y_{1p} - y_{1c}|}{20}$ neither too large nor too small

$\rightarrow$ Determine: $\quad y_1' = f(t_1, y_{1c})$

Milne predictor-corrector step



1. Fit parabola to the latest three points and integrate:

$$Y_{1P} = Y_{-3} + \frac{4h}{3}\left(2Y'_{-2} - Y'_{-1} + 2Y'_{0}\right)$$

(extrapolate)

2. Determine:

$$Y_1' = f(t_1, Y_{1P})$$

3. Fit another parabola to the latest three points and integrate:

$$Y_{1C} = Y_{-1} + \frac{h}{3}\left(Y'_{-1} + 4Y'_{0} + Y'_{1}\right)$$

(interpolate)

4. Test:

$$\frac{|Y_{1P} - Y_{1C}|}{20}$$ neither too large, nor too small

5. Determine:

$$Y_1' = f(t_1, Y_{1C})$$

Equations of higher order:

$$y'' = f(t, y, y')$$
$$y(a) = c$$
$$y'(a) = d$$

Procedure:

→ extrapolation (= prediction) is taking place in highest derivative:

- fit parabola, determine A
- $y'_{1P} = y'_{-3} + \frac{4h}{3}(2y''_{-2} + 2y''_0 - 2y''_{-1})$ (y')

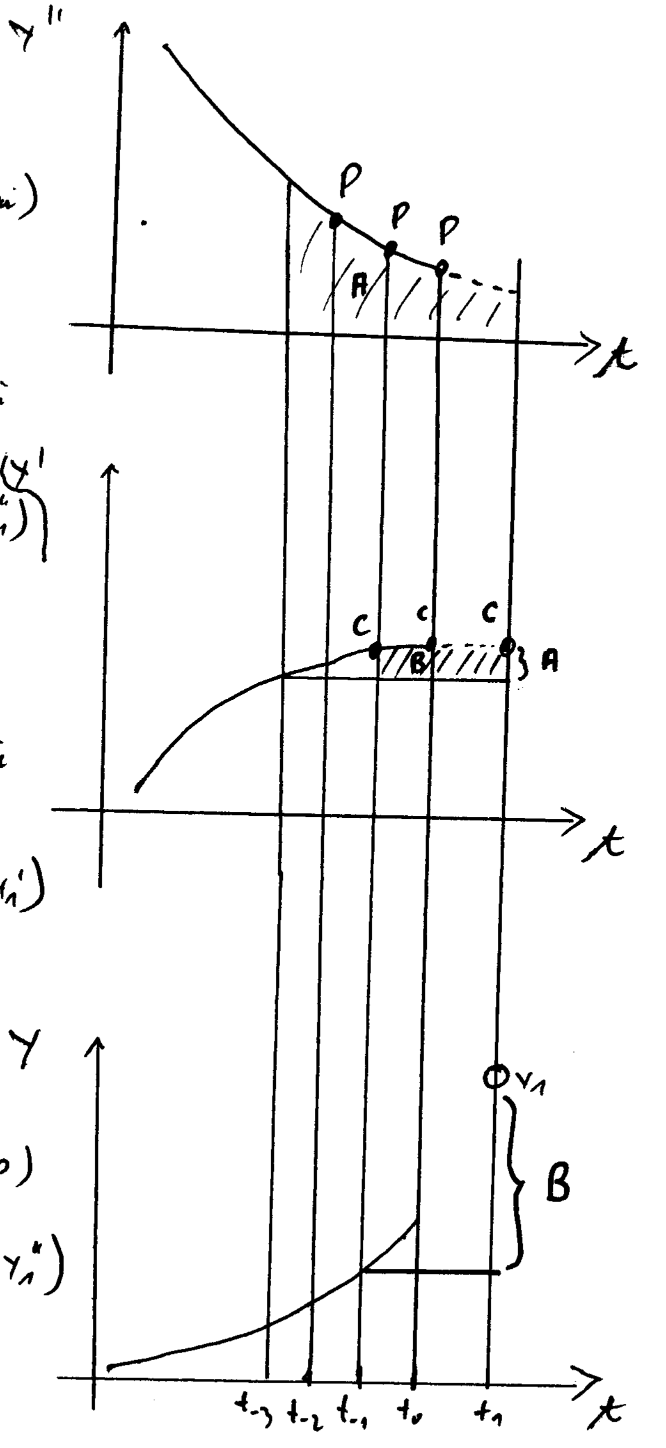
→ interpolation (= corrector) in y' - picture

- fit parabola, determine B
- $y_{1C} = y_{-1} + \frac{h}{3}(y'_{-1} + 4y'_0 + y'_1)$

→ insert y_1, y_1' to get y''_1 $\left(\begin{array}{l} y_1' = y'_{1P} \\ y_1 = y_{1C} \end{array}\right)$

→ interpolation (corrector step)

$$y'_{1C} = y'_{-1} + \frac{h}{3}(y''_{-1} + 4y''_0 + y''_1)$$



→ criterion:

$$\frac{|y'_{1P} - y'_{1C}|}{20}$$

not too small, and not too large!

→ interpolation (corrector step)

$$y_{1C} = y_{-1} + \frac{h}{3}(y'_{-1} + 4y'_0 + y'_1)$$

→ insert $y_{1(C)}, y'_{1(C)}$ to get y''_1

Richardson integration

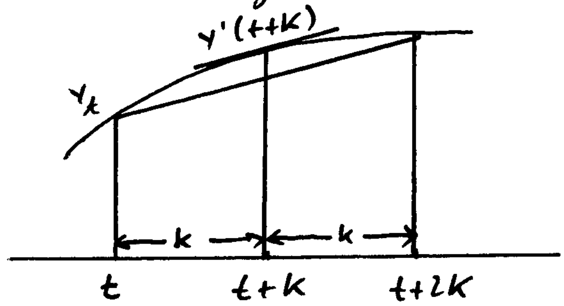
Idea: integrators that extrapolate on h.

Procedure:
- assemble sequence of estimates of y_1 at t_1 using first h, then $h/2$ and even smaller subdivisions of basic step size.
- extrapolate to a y_1 corresponding to $h \to 0$

– Bulirsch / Stoer:

$K = \frac{h}{2j}$ j integral $\}$ even subdivisions of h

Basic substep: linear extrapolation of y from t to $t+2K$, using the slope at $t+k$



$$y(t+2k) = y(t) + 2K \cdot y'(t+k)$$

$\Rightarrow$ use $y(t+2k)$ to determine $y'(t+2k)$ from:

$$y'(t) = f(t, y(t))$$

$\Rightarrow$ with preceding value of y, we determine a new y etc.

First step: $\quad y(K) = y_0 + K\, y_0'$

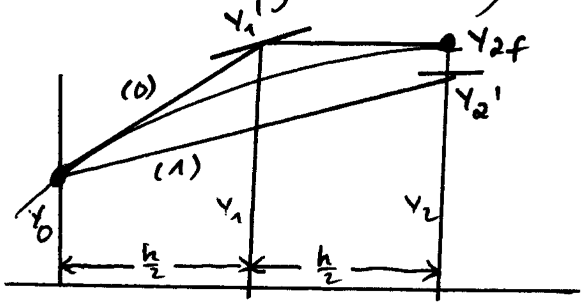
Last step: $\quad y_n(K) = y_{n-1} + K\, y_n'$

Examples: $\boxed{j = 1}$

$$(0)\ Y_1 = Y_0 + k Y_0' \Rightarrow Y_1'$$

$$(1)\ Y_2 = Y_0 + 2k Y_1' \Rightarrow Y_2'$$

$$(2)\ Y_{2f} = Y_1 + k Y_2' \quad \text{(or averaged value)}$$



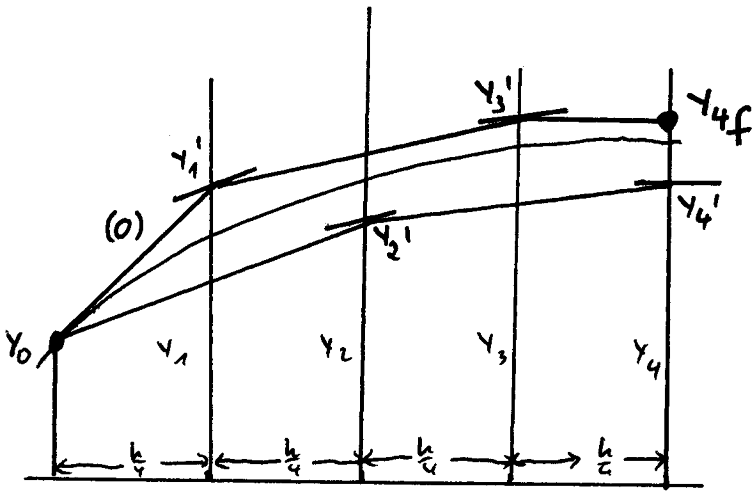
$\boxed{j = 2}$

$$(0)\ Y_1 = Y_0 + k Y_0' \Rightarrow Y_1' \quad \text{(starting step)}$$

$$\left.\begin{array}{l}(1)\ Y_2 = Y_0 + 2k Y_1' \Rightarrow Y_2' \\ (2)\ Y_3 = Y_1 + 2k Y_2' \Rightarrow Y_3' \\ (3)\ Y_4 = Y_2 + 2k Y_3' \Rightarrow Y_4'\end{array}\right\} \text{general steps}$$

$$(4)\ Y_{4f} = Y_3 + k Y_4' \quad \text{(ending step)}$$

averaged value: $Y(t_0 + h) = \dfrac{Y_4 + Y_{4f}}{2}$



Molecular dynamics:

$$\left.\begin{aligned} \ddot{\mathbf{r}}_i(t) &= \mathbf{F}_i(t)/m_i \\ \mathbf{F}_i(t) &= -\frac{\partial \mathcal{U}}{\partial \mathbf{r}_i} \end{aligned}\right\} \text{Newton dynamics}$$

– Leapfrog algorithm:

$$\mathbf{r}_i(t+h) = \mathbf{r}_i(t) + h\,\dot{\mathbf{r}}_i(t+h/2)$$

$$\dot{\mathbf{r}}_i(t+h/2) = \dot{\mathbf{r}}_i(t-h/2) + \frac{h}{m_i}\mathbf{F}_i(t)$$

$$\dot{\mathbf{r}}_i(t) = \frac{\dot{\mathbf{r}}_i(t-h/2) + \dot{\mathbf{r}}_i(t+h/2)}{2} \quad \text{(auxiliary construction)}$$

– Verlet algorithm:

$$\mathbf{r}_i(t+h) = 2\,\mathbf{r}_i(t) - \mathbf{r}_i(t-h) + h^2\,\ddot{\mathbf{r}}_i(t)$$

$$\dot{\mathbf{r}}_i(t) = \frac{\mathbf{r}_i(t+h) - \mathbf{r}_i(t-h)}{2h} \quad \text{(auxiliary constructions)}$$

– Velocity verlet algorithm:

$$\left.\begin{aligned} \mathbf{r}_i(t+h) &= \mathbf{r}_i(t) + h\,\dot{\mathbf{r}}_i(t) + \frac{h^2}{2m_i}\mathbf{F}_i(t) \\ \dot{\mathbf{r}}_i(t+h) &= \dot{\mathbf{r}}_i(t) + \frac{h}{2m_i}\left[\mathbf{F}_i(t) + \mathbf{F}_i(t+h)\right] \end{aligned}\right\} \text{no previous information necessary}$$

– Gear algorithm:

Predictor step:

$$\mathbf{r}_{ip}(t+h) = \mathbf{r}_i(t) + \dot{\mathbf{r}}_i(t)h + \frac{h^2}{2}\ddot{\mathbf{r}}_i(t) + \frac{h^3}{6}\dddot{\mathbf{r}}_i(t)$$

$$\dot{\mathbf{r}}_{ip}(t+h) = \dot{\mathbf{r}}_i(t) + h\,\ddot{\mathbf{r}}_i(t) + \frac{h^2}{2}\dddot{\mathbf{r}}_i(t)$$

$$\ddot{\mathbf{r}}_{ip}(t+h) = \ddot{\mathbf{r}}_i(t) + h\,\dddot{\mathbf{r}}_i(t)$$

Forces: $\mathbf{r}_{ip}(t+h) \longrightarrow \mathbf{F}_i \longrightarrow \ddot{\mathbf{r}}_{ic}(t+h)$

Corrector step:

$$\mathbf{r}_{ic}(t+h) = \mathbf{r}_{ip}(t+h) + \frac{h^2}{12}\underbrace{\left(\ddot{\mathbf{r}}_{ic}(t+h) - \ddot{\mathbf{r}}_{ic}(t+h)\right)}_{\Delta \mathbf{a}_i}$$

$$\dot{\mathbf{r}}_{ic}(t+h) = \dot{\mathbf{r}}_{ip}(t+h) + \frac{5h}{12}\Delta \mathbf{a}_i$$

$$\dddot{\mathbf{r}}_{ic}(t+h) = \dddot{\mathbf{r}}_i(t) + \frac{1}{h}\Delta \mathbf{a}_i$$

- Constraints:

Temperature: $\tau = \ell \Delta t$ } measuring period

$$\left\langle \sum_{i=1}^{N} \frac{m_i}{2} (\dot{\underline{r}}_i)^2 \right\rangle = \frac{1}{\tau} \sum_{n=1}^{\ell} \frac{m_i}{2} (\dot{\underline{r}}_i(n\Delta t))^2$$

$$= \frac{3}{2} N k_B T$$

Velocity rescaling:

$$\dot{\underline{r}}_i \Rightarrow \dot{\underline{r}}_i + \xi \left(\frac{T_0}{T} - 1\right) \dot{\underline{r}}_i \; ; \quad \xi = h / 2\underset{\substack{\downarrow \\ \text{relaxation time}}}{\tau}$$

$$\dot{\underline{r}}_i \Rightarrow \dot{\underline{r}}_i \sqrt{\frac{T_0}{T}}$$

$\Rightarrow$ Keeps system at temperature T_0

Pressure: additional dynamics for volume V

$$\ddot{V} = (\mathcal{P} - P_0) / Q \leftarrow \text{"mass"}$$

Potential: $\vartheta = P_0 V$ → desired pressure

$$\mathcal{P} = \rho k_B \mathcal{T} + \mathcal{W}/V$$

$$\mathcal{T} = \frac{1}{3 N k_B} \sum_{i=1}^{N} |\dot{\underline{r}}_i|^2 \frac{m_i}{2}$$

$$\mathcal{W} = \frac{1}{3} \sum_{i=1}^{N} \underline{r}_i \cdot \underline{F}_i$$

$$\Rightarrow PV = \underbrace{N k_B T}_{\text{see above}} + \langle \mathcal{W} \rangle$$

Partial differential equations (PDE's)

- **Problem:** Determine $u(x,y)$ in $G \subset \mathbb{R}^2$ which solves a partial differential equation of second order:

$$\boxed{A u_{xx} + 2B u_{xy} + C_{yy} + D u_x + E u_y + F u = H}$$

with $A^2 + B^2 + C^2 \neq 0$.

Classification:

- $AC - B^2 > 0 \quad \forall (x,y) \in G$
  elliptic PDE's

- $AC - B^2 < 0 \quad \forall (x,y) \in G$
  hyperbolic PDE's

- $AC - B^2 = 0 \quad \forall (x,y) \in G$
  parabolic PDE's

Boundary conditions: boundary Γ

- $u = \varphi$ on Γ (Dirichlet)
- $\dfrac{\partial u}{\partial n} = \gamma$ on Γ (Neumann)
- $\dfrac{\partial u}{\partial n} + \alpha u = \beta$ on Γ (Cauchy)

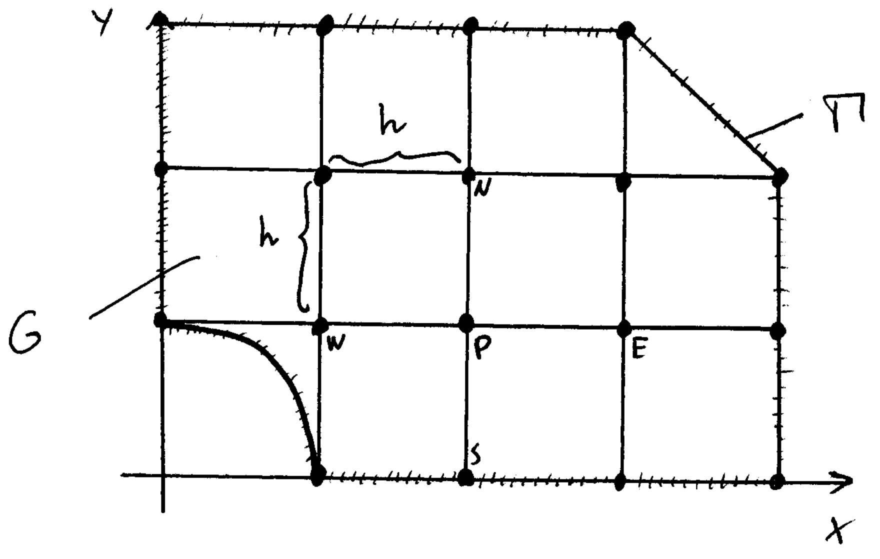
- Examples:

$$\boxed{\begin{array}{ll} u_{xx} + u_{yy} = 0 & \text{Laplace eq.} \\ u_{xx} + u_{yy} = f(x,y) & \text{Poisson eq.} \end{array}}$$

- Be wise, discretize....

(We discuss the numerical solution of the Laplace- and the Poisson equation.)

Step 1: $u(x,y)$ will be represented by its functional values on a set of points in G and Γ.



regular grid

$u(x_i, x_j) \longrightarrow w_{ij}$ (notation)

other grids:
- triangular
- hexagonal
- adaptive

Step 2: Approximate the PDE using $u_{i,j}$.

Example: $P = (x_i, y_i)$, regular grid (width h)

$$
\begin{cases}
u_x(x_i, y_i) \approx \dfrac{u_{i+1,j} - u_{i-1,j}}{2h} \\[2ex]
u_y(x_i, y_i) \approx \dfrac{u_{i,j+1} - u_{i,j-1}}{2h} \\[2ex]
u_{xx}(x_i, y_i) \approx \dfrac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \\[2ex]
u_{yy}(x_i, y_i) \approx \dfrac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2}
\end{cases}
$$

Intuitive notation:

$u_P = u_{i,j}$; $u_N = u_{i,j+1}$;

$u_W = u_{i-1,j}$; $u_S = u_{i,j-1}$;

$u_E = u_{i+1,j}$;

$\Rightarrow$ Poisson - Eq.

$$\frac{u_E - 2u_P + u_W}{h^2} + \frac{u_N - 2u_P + u_S}{h^2} = f_P$$

$(f_P = f(x_i, y_j))$

or: $4u_P - u_N - u_W - u_S - u_E + h^2 f_P = 0$

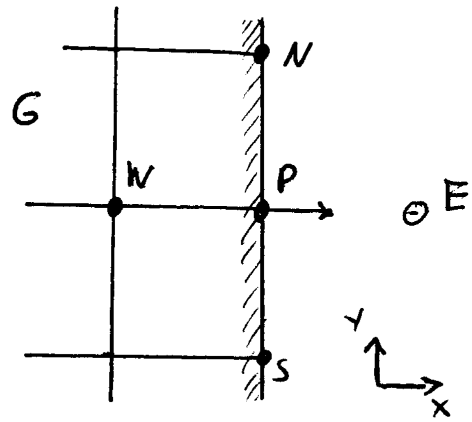
$$\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix} \odot u + h^2 f_P = 0$$

Step 3: boundary conditions

simplest case: regular grid + Dirichlet boundary conditions on Γ

$\Rightarrow$ solve for all inner points in G

Neumann + regular grid:



$$\left.\frac{\partial u}{\partial n}\right|_P \approx \frac{u_E - u_W}{2h} = 0$$

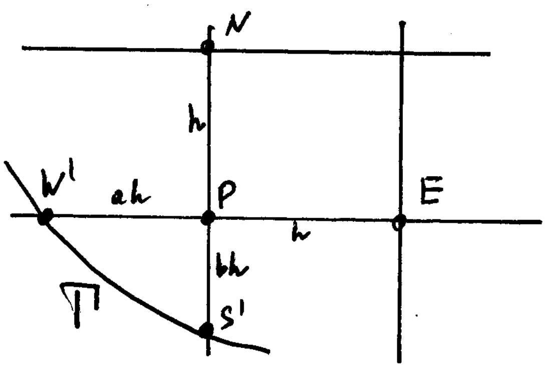
$$\Rightarrow \quad u_E = u_W$$

Poisson-EQ:

$$4u_P - u_N - 2u_W - u_S + h^2 f_P = 0$$

$$\begin{array}{ccc} & -1 & \\ -2 & 4 & \\ & -1 & \end{array} \odot u + h^2 f_P \; ; \quad \left.\frac{\partial u}{\partial n}\right|_P = \left.\frac{\partial u}{\partial x}\right|_P = 0$$

Complex boundary:



$0 < a, b < 1$

Goal:

evaluate u_{xx}, u_{yy} in P

Taylor:

$$u(x+h,y) = u(x,y) + h\,u_x(x,y) + \frac{1}{2}h^2 u_{xx}(x,y) + \frac{1}{6}h^3 u_{xxx}(x,y) + \dots$$

$$u(x-ah,y) = u(x,y) - ah\,u_x(x,y) + \frac{1}{2}a^2h^2 u_{xx}(x,y) - \frac{1}{6}a^3h^3 u_{xxx}(x,y) + \dots$$

$$u(x,y) = u(x,y)$$

Linear combination:

$$c_1\,u(x+h,y) + c_2\,u(x-ah,y) + c_3\,u(x,y)$$

$$= (c_1+c_2+c_3)\,u(x,y) + (c_1 - a c_2)\,h\,u_x(x,y)$$

$$+ (c_1 + a^2 c_2)\,\frac{1}{2}h^2 u_{xx}(x,y) + \dots$$

$$\overset{!}{=} u_{xx}(x,y)$$

$$\Longrightarrow \quad c_1 = \frac{2}{h^2(1+a)}\,; \quad c_2 = \frac{2}{h^2 a(1+a)}\,; \quad c_3 = -\frac{2}{h^2 a}$$

Thus:

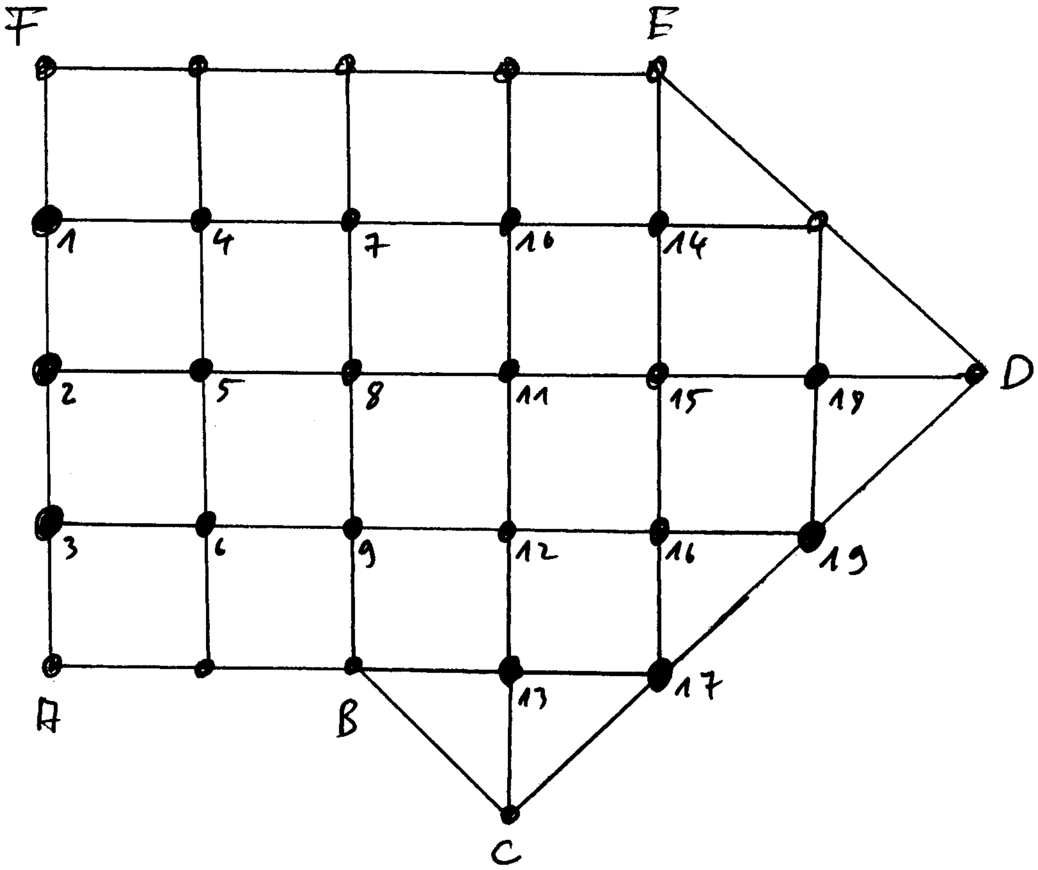
$$u_{xx}(P) \approx \frac{2}{h^2}\left\{\frac{u_E}{1+a} + \frac{u_{W'}}{a(1+a)} - \frac{u_P}{a}\right\}$$

$$u_{yy}(P) \approx \frac{2}{h^2}\left\{\frac{u_N}{1+b} + \frac{u_{S'}}{b(1+b)} - \frac{u_P}{b}\right\}$$

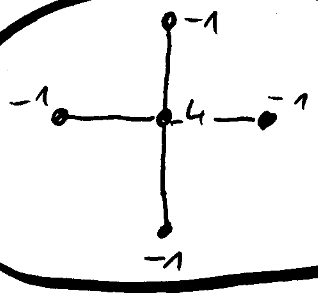
Poisson-Eq.

$$2\left(\frac{1}{a} + \frac{1}{b}\right)u_P - \frac{2}{1+b}u_N - \frac{2}{a(1+a)}u_{W'} - \frac{2}{b(1+b)}u_{S'} - \frac{2}{1+a}u_E + h^2 f_P = 0$$

Step 4: Labels ($\Rightarrow$ linear set of equations ....)

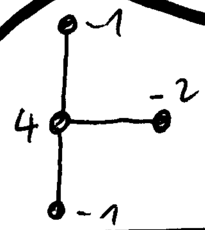


Rules: - in G:



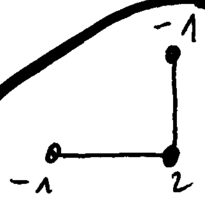
$\odot\, u + h^2 f = 0$

- on $\overline{FA}$:



$\odot\, u + \frac{1}{2} h^2 f = 0$

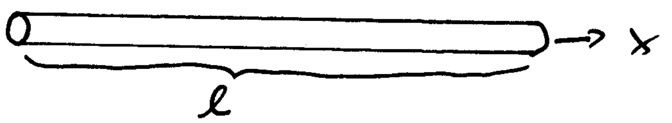
- on $\overline{CD}$ (where $u_S = u_W$ and $u_E = u_N$)



$\odot\, u + \frac{1}{2} h^2 f = 0$

etc.

Laplace equation in 1D (stability)

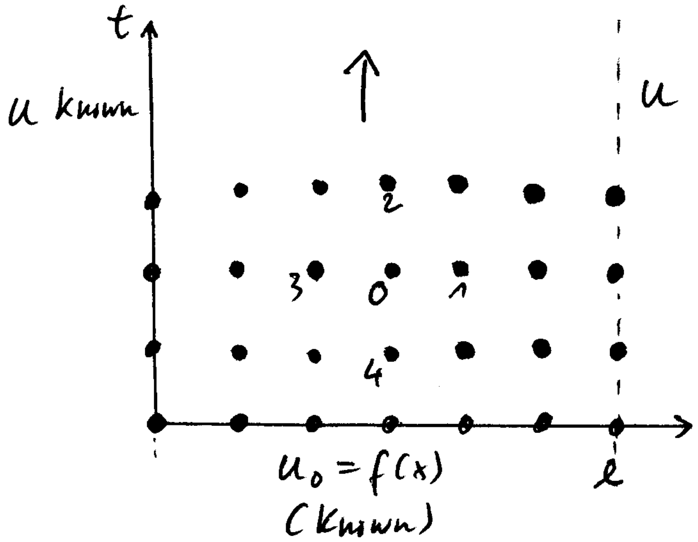


temperature $u(x,t)$

$u_0 = f(x)$

(boundary cond., initial cond.)

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t} \quad \text{(parabolic PDE)}$$



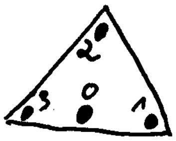
Forward difference:

$$\frac{u_1 - 2u_0 + u_3}{h^2} = \frac{u_2 - u_0}{k}$$

Backward difference:

$$\frac{u_1 - 2u_0 + u_3}{h^2} = \frac{u_0 - u_4}{k}$$

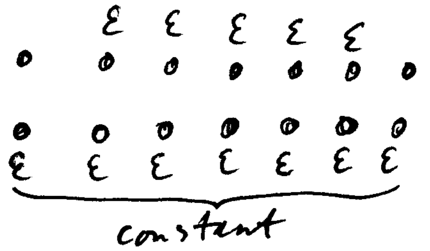
Forward difference:



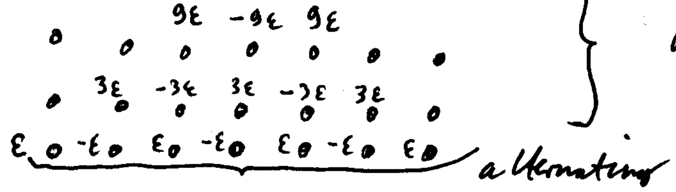
- no simultaneous equations have to be solved!
- boundary values enter only in the calculation of extreme interior points.

$$\boxed{\frac{k}{h^2} = 1}$$

$$u_2 = u_1 - u_0 + u_3$$



$\varepsilon_2 = \varepsilon_1 - \varepsilon_0 + \varepsilon_3$
$= \varepsilon$
(error remains unchanged)



algorithm becomes unstable

$\boxed{\frac{K}{h^2} = \frac{1}{4}}$

$$\boxed{u_2 = \frac{1}{4} u_1 + \frac{1}{2} u_0 + \frac{1}{4} u_3}$$

constant pattern: $\varepsilon_2 = \varepsilon$ (as before)

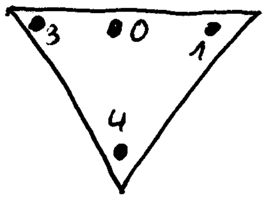
alternating pattern: $\varepsilon_2 = \frac{\varepsilon}{4} - \frac{\varepsilon}{2} + \frac{\varepsilon}{4} = 0$ (noise suppressor)

$\boxed{\frac{K}{h^2} = \frac{1}{2}}$

$$\boxed{u_2 = \frac{1}{2}(u_1 + u_3)}$$ simple, but.....

$$\left.\begin{array}{cccccc} o & \times & o & \times & o & \times \\ \times & o & \times & o & \times & o \\ o & \times & o & \times & o & \times \end{array}\right\}$$ checker board (unphysical!!!)

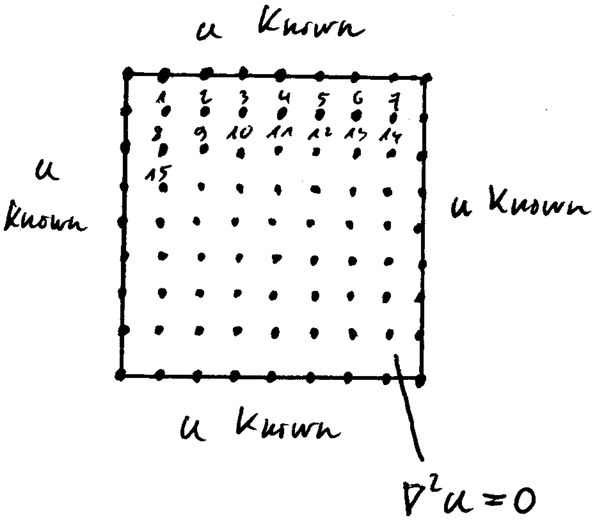
Backward difference:



$$\boxed{\frac{K}{h^2}(u_1 - 2u_0 + u_3) - u_0 = -u_4}$$

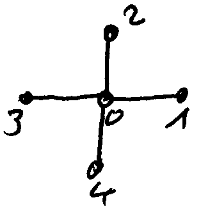
- leads to a system of equations
- boundary conditions enter that system of equations
- algorithm is stable for all $\frac{K}{h^2}$!
- still subject to truncation errors (grid....)

Laplace's equation in 2D



$$\boxed{\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \nabla^2 u = 0}$$

Discretized PDE:



$$\left.\frac{\partial^2 u}{\partial x^2}\right|_0 \approx \frac{u_1 - 2u_0 + u_3}{h^2}$$

$$\left.\frac{\partial^2 u}{\partial y^2}\right|_0 \approx \frac{u_2 - 2u_0 + u_4}{h^2}$$

$$\Rightarrow \quad u_1 + u_2 + u_3 + u_4 - 4u_0 = 0$$

Similar procedure in 3D...

Boundaries:

$$u_2 - 4u_3 + u_4 + u_{10} = -u_6$$

etc.

System of linear algebraic equations:

$$\begin{array}{c|c}
 & \begin{array}{ccccccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} &
\underbrace{\left[\begin{array}{ccccccccccccccccc}
-4 & 1 & & & & & & 1 & & & & & & & & & \\
1 & -4 & 1 & & & & & & 1 & & & & & & & & \\
 & 1 & -4 & 1 & & & & & & 1 & & & & & & & \\
 & & 1 & -4 & 1 & & & & & & 1 & & & & & & \\
 & & & 1 & -4 & 1 & & & & & & 1 & & & & & \\
 & & & & 1 & -4 & 1 & & & & & & 1 & & & & \\
 & & & & & 1 & -4 & 0 & & & & & & 1 & & & \\
1 & & & & & & 0 & -4 & 1 & & & & & & 1 & & \\
 & 1 & & & & & & 1 & -4 & 1 & & & & & & 1 & \\
 & & 1 & & & & & & 1 & -4 & 1 & & & & & & 1
\end{array}\right]}_{\text{sparse matrix}}
\end{array}
\left[\begin{array}{c} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \\ u_{10} \end{array}\right]
=
\left[\begin{array}{c} -b_0 - b_3 \\ -b_2 \\ -b_3 \\ -b_4 \\ -b_5 \\ -b_6 \\ -b_7 - b_8 \\ b_8 \\ 0 \\ 0 \end{array}\right]$$

boundary

Corner

or: $\boxed{\underline{\underline{A}}\,\underline{u} = \underline{b}}$