Laplace Gleichung, Jacobi & SOR

15 April 2019 12:20

Partielle Differenzialgleichungen

$$\frac{\partial^2 u}{\partial t^2} = v^2 \frac{\partial^2 u}{\partial x^2}$$ 1-D Wellengleichung

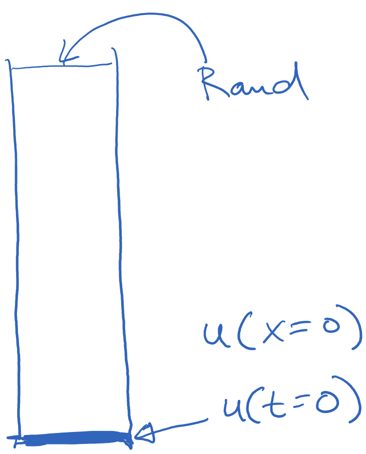
v - Wellengeschwindigkeit

(hyperbolische)

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2}$$ Diffusionsgleichung

(parabolische)

Raum & Zeit < Anfangsbedingungen, Randbedingungen



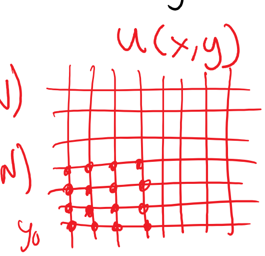
$$\underbrace{\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}}_{\text{Randbedingungen}} = \rho(x,y)$$ Poisson Gleichung.

$\rho = 0 \implies \nabla^2 u = 0$ Laplace Gleichung.

Elektrostatik im Vacuum

$$\nabla^2 = \left(\frac{\partial}{\partial x^2} + \frac{\partial}{\partial y^2}\right)$$

Wir führen ein Gitter ein:



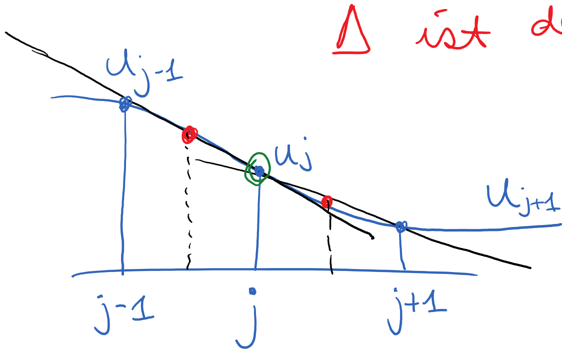
$x_j = x_0 + j\Delta \qquad j = 0, 1, 2, \ldots, J \; (N)$

$y_l = y_0 + l\Delta \qquad l = 0, 1, 2, \ldots, L \; (N)$

Δ ... Gitterabstand

Δ ist der Gitterabstand

u_{jl}



$$\left.\frac{\partial u}{\partial x}\right|_{j-1/2} \approx \frac{u_j - u_{j-1}}{x_j - x_{j-1}} \quad \Delta$$

$$\left.\frac{\partial u}{\partial x}\right|_{j+1/2} \approx \frac{u_{j+1} - u_j}{x_{j+1} - x_j} \quad \Delta$$

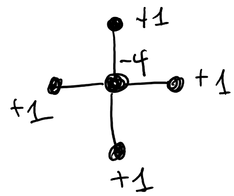
$$\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right) \approx \frac{\left.\frac{\partial u}{\partial x}\right|_{j+\frac{1}{2}} - \left.\frac{\partial u}{\partial x}\right|_{j-1/2}}{\Delta}$$

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{j} \approx \frac{(u_{j+1} - u_j) - (u_j - u_{j-1})}{\Delta^2}$$

$$\approx \frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta^2}$$

$$\left.\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right|_{j,l} \approx \frac{u_{j+1,l} + u_{j-1,l} + u_{j,l+1} + u_{j,l-1} - 4u_{j,l}}{\Delta^2}$$

$\approx$



$\mathcal{O}(\Delta^3)$ Fehler

$$\nabla^2 u = 0 \Rightarrow \frac{u_{j+1,l} + u_{j-1,l} + u_{j,l+1} + u_{j,l-1} - 4u_{j,l}}{\Delta^2} = 0$$

$$\boxed{u_{jl}^{(n+1)} = \frac{1}{4}\left(u_{j+1,l}^{(n)} + u_{j-1,l}^{(n)} + u_{j,l+1}^{(n)} + u_{j,l-1}^{(n)}\right)}$$

iteration $\mathcal{O}(N^2)$ Δ

4 Flops

Jacobi Methode

$N_{iter} \sim \frac{1}{2} p J^2$ } Wenn der Fehler um einen Faktor 10^{-p} reduziert

$N_{iter} \sim \frac{1}{2} p J^2$ wenn der Fehler um einen Faktor 10^{-p} reduziert werden soll.

$$Ops = \frac{3}{2} N^2 \cdot N^2 \cdot 4 = 6 N^4 \text{ Flops}$$

Successive / Stufenweise Over-Relaxation (SOR)

$$u_{je}^{(n+1)} = u_{je}^{(n)} + \underbrace{\frac{\omega}{4}\left(u_{j+1,e}^{(n)} + u_{j-1,e}^{(n)} + u_{j,e+1}^{(n)} + u_{j,e+1}^{(n)} - 4u_{je}^{(n)}\right)}_{\text{"Fehler"}}$$

R_{je}

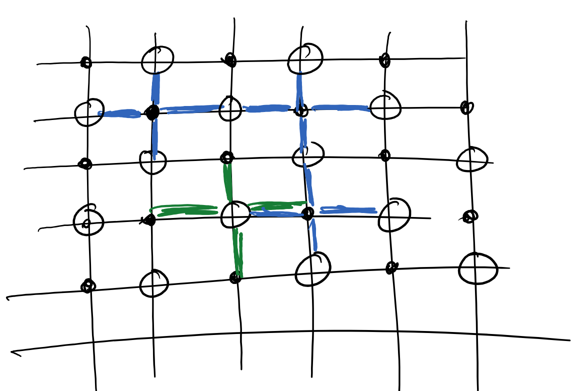
$\omega = 1 \Rightarrow$ Jacobi

$\omega > 1 \Rightarrow$ SOR

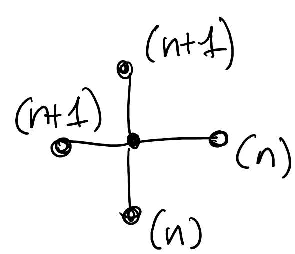
$0 < \omega < 2$ Stabil

Für optimales $\omega \Rightarrow N_{iter} \sim \frac{1}{3} p J$!

$$\omega \simeq \frac{2}{1 + \pi/J}$$ sollte gut funktionieren



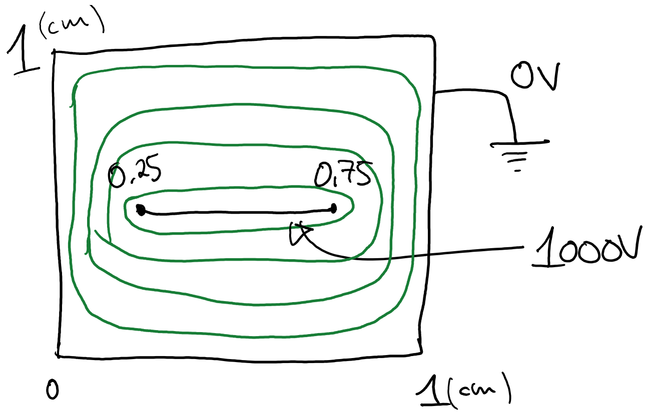
"In Place" Algorithm



Gauss Seidel Methode

$N_{iter} \sim \frac{1}{2} N_{Jacobi}$

NO IF then else



NO IF-then-else !

$R_{jl} = 0$ für Rand

$R_{jl} = \frac{\omega}{4}$ sonnst

Niter

Schwarz

```
for (l=0; l<=L; ++l) {
    for (j = l&1; j<=J; j+=2) {
        u_jl += R_jl [ ]
    }
}
```

Weiss

```
for (l=0; l<=L; ++l) {
    for (j = (l+1)&1; j<=J; j+=2) {
        u_jl += R_jl [ ]
    }
}
```