Spin 1 - Partielle Differenzialgleichungen - Elektrostatik, SOR

Monday, April 25, 2016 12:53 PM

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

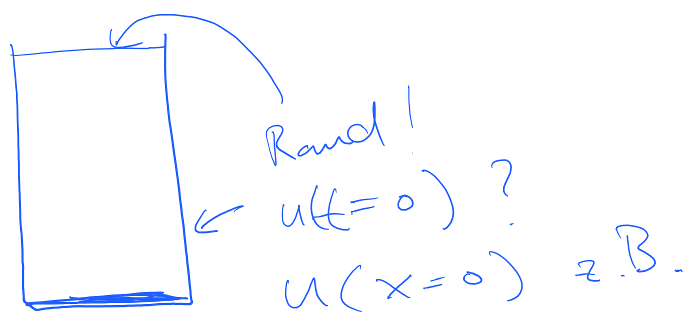
v - Wellengeschwindigkeit

(hyperbolischen)

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

(parabolische)

Raum & Zeit $<$ Anfangs/bedingungen, Randbedingungen



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

Randbedingungen

$\nabla^2 u = 0$ $\quad \rho = 0$ Laplace Gleichung

Elektrostatik im Vakuum

Wir führen ein Gitter ein:

$$x_j = x_0 + j\Delta \qquad j = 0, 1, 2, \dots, J$$

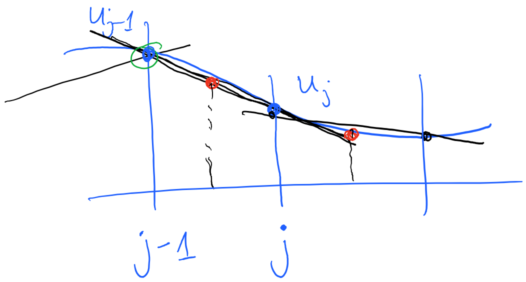
$$y_\ell = y_0 + \ell\Delta \qquad \ell = 0, 1, 2, \dots, L$$

Δ ist der Gitterabstand

$u_{j-1},$

$$\partial u \Big| \sim \frac{u_j - u_{j-1}}{}$$

Δ ist der ...



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

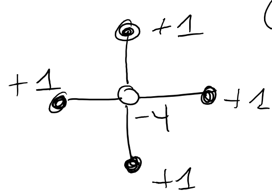
$$\left.\frac{\partial u}{\partial x}\right|_{j+\frac{1}{2}} \approx \frac{u_{j+1} - u_j}{\Delta}$$

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

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

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

$O(\Delta^2)$



Iteration

$$\nabla^2 u = 0 \implies u_{jl}^{(n+1)} = \frac{1}{4}\left(u_{j+1,l}^{(n)} + u_{j-1,l}^{(n)} + u_{j,l+1}^{(n)} + u_{j,l-1}^{(n)}\right)$$

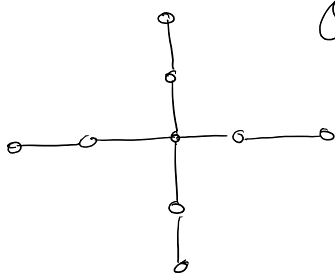
Jacobi Methode

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

Nehmen wir an es ist ein Quadrat: $L = J$

Anzahl Gitterpunkte $= J^2 \times 4$ flops

$\times N_{iter} = 2pJ^4$ Flops



$O(\Delta^4)$

Geht es besser? Ja deutlich.

Successive / Stufenweise Over-Relaxation
(SOR)

$$u^{neu}_{j,l} = u^{ALT}_{jl} + \frac{\omega}{4}\left[u_{j+1,l} + u_{j-1,l} + u_{j,l+1} + u_{j,l-1} - 4u_{jl}\right]^{ALT} \sim \text{"Fehler"}$$

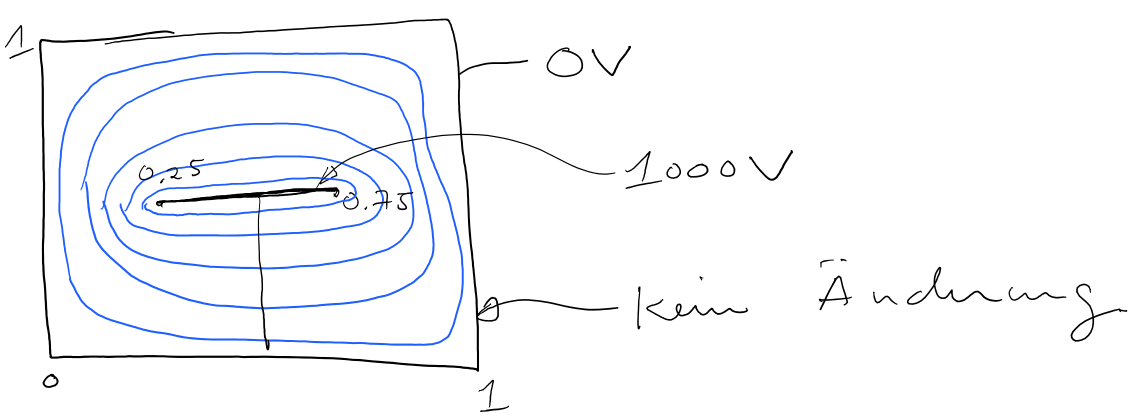
$\omega = 1 \Rightarrow$ Jacobi

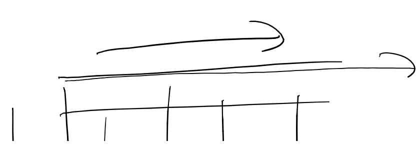
$\omega > 1 \Rightarrow$ SOR

$0 < \omega < 2$ Stabil

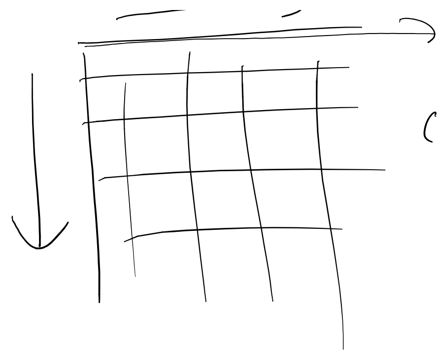
Für optimales $\omega \Rightarrow N_{iter} \sim \frac{1}{3} pJ$!

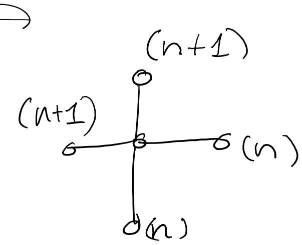
$$\omega \simeq \frac{2}{1 + \pi/J}$$ sollte gut funktionieren.





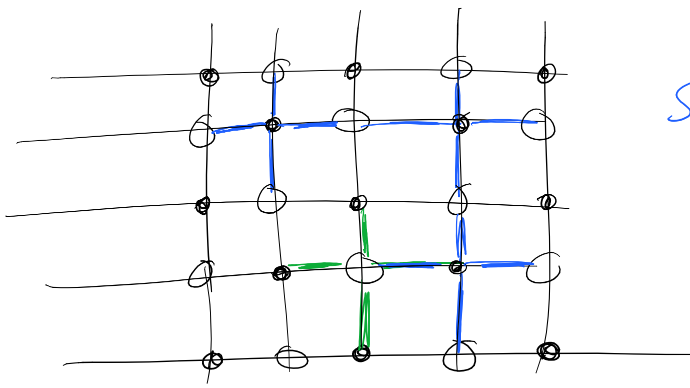
$(n+1)$ Gauss-





Gauss-Seidel

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

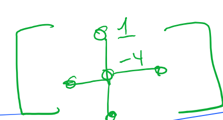


Ich kann alle Schwarze Punkte gleichzeitig Rechnen

↳ parallele Rechnen.

Ich brauche somit nur 1 2-D Array für u.

KEIN if-then-else!

$u_{j\ell} \mathrel{+}= R_{j\ell}$ 

$R_{j\ell} = 0$ für Rand

$= \frac{\omega}{4}$ sonst!

Schwarze Punkte

```
for (l=0; l≤L; ++l) {
    for (j= l&1; j≤J; j+=2) {
    }
}
```

Weisse Punkte

```
for (l=0; l≤L; ++l) {
    for (j=(l+1)&1; j≤J; j+=2) {
    }
```

Punkte