SINS 8: Laplace Equation, Jacobi and SOR Methods

04 November 2019 12:59

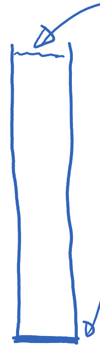
Partial Differential Equations

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

(hyperbolic)

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

(parabolic)



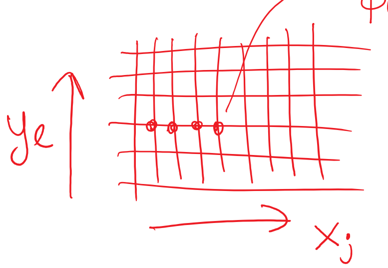
$$\underbrace{\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}}_{\text{Only Boundary Conditions}} = \rho(x,y)$$ Poisson Equation

(elliptic P.D.E.s)

$\rho = 0$: $$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad (\text{or}) \quad \nabla^2 u = 0$$ Laplace Equation

Electrostatics in Vacuum $\phi \equiv u$

Discretize the operator ∇^2



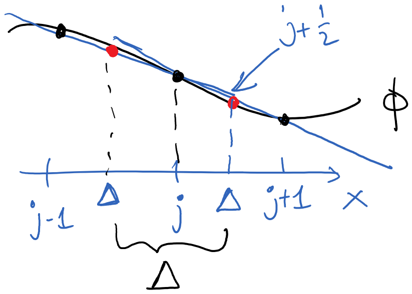
$x_j = x_0 + j\Delta$

$y_\ell = y_0 + \ell\Delta$

$j = 0, 1, 2, \ldots J\ (N)$

Δ is the grid spacing

$\ell = 0, 1, 2, \ldots, L \ (N)$



$$\left.\frac{\partial \phi}{\partial x}\right|_{\boxed{j+\frac{1}{2}}} \simeq \frac{\phi_{j+1} - \phi_j}{\Delta}$$

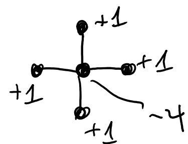
$$\left.\frac{\partial \phi}{\partial x}\right|_{j-\frac{1}{2}} \simeq \frac{\phi_j - \phi_{j-1}}{\Delta}$$

$$\left.\frac{\partial^2 \phi}{\partial x^2}\right|_{j} \simeq \frac{\left.\frac{\partial \phi}{\partial x}\right|_{j+\frac{1}{2}} - \left.\frac{\partial \phi}{\partial x}\right|_{j-\frac{1}{2}}}{\Delta}$$

$$\boxed{\left.\frac{\partial^2 \phi}{\partial x^2}\right|_{j} \simeq \frac{\phi_{j+1} - 2\phi_j + \phi_{j-1}}{\Delta^2}}$$

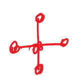
$$\left.\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2}\right|_{j,\ell} \simeq \frac{\phi_{j+1,\ell} + \phi_{j-1,\ell} + \phi_{j,\ell+1} + \phi_{j,\ell-1} - 4\phi_{j,\ell}}{\Delta^2}$$

$\simeq$



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

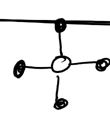
$\nabla^2 \phi = 0$



$= 0$

$$\frac{\phi_{j+1,\ell} + \phi_{j-1,\ell} + \phi_{j,\ell+1} + \phi_{j,\ell-1} - 4\phi_{j,\ell}}{\Delta^2} = 0$$

$$\boxed{\phi^{(n+1)}_{j,\ell} = \frac{1}{4}\left(\phi^{(n)}_{j+1,\ell} + \phi^{(n)}_{j-1,\ell} + \phi^{(n)}_{j,\ell+1} + \phi^{(n)}_{j,\ell-1}\right)}$$



Average

Jacobi

iterate

4 Flops

$J \cdot L$ times

$\mathcal{O}(N^2)$

to go over the grid once!

$N_{iter} \sim \frac{1}{2} p J^2$ $\boxed{\text{on a } J \times J \text{ grid}}$

to reduce the error by a factor of 10^{-p}

Number of ops to reduce the error

by a factor 10^3 (10^{-3})

$$OPS = \frac{3}{2} J^2 \cdot 4J^2 = 6J^4 \text{ ops}$$

$$\mathcal{O}(N^4)$$

N^3: Successive Over-Relaxation (SOR)

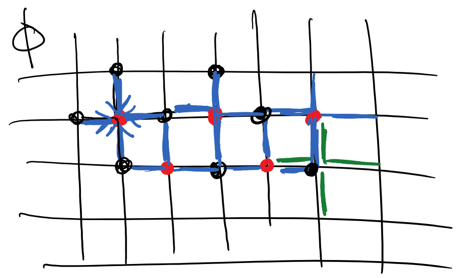
$$\phi_{je}^{(n+1)} = \phi_{je}^{(n)} + \underbrace{\underbrace{\frac{w}{4}}_{R_{je}} \left(\phi_{j+1,e}^{(n)} + \phi_{j-1,e}^{(n)} + \phi_{j,e+1}^{(n)} + \phi_{j,e-1}^{(n)} - 4\phi_{je}^{(n)} \right)}_{\text{Correction Term}}$$

$w = 1 \Rightarrow$ Jacobi

$w > 1 \Rightarrow$ SOR

If w is optimal $\Rightarrow N_{iter} \sim \frac{1}{3} pJ$!

$w \simeq \dfrac{2}{1 + \pi/J}$ should work well in most cases.

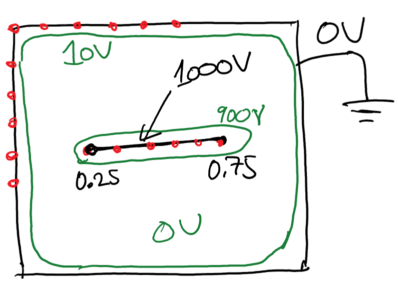


Checker board update

First do all black points (all independent)

Second do all red points

"In Place Algorithm"



No IF-THEN-ELSE

$R_{je} = 0$ at a Boundary

$R_{je} = \dfrac{w}{4}$ otherwise

$$\phi_{je}^{(n+1)} = \phi_{je}^{(n)} + R_{je} \cdot (\ )^{(n)}$$

```
Black { for (l=0 ...   ) {
          for (j = l&1 ...   ) {
               φ_je += R_je [ ]
```

iterate

bla { for (j = ...) {

Φ_{jl} += R_{jl}[ ]

} }

iterate

red { for (l = 0 ...

for (j = (l+1)41 ; ...

Φ_{jl} += R_{jl}[ ]

}

}

```
from scipy import ndimage
W = np.array([[0,1,0],[1,-4,1],[0,1,0]])
ndimage.convolve(U, W, output=C, mode="constant", cval=0)
np.multiply(R, C, out=M)
U = U + M
```

P : Checker board pattern ← How?

Boolean Array < True / False

```
U[P] = U[P] + M[P]

U[~P] = U[~P] + M[~P]
```

```
plt.countour(U)
```