SM315 Lecture Notes
Approximating Laplace's Equation with FD Equations

1. Space Time Discretization of 2D Wave Equation in One-Dimension with a forcing function

a. Let: $u_{j,l}$ by the value of the function $u(x,y)$ at j^{th} x grid point and l^{th} y grid point.

b. Recall Laplace's for a rectangle:

$$\begin{cases} \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = 0 \\ BC: \quad u(0,y) = f_1(y) \quad u(L,y) = f_2(y) \\ \qquad\;\; u(x,0) = g_1(x) \quad u(x,H) = g_2(x) \end{cases}$$

c. Descritization becomes:

$$\left(\frac{u_{j-1,l} - 2u_{j,l} + u_{j+1,l}}{\Delta x^2} + \frac{u_{j,l-1} - 2u_{j,l} + u_{j,l+1}}{\Delta y^2} \right) = 0$$

d. Assume that $\Delta x^2 = \Delta y^2 \rightarrow \dfrac{u_{j-1,l} + u_{j,l-1} - 4u_{j,l} + u_{j+1,l} + u_{j,l+1}}{\Delta x^2} = 0$

e. Therefore: $u_{j,l} = \dfrac{u_{j-1,l} + u_{j,l-1} + u_{j+1,l} + u_{j,l+1}}{4}$.

f. Why now apply the an iterative process to get an equilibrium answer, i.e. let $u^0_{j,l}$ be an initial guess for the temperature at the interior points of the rectangle, then

$$u^1_{j,l} = \frac{u^0_{j-1,l} + u^0_{j,l-1} + u^0_{j+1,l} + u^0_{j,l+1}}{4}$$

- Note: Points on boundaries are determined by boundary conditions.

g. Continue this iteration for all of the interior points until the difference between $u^{m+1}_{j,l}$ and $u^m_{j,l}$ is an acceptable tolerance δ (you chose delta), i.e:

$$u^{m+1}_{j,l} = \frac{u^m_{j-1,l} + u^m_{j,l-1} + u^m_{j+1,l} + u^m_{j,l+1}}{4} \text{ until } \left| u^{m+1}_{j,l} - u^m_{j,l} \right| \le \delta$$

h. Above method is called ***Jacobian Iteration***, if old values of *u* are used in each iteration stop.

i. Above method is called ***Guass-Seidel Iteration***, if calculated values immediately replace the old values of *u*. The new values are use in subsequent iterations.

j. One is not necessarily better that the other.

2. Example – Recall Previous Project

$$\begin{cases} \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = 0 \\ BC: \quad u(0,y) = 2y(y-1) \qquad u(1,y) = 0 \\ \qquad\quad u(x,0) = 0 \qquad\qquad u(x,1) = 2x(1-x) \end{cases}$$

a. Using MATLAB (***go1***)to compare Jacobian vs. Gauss-Seidel. In demo we let $\Delta x = \Delta y = .01$ and tolerance $\delta = .000001$.

3. S-O-R Iteration: A Scheme to Reduce the Number Iterations to Reach Tolerance.

a. Iterative Scheme: $u_{j,l}^{m+1} = u_{j,l}^{m} + \omega\left(u_{j-1,l}^{m} + u_{j,l-1}^{m} + u_{j+1,l}^{m} + u_{j,l+1}^{m} - 4u_{j,l}^{m}\right)$

b. Here ω is called the relaxation factor.

- If improperly chosen (i.e. $\omega \geq .5$) scheme is unstable.
- If properly chosen, number of iterations is reduced.
- Note: if $\omega = .25$ the above scheme reduces to the Jacobi or Gauss-Seidel scheme.

c. It turns out that the best choice for ω in a square domian is $\omega = .\dfrac{1 - pi/N_x}{2}$ where N_x is the number of grid points in the x-direction (hard analysis here … don't worry about it).

- Note N_x will also be the number of grid points in the y-direction if the domain is square and $\Delta x = \Delta y$.

d. MATLAB example compares Gauss-Sidel to S-O-R iterations (***go2***)