Lecture 2:

Total Recall

Math, Part 2

Ordinary diff. equations

- First order ODE, one boundary/initial condition:

$$\frac{dy}{dx} = f(x,y), \qquad y(x_0) = y_0$$

- Second order ODE

$$\frac{d^2y}{dx^2} + f(x,y)\frac{dy}{dx} = g(x,y)$$

ODEs contn'd

- 2nd order ODE can be replaced with a system of 1st order ODEs:

$$\frac{du}{dx} = v \cdot g(x)$$

$$\frac{dv}{dx} = h(x, u)$$

- Typical situation in RT involves two-point boundary conditions

$$y(x_0) = y_0, \qquad y(x_1) = y_1$$

- … or initial condition

$$y(x_0) = y_0, \qquad \left.\frac{dy}{dx}\right|_{x_0} = y_0'$$

Runge-Kutta

- For the 1st order ODE the Euler method gives:

$$y_{i+1} = y_i + (x_{i+1} - x_i) f(x_i, y_i)$$

this also happens to be first order RK scheme

- 4th order RK:

$$k_1 = h \cdot f(x_i, y_i)$$
$$k_2 = h \cdot f(x_i + 0.5h, y_i + 0.5k_1)$$
$$k_3 = h \cdot f(x_i + 0.5h, y_i + 0.5k_2)$$
$$k_4 = h \cdot f(x_i + h, y_i + k_3)$$
$$y_{i+1} = y_i + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + O(h^5)$$

- Note that RK is directly applicable to a system of ODE and therefore to any order ODE with *initial* conditions

Finite-differences

- For the 1st order ODE:

$$\frac{dy}{dx} = f(x, y)$$

$$y_{k+1} - y_k = (x_{k+1} - x_k) f(\frac{x_{k+1} + x_k}{2}, \frac{y_{k+1} + y_k}{2})$$

- For the 2nd order ODE:

$$\frac{d^2 y}{dx^2} = f(x, y)$$

$$\frac{\left(\frac{y_{k+1} - y_k}{x_{k+1} - x_k}\right) - \left(\frac{y_k - y_{k-1}}{x_k - x_{k-1}}\right)}{\left(\frac{x_{k+1} + x_k}{2}\right) - \left(\frac{x_k + x_{k-1}}{2}\right)} = f(x_k, y_k)$$

2nd order scheme

$$\frac{d^2y}{dx^2} + f(x)\frac{dy}{dx} = g(x) + \alpha y$$

This particular form of the 2nd order ODE can be approximated by the following simple FD scheme:

$$A_k y_{k+1} + B_k y_k + C_k y_{k-1} = D_k \quad \text{for} \quad k = 2, N-1$$

For $k = 1$ and $k = N$ equations are given by boundary conditions.

Home work 2a:

Derive the value for coefficients in the numerical scheme above

- This numerical scheme results in 3-diagonal matrix:

$$\begin{pmatrix} b_1 & c_1 & & \\ a_2 & b_2 & c_2 & \\ & & \ddots & c_{N-1} \\ & & a_N & b_N \end{pmatrix} \times \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_N \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \cdots \\ f_N \end{pmatrix}$$

- Solving a system of ODEs leads to a block-diagonal SLE!

RK versus Finite-diff.

- RK typically has homogeneous convergence, allows higher orders, has good round-off stability (because errors are controlled on every step).
- High accuracy my be very expensive, specially in multidimensions (fast increase of the number of steps).
- FD converges (if it does) much faster. More complex schemes (2nd order and higher) gain stability and convergence speed.
- Expensive and difficult to check maximum error.
- ✓ Cook-book: to study solution locally when you have time use RK. When high accuracy must be combined with high performance use FD.

Partial Differential Equations

- PDEs come in three flavors: *hyperbolic*, *parabolic*, and *elliptic*
- A typical example of a hyperbolic equation is a *wave* equation:

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

where v is the velocity of wave propagation

PDEs examples

An example of parabolic equation is the *diffusion* equation:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(D \frac{\partial u}{\partial x} \right)$$

where $D(>0)$ is the diffusion coefficient

PDEs examples

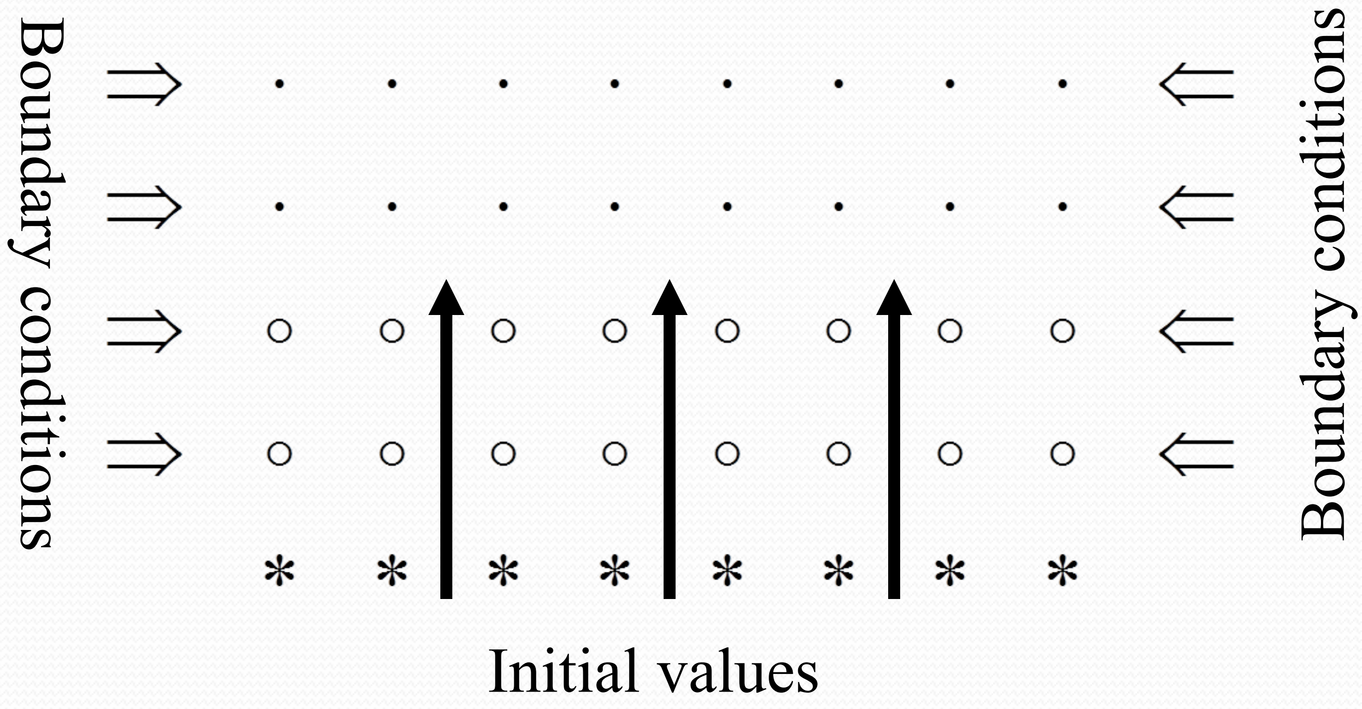
Poisson equation is an example of elliptic type:

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

where ρ is the source function (density of charges). If ρ is zero for the whole domain we get a special case of Laplace equation

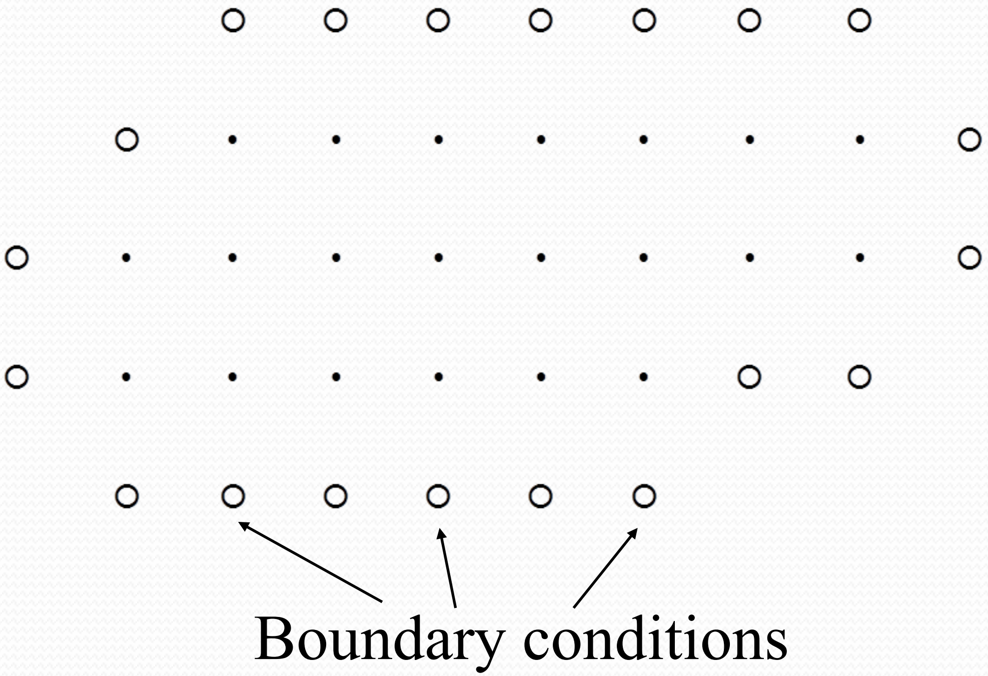
Numerical schemes

Numerically, the first two types are *initial value* problems (Cauchy problems):



Numerical schemes

An elliptic equation is a boundary condition problem:



Numerical scheme for Poisson equation

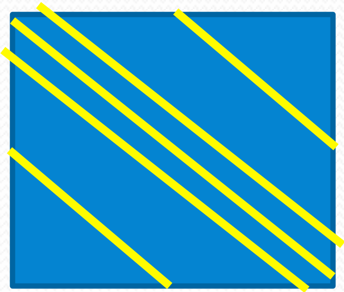
Assuming an equispaced rectangular grid in 2D with stepsize Δ:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta^2} = \rho_{i,j}$$

In matrix form the scheme looks like this:

$$u_{k+N_x+1} + u_{k-(N_x+1)} + u_{k+1} + u_{k-1} - 4u_k = \Delta^2 \rho_k$$

which is a 5-diagonal SLE



Boundaries and generalization

- The scheme on the previous slide only holds inside the domain. The points for $i = 1, N_x$ and $j = 1, N_y$ are described by the boundary conditions leading to a system of linear equation $\mathbf{A}u = b$.
- In our case, elements on each diagonal of **A** are constant. In general case, matrix elements along diagonals change.
- In 3D more diagonals are present.

Convergence and stability

- *Approximation* – the accuracy of approximation of the analytical equation(s) by numerical scheme.
- *Convergence* - the property of the numerical scheme to get closer to the exact solution when the grid becomes denser in some regular way.
- *Stability* - the stability of numerical scheme characterizes the way errors (e.g. finite difference approximation of derivatives) are accumulated during the integration. Stability of the computer implementation of numerical scheme also includes round-off error accumulation.

Computational errors

- Floating point numbers are stored in 32- or 64-bit long words (IEEE):

`0 11010000 11011011110111001000011`

sign exponent mantissa

`1 10000001101 0000101110100100010001011111011100...`

sign exponent mantissa

- Multiplications/divisions do not loose much precision but subtraction/addition is a danger

HOME WORK 2b: convergence

Numerical differentiating:

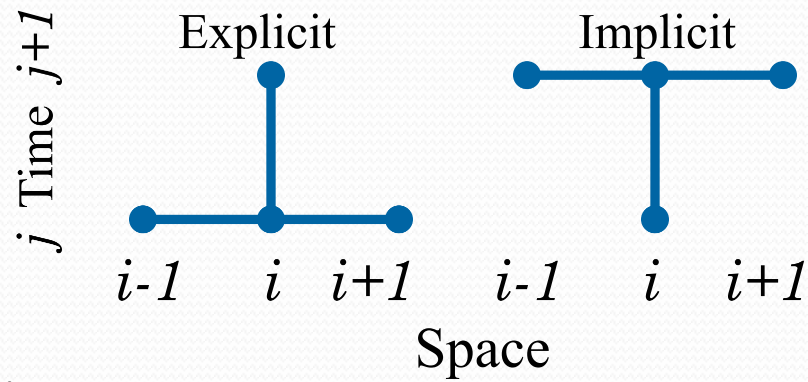
$$\frac{dy}{dx} \approx \frac{y_{i+1} - y_i}{x_{i+1} - x_i}; \qquad \lim_{x_{i+1} \to x_i} \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = ?$$

Is there an optimal Δx?

Diffusion equation

Initial value problem:

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



can be approximated as:

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} = D\left(\alpha\frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{\Delta x^2} + \right.$$

$$\left. + \beta\frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{\Delta x^2}\right)$$

Stability analysis of numerical schemes

- We represent the approximation errors our solution u_l^m with Fourier components:

$$\delta u_l^m \equiv u_l^m - \bar{u}_l^m = \zeta_m e^{am\Delta t} \sum_k v_k e^{ikl\Delta x}$$

- Substituting individual components to the finite difference scheme we find the condition that results in $\left|\delta u_l^{m+1}/\delta u_l^m\right| > 1$ corresponding to the unlimited error growth. For the diffusion equation this condition is: $\frac{D\Delta t}{\Delta x^2} < C$ Courant condition

Explicit or implicit?

- New sense of stability: *how dramatic will be the solution after many time steps if we change the initial conditions a little bit?*
- Explicit schemes ($\alpha=1$, $\beta=0$) tend to have better convergence
- Purely implicit schemes ($\alpha=0$, $\beta=1$) tend to be more stable
- Combining the two ($\alpha>0$, $\beta>0$) helps to get the optimal scheme

Integration: Gauss quadratures

- For any "reasonable" function *f(x)* the integral with kernel *K(x)* can be approximated as a sum of function values in nodes x_i multiplied by weights.

$$\int_b^a K(x) f(x) dx \approx \sum_{i=1}^{N} \omega_i f(x_i)$$

- If f(x) can be represented by a polynomial of order $2 \cdot \mathrm{N}$, the quadrature formula is exact.

Non-linear equations

For system of non-linear equations we often use Newton-Raphson scheme:

$$F_i(x_1, x_2, \ldots, x_N) = 0, \quad i = 1, \ldots N$$

$$0 = F_i(\vec{x} + \delta\vec{x}) = F_i(\vec{x}) + \sum_{j=1}^{N} \frac{\partial F_i(\vec{x})}{\partial x_j} \delta x_j$$

$$\left(\frac{\partial F_i(\vec{x})}{\partial x_j} \right) \cdot \left(\delta\vec{x} \right) = -\left(\vec{F}(\vec{x}) \right); \; \vec{x}_{\text{new}} = \vec{x}_{\text{old}} + \delta\vec{x}$$

Home work 2:

c. Write a simple program for 4x4 matrix inversion with Gauss-Jordan elimination and partial pivoting. Test round-off stability by doing many inversions of a random 4 by 4 matrix.

d. Propose the scheme (flow chart) for solving SLE with 4x4 block-diagonal matrix with 1 row overlap between blocks

e. Write (or use NR routine) 4th order RK and FD scheme for the equation:

$$\frac{d^2y}{dx^2} = y + e^x; \quad y(0) = 1; \quad y(15) = 10^3$$