Classification of EM Problems, Finite Difference Discretization Scheme and Solution of Sparse Matrix Equation Ax=b

Dragica Vasileska, Professor
Arizona State University

Classification of EM Problems

Classifying EM problems will help us later to answer the question of what method is best for solving a given problem. Continuum problems are categorized differently depending on the particular item of interest, which could be one of these:

(1) the solution region of the problem,

(2) the nature of the equation describing the problem, or

(3) the associated boundary conditions.

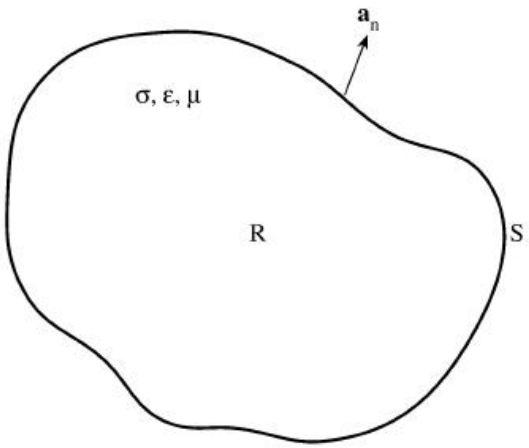
(In fact, the above three items define a problem uniquely.)

It will soon become evident that these classifications are sometimes not independent of each other.

Classification of Solution Regions

In terms of the solution region or problem domain, the problem could be

- **Interior problem, also variably called an inner, closed, or bounded problem, or**
- **Exterior problem, also variably called an outer, open, or unbounded problem.**



Solution region R with boundary S.

- Consider the solution region R with boundary S.
- If part or all of S is at infinity, R is exterior/open, otherwise R is interior/closed.
- For example, wave propagation in a waveguide is an interior problem, whereas while wave propagation in free space — scattering of EM waves by raindrops, and radiation from a dipole antenna— are exterior problems

Classification of Solution Regions

A problem can also be classified in terms of the electrical, constitutive properties *(σ, ε, μ)* of the solution region. The solution region could be

- ***linear (or nonlinear),***
- ***homogeneous (or inhomogeneous),***
- ***isotropic (or anisotropic).***

We shall be concerned, for the most part, with linear, homogeneous, isotropic media.

Classification of Differential Equations

EM problems are classified in terms of the equations describing them. The equations could be differential or integral or both. Most EM problems can be stated in terms of an operator equation

$$L\phi = g$$

where *L* is an operator (differential, integral, or integro-differential), *g* is the known excitation or source, and ϕ is the unknown function to be determined. A typical example is the electrostatic problem involving Poisson's equation.

$$-\nabla^2 V = \rho / \varepsilon$$

Laplacian operator: $L = -\nabla^2$ **Source term:** $g = \rho / \varepsilon$ **and** $\phi = V$

In integral form, Poisson's equation is of the form $V = \int \frac{\rho dv}{4\pi\varepsilon r^2}$

so that $L = \int \frac{dv}{4\pi r^2}, \quad g = V, \quad \phi = \frac{\rho}{\varepsilon}$

In this section, we shall limit our discussion to differential equations

Classification of Differential Equations

EM problems involve linear, second-order differential equations. In general, a second-order partial differential equation (PDE) is given by

$$a\frac{\partial^2\phi}{\partial x^2}+b\frac{\partial^2\phi}{\partial x\partial y}+c\frac{\partial^2\phi}{\partial y^2}+d\frac{\partial\phi}{\partial x}+e\frac{\partial\phi}{\partial y}+f\phi=g$$

$$a\phi_{xx}+b\phi_{xy}+c\phi_{yy}+d\phi_x+e\phi_{xy}+f\phi=g$$

- **The coefficients, *a, b* and *c* in general are functions of *x* and *y*; they may also depend on ϕ itself, in which case the PDE is said to be *nonlinear*.**

- **A PDE in which g(x, y) in *equals* zero is termed homogeneous; it is inhomogeneous if g(x, y) *is not equal to* 0.**

Differential operator ⟹ $$L=a\frac{\partial^2}{\partial x^2}+b\frac{\partial^2}{\partial x\partial y}+c\frac{\partial^2}{\partial y^2}+d\frac{\partial}{\partial x}+e\frac{\partial}{\partial y}+f$$

Classification of Differential Equations

A PDE in general can have both *boundary* values and *initial* values. PDEs whose boundary conditions are specified are called *steady-state equations*. If only initial values are specified, they are called *transient equations*.

Any linear second-order PDE can be classified as *elliptic*, *hyperbolic*, or *parabolic* depending on the coefficients *a, b*, and *c*.

$$\begin{aligned} \text{elliptic if} \quad & b^2 - 4ac < 0 \\ \text{hyperbolic if} \quad & b^2 - 4ac > 0 \\ \text{parabolic if} \quad & b^2 - 4ac = 0 \end{aligned}$$

The terms *hyperbolic, parabolic,* and *elliptic* are derived from the fact that the quadratic equation

$$ax^2 + bxy + cy^2 + dx + ey + f = 0$$

represents a hyperbola, parabola, or ellipse if $b^2 - 4ac$ is positive, zero, or negative, respectively.

Classification of Differential Equations

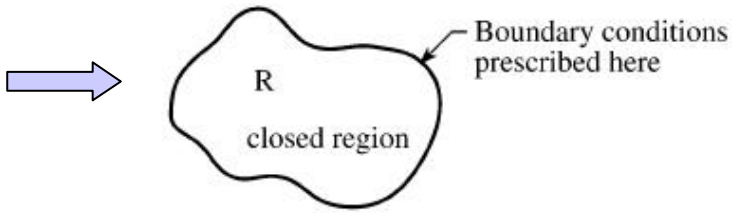
In each of these categories, there are PDEs that model certain physical phenomena. Such phenomena are not limited to EM but extend to almost all areas of science and engineering. Thus the mathematical model specified here arises in problems involving heat transfer, boundary-layer flow, vibrations, elasticity, electrostatic, wave propagation, and so on.

Elliptic PDEs are associated with steady-state phenomena, i.e., boundary-value problems. Typical examples of this type of PDE include Laplace's equation and Poisson equation

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \qquad \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = g(x, y)$$

where in both cases *a* = *c* = 1, *b* = 0.

An elliptic PDE usually models an interior problem, and hence the solution region is usually closed or bounded



Classification of Differential Equations

Hyperbolic PDEs arise in propagation problems. The solution region is usually *open* so that a solution advances outward indefinitely from *initial conditions* while always satisfying *specified boundary conditions*. A typical example of hyperbolic PDE is the wave equation in one dimension

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{1}{v^2}\frac{\partial^2 \phi}{\partial t^2}$$

where $a = v^2$, $b = 0$, $c = -1$.

If the time-dependence is suppressed the equation is merely the steady-state solution.

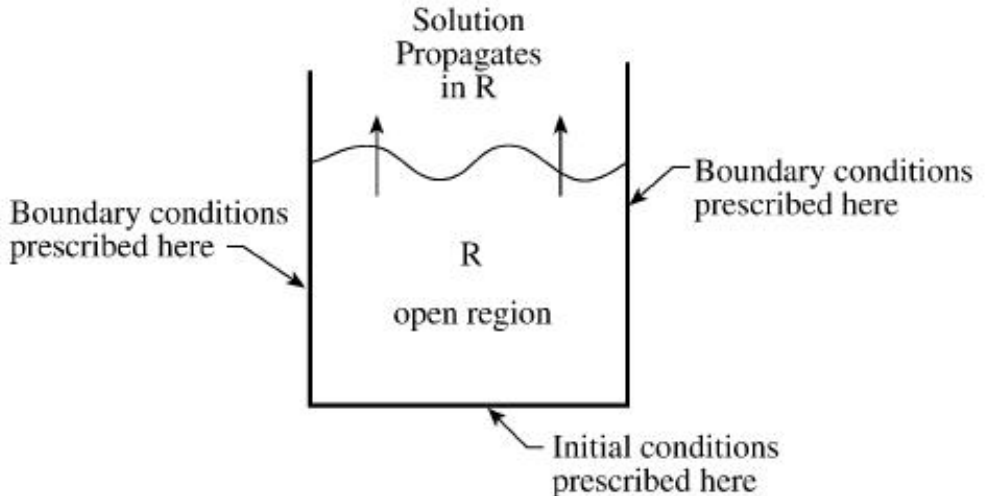
Parabolic PDEs are generally associated with problems in which the quantity of interest varies slowly in comparison with the random motions which produce the variations. The most common parabolic PDE is the diffusion (or heat) equation in one dimension

$$\frac{\partial^2 \phi}{\partial x^2} = k\frac{\partial \phi}{\partial t}$$

where $a = 1$, $b = 0 = c$.

Classification of Differential Equations

- **In hyperbolic and parabolic PDEs, the solution region is usually open**



- **The initial and boundary conditions typically associated with parabolic equations resemble those for hyperbolic problems except that only one initial condition at t=0 is necessary since parabolic equation is only first order in time.**

- **Also, parabolic and hyperbolic equations are solved using similar techniques, whereas elliptic equations are usually more difficult and require different techniques.**

Classification of Differential Equations

Note that:

- **Since the coefficients *a, b*, and *c* are in general functions of *x* and *y*, the classification of the second-order PDE may change from point to point in the solution region**
- **PDEs with more than two independent variables *(x, y, z, t, ...)* may not fit as neatly into the classification above.**

Classification of Partial Differential Equations

Type	Sign of $b^2 - 4ac$	Example	Solution region
Elliptic	$-$	Laplace's equation: $\Phi_{xx} + \Phi_{yy} = 0$	Closed
Hyperbolic	$+$	Wave equation: $u^2\Phi_{xx} = \Phi_{tt}$	Open
Parabolic	0	Diffusion equation: $\Phi_{xx} = k\Phi_t$	Open

Classification of Differential Equations

The type of problem represented by $L\phi=g$ is said to be *deterministic,* since the quantity of interest can be determined directly. Another type of problem where the quantity is found indirectly is called *nondeterministic* or *eigenvalue.* The *standard eigenproblem* is of the form

$$L\phi = \lambda\phi$$

A more general version is the *generalized eigenproblem* having the form

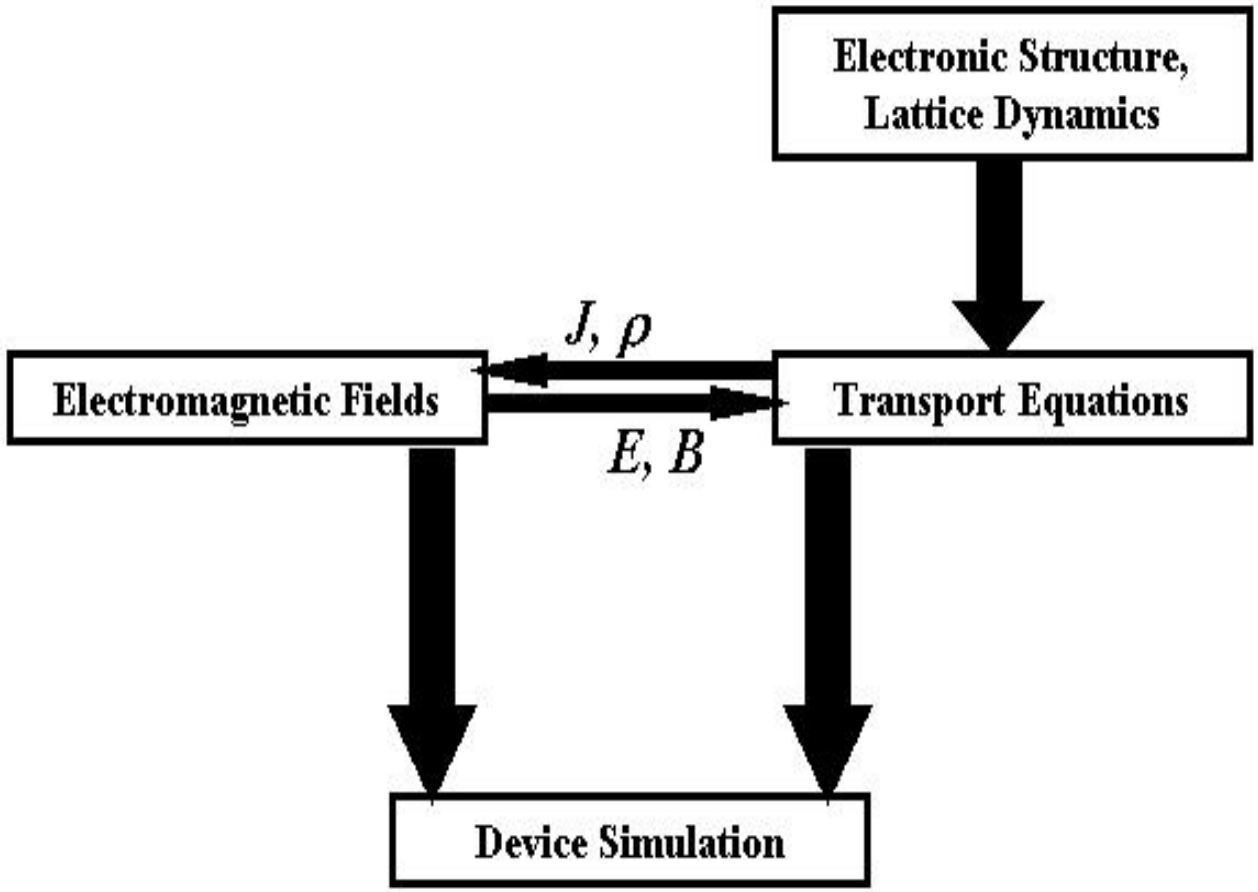
$$L\phi = M\lambda\phi$$

where *M*, like *L*, is a linear operator for EM problems.

Here, only some particular values of λ called *eigenvalues* are permissible; associated with these values are the corresponding solutions called *eigenfunctions.* Eigenproblems are usually encountered in vibration and waveguide problems where the eigenvalues λ correspond to physical quantities such as *resonance* and *cutoff* frequencies, respectively.

Why Numerical Analysis?

Coupling of Transport Equations to Poisson and Band-Structure Solvers



D. Vasileska and S.M. Goodnick, *Computational Electronics*, published by Morgan & Claypool , 2006.

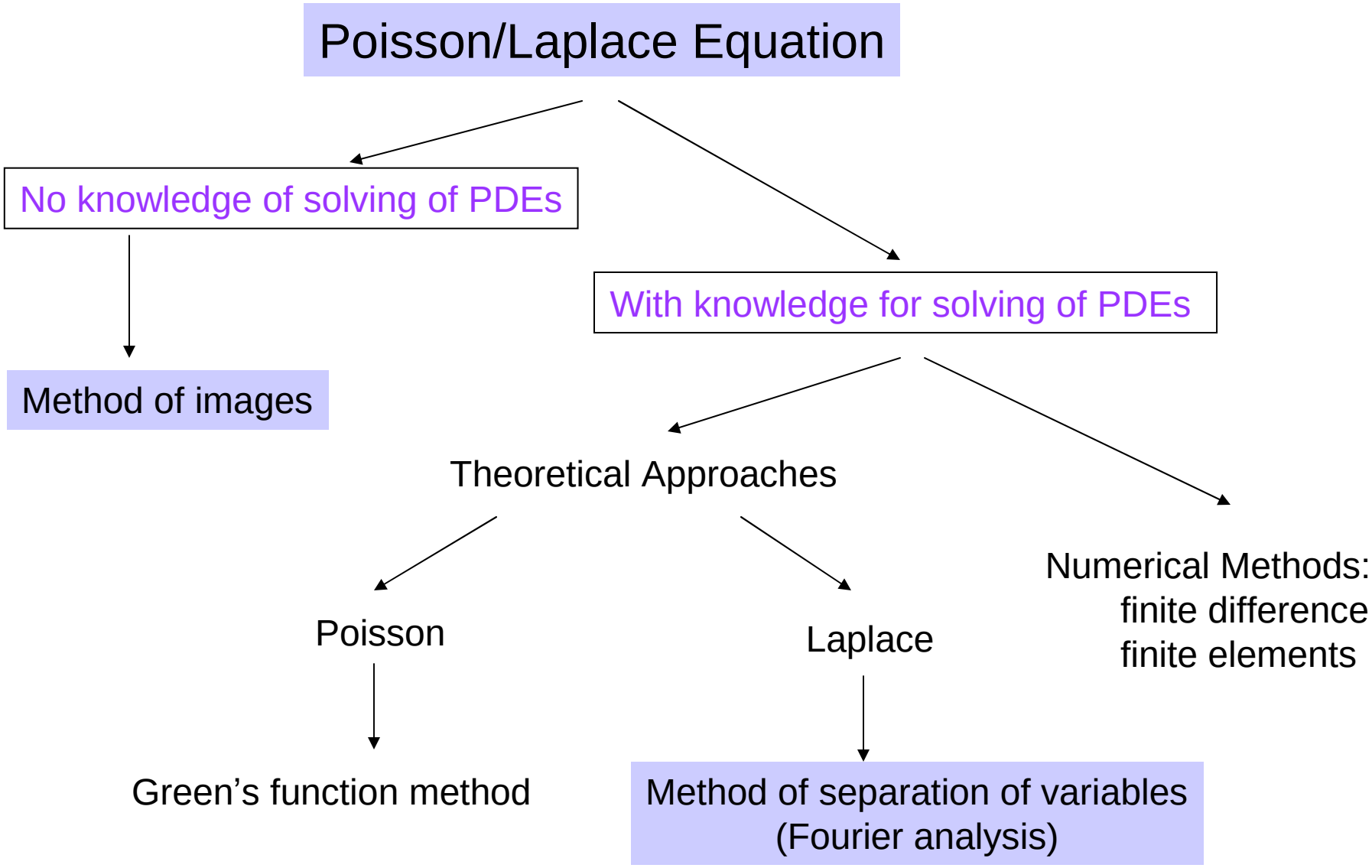
Drift-Diffusion Approach as an Example

Constitutive Equations

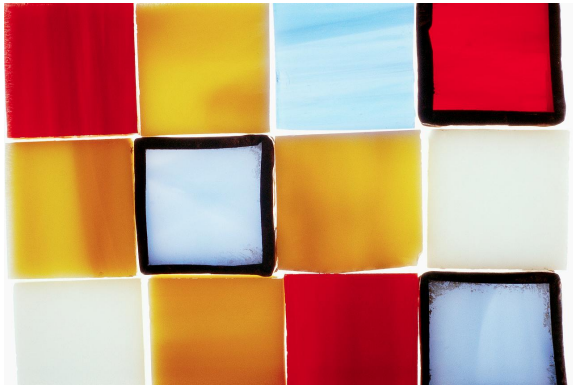
- Poisson $\nabla \cdot \varepsilon \nabla V = -\left(p - n + N_D^+ - N_A^- \right)$
- Continuity Equations
$$\frac{\partial n}{\partial t} = \frac{1}{q} \nabla \cdot \mathbf{J}_n + U_n$$
$$\frac{\partial p}{\partial t} = -\frac{1}{q} \nabla \cdot \mathbf{J}_p + U_p$$
- Current Density Equations
$$J_n = qn(x)\mu_n E(x) + qD_n \frac{dn}{dx}$$
$$J_p = qp(x)\mu_p E(x) - qD_p \frac{dn}{dx}$$

S. Selberherr: "Analysis and Simulation of Semiconductor Devices“, Springer, 1984.

Poisson/Laplace Equation Solution



Finite Difference Methods



Introduction

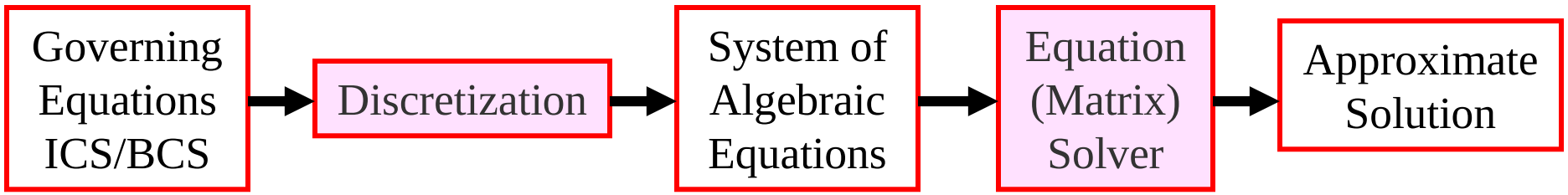
➢**It is rare for real-life EM problems to fall neatly into a class that can be solved by the analytical methods presented in the preceding chapter. Classical approaches may fail if:**

• the PDE is not linear and cannot be linearized without seriously affecting the result
• the solution region is complex
• the boundary conditions are of mixed types
• the boundary conditions are time-dependent
• the medium is inhomogeneous or anisotropic

➢**Whenever a problem with such complexity arises, numerical solutions must be employed.**

➢**Of the numerical methods available for solving PDEs, those employing finite differences are more easily understood, more frequently used, and more universally applicable than any other.**

Numerical Solution Details



Continuous Solutions

Finite-Difference

Finite-Volume

Finite-Element

Spectral

Boundary Element

Hybrid

Discrete Nodal Values

Tridiagonal

SOR

Gauss-Seidel

Krylov

Multigrid

$\phi_i(x,y,z,t)$

$p(x,y,z,t)$

$n(x,y,z,t)$

D. Vasileska, *EEE533 Semiconductor Device and Process Simulation* Lecture Notes, Arizona State University, Tempe, AZ.

Introduction

- **The finite difference method (FDM) was first developed by A.Thom in the 1920s under the title "the method of squares" to solve nonlinear hydrodynamic equations. Since then, the method has found applications in solving different field problems.**
- **The finite difference techniques are based upon approximations which permit replacing differential equations by finite difference equations.**
- **These finite difference approximations are algebraic in form; they relate the value of the dependent variable at a point in the solution region to the values at some neighboring points. Thus a finite difference solution basically involves three steps:**

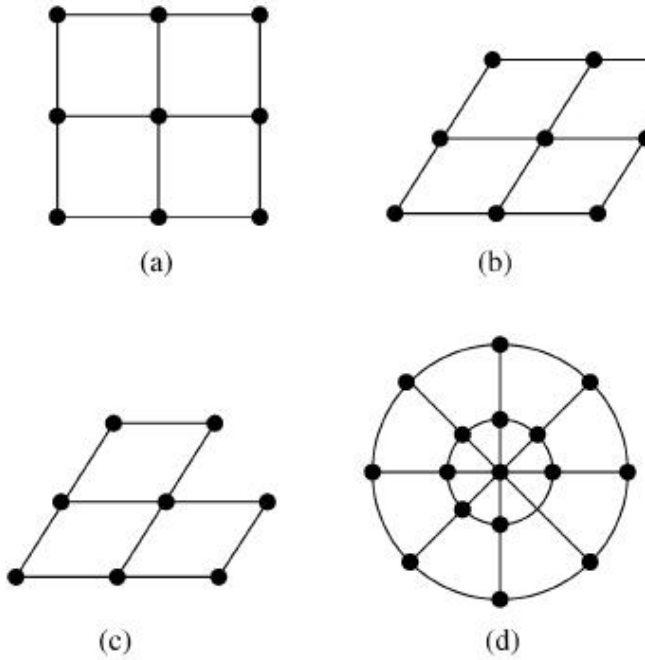
(1) dividing the solution region into a grid of nodes

(2) approximating the given differential equation by finite difference equivalent that relates the dependent variable at a point in the solution region to its values at the neighboring points

(3) solving the difference equations subject to the prescribed boundary conditions and/or initial conditions

MESH TYPE

The course of action taken in three steps is dictated by the nature of the problem being solved, the solution region, and the boundary conditions. The most commonly used grid patterns for two-dimensional problems are



Common grid patterns: (a) rectangular grid, (b) skew grid, (c) triangular grid, (d) circular grid.

Mesh Size

- Regarding the grid set-up, there are several points that need to be made:
  - ✓ In critical device regions, where the charge density varies very rapidly, the mesh spacing has to be smaller than the extrinsic Debye length determined from the maximum doping concentration in that location of the device

$$L_D = \sqrt{\frac{\varepsilon k_B T}{N_{\max} e^2}}$$

  - ✓ Cartesian grid is preferred for particle-based simulations
  - ✓ It is always necessary to minimize the number of node points to achieve faster convergence
  - ✓ A regular grid (with small mesh aspect ratios) is needed for faster convergence

Example for Meshing

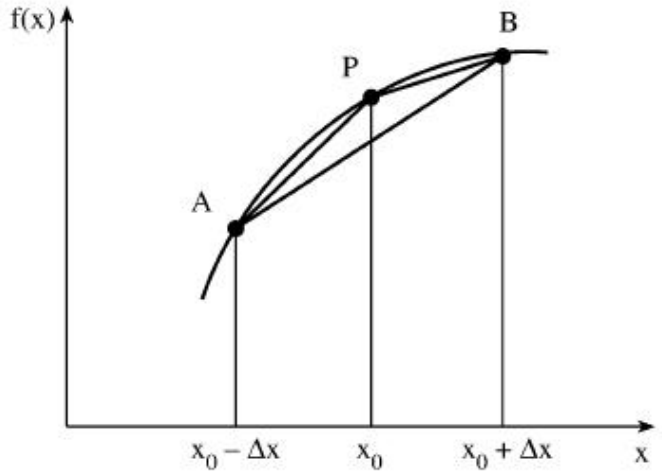
The function below is used to generate non-uniform mesh with constant mesh aspect ratio **r**. Input parameters are initial mesh size (**X0**), total number of mesh points (**N**) and the size of the domain over which we want these mesh points distributed (**XT**).

```
FUNCTION R_COEFF(XO, N, XT)
IMPLICIT REAL*4(A-H, O-Z)
LOGICAL FLAG$CONV
flag$conv = .false.
r = 3.
do while(.not.flag$conv)
    term1 = 1./(r-1.D0)
    term2 = r**float(N)-1.D0
    rnum  = xo*term1*term2 - xt
    term3 = float(N)*r**float(N-1)
    denom = xo*(term1*term3-term1*term1*term2)
    r_new = r - rnum/denom
    error = abs(r-r_new)/abs(r_new)
    if(error.le.1.e-10)then
       flag$conv = .true.
    else
       r = r_new
    endif
enddo
r_coeff = r
RETURN
END
```

This is determined by the maximum doping in the device in a particular region.

Finite Difference Schemes

Before finding the finite difference solutions to specific PDEs, we will look at how one constructs finite difference approximations from a given differential equation. This essentially involves estimating derivatives numerically. Let's assume *f(x)* shown below:



Estimates for the derivative of *f (x)* at *P* using forward, backward, and central differences.

Finite Difference Schemes

We can approximate derivative of f(x), slope or the tangent at P by the slope of the arc PB, giving the *forward-difference* formula,

$$f'(x_0) \cong \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

or the slope of the arc AP, yielding the *backward-difference* formula,

$$f'(x_0) \cong \frac{f(x_0) - f(x_0 - \Delta x)}{\Delta x}$$

or the slope of the arc AB, resulting in the *central-difference* formula,

$$f'(x_0) \cong \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2\Delta x}$$

Finite Difference Schemes

We can also estimate the second derivative of *f (x)* at P as

$$f''(x_0) \cong \frac{f'(x_0 + \Delta x/2) - f(x_0 - \Delta x/2)}{\Delta x}$$

$$= \frac{1}{\Delta x}\left[\frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} - \frac{f(x_0) - f(x_0 - \Delta x)}{\Delta x}\right]$$

or

$$f''(x_0) \cong \frac{f(x_0 + \Delta x) - 2f(x_0) + f(x_0 - \Delta x)}{\Delta x^2}$$

Any approximation of a derivative in terms of values at a discrete set of points is called *finite difference* approximation.

Finite Difference Schemes

The approach used above in obtaining finite difference approximations is rather intuitive. A more general approach is using Taylor's series. According to the wellknown expansion,

$$f(x_0+\Delta x) = f(x_0) + \Delta x f'(x_0) + \frac{1}{2!}(\Delta x)^2 f''(x_0) + \frac{1}{3!}(\Delta x)^3 f'''(x_0) + \cdots$$

and

$$f(x_0-\Delta x) = f(x_0) - \Delta x f'(x_0) + \frac{1}{2!}(\Delta x)^2 f''(x_0) - \frac{1}{3!}(\Delta x)^3 f'''(x_0) + \cdots$$

Upon adding these expansions,

$$f(x_0+\Delta x) + f(x_0-\Delta x) = 2f(x_0) + (\Delta x)^2 f''(x_0) + O(\Delta x)^4$$

where $O(\Delta x)^4$ is the error introduced by truncating the series. We say that this error is of the order $(\Delta x)^4$ or simply $O(\Delta x)^4$. Therefore, $O(\Delta x)^4$ represents terms that are not greater than $(\Delta x)^4$. Assuming that these terms are negligible,

$$f''(x_0) \cong \frac{f(x_0+\Delta x) - 2f(x_0) + f(x_0-\Delta x)}{\Delta x^2}$$

Finite Difference Schemes

Subtracting

$$f(x_0 - \Delta x) = f(x_0) - \Delta x f'(x_0) + \frac{1}{2!}(\Delta x)^2 f''(x_0) - \frac{1}{3!}(\Delta x)^3 f'''(x_0) + \cdots$$

from

$$f(x_0 + \Delta x) = f(x_0) + \Delta x f'(x_0) + \frac{1}{2!}(\Delta x)^2 f''(x_0) + \frac{1}{3!}(\Delta x)^3 f'''(x_0) + \cdots$$

We obtain

$$f(x_0 + \Delta x) - f(x_0 - \Delta x) = 2(\Delta x) f'(x_0) + \mathrm{O}(\Delta x)^3$$

and neglecting terms of the order $(\Delta x)^3$ yields

$$f'(x_0) \cong \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2\Delta x}$$

This shows that the leading errors of the order $(\Delta x)^2$. Similarly, the *forward* and *backward difference* formula have truncation errors of $O(\Delta x)$.

Finite Differencing of Elliptic PDEs

A typical elliptic PDE is Poisson's equation, which in two dimensions is given by

$$\nabla^2\Phi = \frac{\partial^2\Phi}{\partial x^2} + \frac{\partial^2\Phi}{\partial y^2} = g(x, y)$$

We can use the central difference approximation for the partial derivatives of which the simplest forms are

$$\frac{\partial^2\Phi}{\partial x^2} = \frac{\Phi(i+1, j) - 2\Phi(i, j) + \Phi(i-1, j)}{(\Delta x)^2} + \mathrm{O}(\Delta x)^2$$

$$\frac{\partial^2\Phi}{\partial y^2} = \frac{\Phi(i, j+1) - 2\Phi(i, j) + \Phi(i, j-1)}{(\Delta y)^2} + \mathrm{O}(\Delta x)^2$$

where $x = i\Delta x$, $y = j\Delta y$, and $i, j = 0, 1, 2, \ldots$.

Finite Differencing of Elliptic PDEs

If we assume that $\Delta x = \Delta y = h$, to simplify calculations, we get

$$\left[\Phi(i+1, j) + \Phi(i-1, j) + \Phi(i, j+1) + \Phi(i, j-1)\right] - 4\Phi(i, j) = h^2 g(i, j)$$

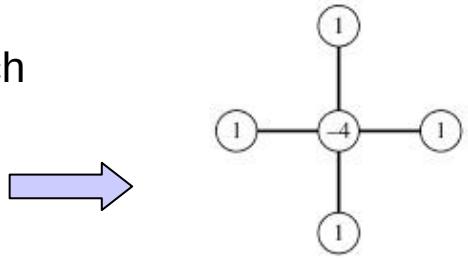
or

$$\Phi(i, j) = \frac{1}{4}\left[\Phi(i+1, j) + \Phi(i-1, j) + \Phi(i, j+1) + \Phi(i, j-1) - h^2 g(i, j)\right]$$

at every point *(i, j)* in the mesh for Poisson's equation. The spatial increment *h* is called the *mesh size.* A special case is when the source term vanishes, i.e., *g(x, y)* = 0. This leads to Laplace's equation. Thus for Laplace's equation,

$$\Phi(i, j) = \frac{1}{4}\left[\Phi(i+1, j) + \Phi(i-1, j) + \Phi(i, j+1) + \Phi(i, j-1)\right]$$

It is worth noting that the value of Φ for each point is the average of those at the four surrounding points. The five-point computational molecule for the second order difference scheme is illustrated, where values of the coefficients are shown.



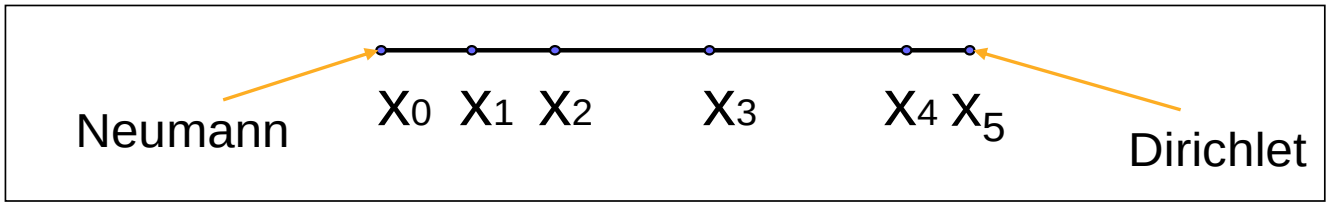
Computational molecule for Laplace's equation based on second order approximation

Boundary Conditions

- There are three types of boundary conditions that are specified during the discretization process of the Poisson equation:
  - Dirichlet (this is a boundary condition on the potential)
  - Neumann (this is a boundary condition on the derivative of the potential, i.e. the electric field)
  - Mixed boundary condition (combination of Dirichlet and Neumann boundary conditions)
- Note that when applying the boundary conditions for a particular structure of interest, at least one point MUST have Dirichlet boundary conditions specified on it to get the connection to the real world.

1D Discretization

- The resultant finite difference equations can be represented in a matrix form Au= f, where:



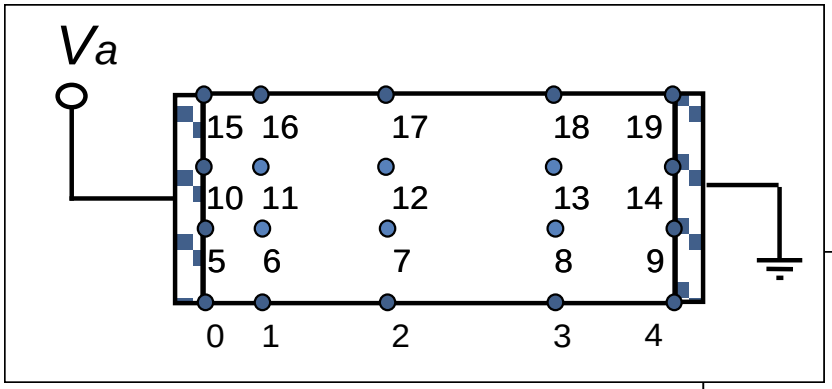
$$\mathbf{u} = (\Phi_0, \Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5)\ , \quad \mathbf{f} = (f_0, f_1, f_2, f_3, f_4, \Phi_5)\ ,$$

$$\mathbf{A} = \begin{pmatrix} b_0 & 2c_0 & & & & \\ a_1 & b_1 & c_1 & & 0 & \\ & a_2 & b_2 & c_2 & & \\ & & a_3 & b_3 & c_3 & \\ & 0 & & a_4 & b_4 & c_4 \\ & & & & 0 & 1 \end{pmatrix} \text{ where}$$

$$a_i = \frac{2}{dx_i^w (dx_i^w + dx_i^e)}\ ;\ b_i = \frac{2}{dx_i^e dx_i^w}\ ;\ c_i = \frac{2}{dx_i^e (dx_i^w + dx_i^e)}$$

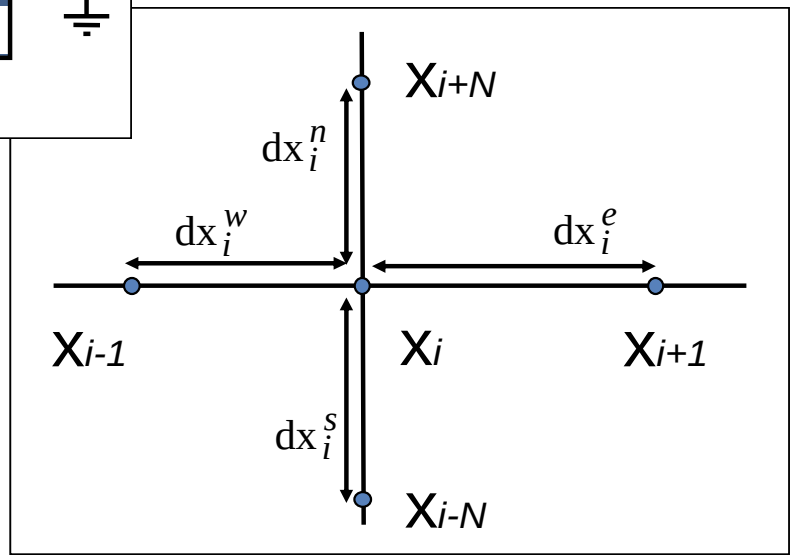
2D Discretization

- In 2D, the finite-difference discretization of the Poisson equation leads to a five point stencil:



N=5,M=4

Dirichlet: 0,4,5,9,10,14,15,19
Neuman: 1,2,3,16,17,18



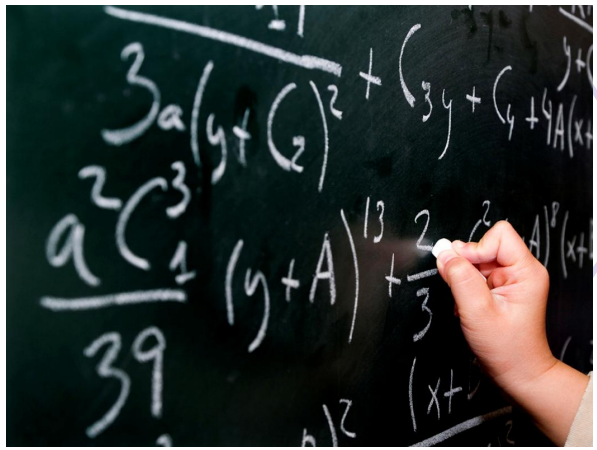
2D Discretization (cont'd)

$$
\begin{pmatrix}
1 & & & & & & & & & & & & & & & & & & & \\
b_1 & c_1 & d_1 & & & & 2e_1 & & & & & & & & & & & & & \\
 & b_2 & c_2 & d_2 & & & & 2e_2 & & & & & & & & & & & & \\
 & & b_3 & c_3 & d_3 & & & & 2e_3 & & & & & & & & & & & \\
 & & & & 1 & & & & & & & & & & & & & & & \\
 & & & & & 1 & & & & & & & & & & & & & & \\
a_6 & & & & & b_6 & c_6 & d_6 & & & & & e_6 & & & & & & & \\
 & a_7 & & & & & b_7 & c_7 & d_7 & & & & & e_7 & & & & & & \\
 & & a_8 & & & & & b_8 & c_8 & d_8 & & & & & e_8 & & & & & \\
 & & & & & & & & & 1 & & & & & & & & & & \\
 & & & & & & & & & & 1 & & & & & & & & & \\
 & & & & & a_{11} & & & & & b_{11} & c_{11} & d_{11} & & & & e_{11} & & & \\
 & & & & & & a_{12} & & & & & b_{12} & c_{12} & d_{12} & & & & e_{12} & & \\
 & & & & & & & a_{13} & & & & & b_{13} & c_{13} & d_{13} & & & & e_{13} & \\
 & & & & & & & & & & & & & & 1 & & & & & \\
 & & & & & & & & & & & & & & & 1 & & & & \\
 & & & & & & & & & & 2a_{16} & & & & & b_{16} & c_{16} & d_{16} & & \\
 & & & & & & & & & & & 2a_{17} & & & & & b_{17} & c_{17} & d_{17} & \\
 & & & & & & & & & & & & 2a_{18} & & & & & b_{18} & c_{18} & d_{18} \\
 & & & & & & & & & & & & & & & & & & & 1
\end{pmatrix}
\begin{pmatrix} \Phi_0 \\ \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \\ \Phi_5 \\ \Phi_6 \\ \Phi_7 \\ \Phi_8 \\ \Phi_9 \\ \Phi_{10} \\ \Phi_{11} \\ \Phi_{12} \\ \Phi_{13} \\ \Phi_{14} \\ \Phi_{15} \\ \Phi_{16} \\ \Phi_{17} \\ \Phi_{18} \\ \Phi_{19} \end{pmatrix}
=
\begin{pmatrix} V_a \\ f_1 \\ f_2 \\ f_3 \\ 0 \\ V_a \\ f_6 \\ f_7 \\ f_8 \\ 0 \\ V_a \\ f_{11} \\ f_{12} \\ f_{13} \\ 0 \\ V_a \\ f_{16} \\ f_{17} \\ f_{18} \\ 0 \end{pmatrix}
$$

Dirichlet: 0,4,5,9,10,14,15,19

Neuman: 1,2,3,16,17,18

Poisson Equation Linearization and Discretization



Poisson Equation Linearization

- The 1D Poisson equation is of the form:

$$\frac{d^2\varphi}{dx^2} = -\frac{e}{\varepsilon}\left(p - n + N_D - N_A\right)$$

$$n = n_i \exp\left(\frac{E_F - E_i}{k_B T}\right) = n_i \exp(\varphi / V_T)$$

$$p = n_i \exp\left(\frac{E_i - E_F}{k_B T}\right) = n_i \exp(-\varphi / V_T)$$

$\varphi^{new} => \varphi^{old} + \delta$

$$\frac{d^2\varphi^{new}}{dx^2} = -\frac{en_i}{\varepsilon}\left(e^{-\varphi^{old}/V_T} - e^{\varphi^{old}/V_T} + C/n_i\right) +$$

$$+\frac{en_i}{\varepsilon}\delta\left(e^{-\varphi^{old}/V_T} + e^{\varphi^{old}/V_T}\right)$$

$$\frac{d^2\varphi^{new}}{dx^2} - \frac{en_i}{\varepsilon}\left(e^{-\varphi/V_T} + e^{\varphi/V_T}\right)\varphi^{new} = -\frac{en_i}{\varepsilon}\left(e^{-\varphi^{old}/V_T} - e^{\varphi^{old}/V_T} + C/n_i\right) -$$

$$-\frac{en_i}{\varepsilon}\left(e^{-\varphi^{old}/V_T} + e^{\varphi^{old}/V_T}\right)\varphi^{old}$$

$$\delta = \varphi^{new} - \varphi^{old}$$

Renormalized Form: $L_D=\sqrt{\varepsilon V_T/qn_i}$

$$\frac{d^2\varphi^{new}}{dx^2} = -\left(p - n + C\right) + \delta\left(p + n\right)$$

$$\frac{d^2\varphi^{new}}{dx^2} - \left(p + n\right)\varphi^{new} = -\left(p - n + C\right) - \left(p + n\right)\varphi^{old}$$

$$\delta = \varphi^{new} - \varphi^{old}$$

Finite Difference Representation

$$\frac{1}{\Delta^2}\varphi_{i+1}^{n+1} - \left(\frac{2}{\Delta^2} + n_i + p_i\right)\varphi_i^{n+1} + \frac{1}{\Delta^2}\varphi_{i-1}^{n+1} =$$

$$= -(p_i - n_i + C_i) - (p_i + n_i)\varphi_i^n$$

$$n+1 \rightarrow new$$

$$n \rightarrow old$$

Equilibrium:

$$n_i = \exp(\varphi_i^n), \quad p_i = \exp(-\varphi_i^n)$$

Non-Equilibrium:

n calculated using PM coupling and *p* still calculated as in equilibrium case (quasi-equilibrium approximation)

Solution of Matrix Equations

Solution Methods

- The variety of methods for solving Poisson equation include:

Direct methods

- Gaussian elimination
- *LU* decomposition method

Iterative methods

- Mesh relaxation methods
  - Jacobi
  - Gaus-Seidel
  - Successive over-relaxation method (SOR)
  - Alternating directions implicit (ADI) method
- Matrix methods
  - Thomas tridiagonal form
  - Sparse matrix methods: Stone's Strongly Implicit Procedure (SIP), Incomplete Lower-Upper (ILU) decomposition method
  - Conjugate Gradient (CG) methods: Incomplete Choleski Conjugate Gradient (ICCG), Bi-CGSTAB
  - Multi–Grid (MG) method

LU Decomposition

- If LU decomposition exists, then for a tri-diagonal matrix **A**, resulting from the finite-difference discretization of the 1D Poisson equation, one can write

$$\begin{pmatrix} a_1 & c_1 & & & \\ b_2 & a_2 & c_2 & & \\ & \dots & \dots & \dots & \\ & & b_{n-1} & a_{n-1} & c_{n-1} \\ & & & b_n & a_n \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ \beta_2 & 1 & & & \\ & \beta_3 & \dots & & \\ & & \dots & \dots & \\ & & & \beta_n & 1 \end{pmatrix} \begin{pmatrix} \alpha_1 & c_1 & & & \\ & \alpha_2 & c_2 & & \\ & & \dots & \dots & \\ & & & \dots & c_{n-1} \\ & & & & \alpha_n \end{pmatrix}$$

where $\alpha_1 = a_1, \quad \beta_k = \dfrac{b_k}{\alpha_{k-1}}, \quad \alpha_k = a_k - \beta_k c_{k-1}, \quad k = 2,3,\dots,n$

Then, the solution is found by forward and back substitution:

$$g_1 = f_1, \qquad g_i = f_i - \beta_i g_{i-1}, \quad i = 2,3,\dots,n,$$

$$x_n = \frac{g_n}{\alpha_n}, \qquad x_i = \frac{g_i - c_i x_{i+1}}{\alpha_i}, \quad i = n-1,\dots,2,1$$

Iterative Methods

- Iterative (or relaxation) methods start with a first approximation which is successively improved by the repeated application (i.e. the “iteration”) of the same algorithm, until a sufficient accuracy is obtained.

- In this way, the original approximation is “relaxed” toward the exact solution which is numerically more stable.

- Iterative methods are used most often for large sparse system of equations, and always when a good approximation of the solution is known.

- Error analysis and convergence rate are two crucial aspects of the theory of iterative methods.

Error Equation

- The simple iterative methods for the solution of **Ax** = **f** proceed in the following manner:

  A sequence of approximations $\mathbf{v}^0,\mathbf{v}^1,...,\mathbf{v}^n,...,$ of **x** is constructed that converges to **x**. Let $\mathbf{v}^i$ be an approximation to **x** after the *i-th* iteration. One may define the residual

  $$\mathbf{r}^i = \mathbf{f} - \mathbf{A}\mathbf{v}^i$$

  as a computable measure of the deviation of $\mathbf{v}^i$ from **x**. Next, the algebraic error $\mathbf{e}^i$ of the approximation $\mathbf{v}^i$ is defined by

  $$\mathbf{e}^i = \mathbf{x} - \mathbf{v}^i.$$

  From the previous equations one can see that $\mathbf{e}^i$ obeys the so-called residual equation:

  $$\mathbf{A}\mathbf{e}^i = \mathbf{r}^i$$

Jacobi, Gauss-Seidel Methods

The expansion of **Ax=f** gives the relation

$$x_k = \frac{-\sum_{j=1,\, j\neq k}^{n} a_{ki} x_j + f_k}{a_{kk}}, \quad k = 1,2,\ldots,n\ ; \quad a_{kk} \neq 0$$

In **Jacobi's method** the sequence $\mathbf{v^0, v^1, \ldots, v^n, \ldots}$ is computed by

$$v_k^{i+1} = \frac{-\sum_{j=1,\, j\neq k}^{n} a_{kj} v_j^i + f_k}{a_{kk}}, \quad k = 1,2,\ldots,n\ ; \quad a_{kk} \neq 0$$

Note that one does not use the improved values until after a complete iteration. The closely related **Gauss-Seidel's method** solves this problem as follows:

$$v_k^{i+1} = \frac{-\sum_{j=1}^{k-1} a_{kj} v_j^{i+1} - \sum_{j=k+1}^{n} a_{kj} v_j^i + f_k}{a_{kk}}, \quad k = 1,2,\ldots,n\ ; \quad a_{kk} \neq 0$$

SOR Method

By a simple modification of Gauss-Seidel's method it is often possible to improve the rate of convergence. Following the definition of residual $\mathbf{r}^i = \mathbf{f} - \mathbf{A}\mathbf{v}^i$, the Gauss-Seidel formula can be written $v_k^{i+1} = v_k^i + r_k^i$, where

$$r_k^i = \frac{-\sum_{j=1}^{k-1} a_{kj} v_j^{i+1} - \sum_{j=k}^{n} a_{kj} v_j^i + f_k}{a_{kk}}, \quad k = 1,2,\ldots,n \; ; \quad a_{kk} \neq 0$$

The iterative method

$$v_k^{i+1} = v_k^i + \omega r_k^i$$

is the so-called **successive over-relaxation (SOR) method.** Here, ω, the relaxation parameter, should be chosen so that the rate of convergence is maximized. The rate of convergence of the SOR is often surprisingly higher than the one of Gauss-Seidel's method. The value of ω depends on the grid spacing, the geometrical shape of the domain, and the type of boundary conditions imposed on it.

Convergence

- ✓Any stationary iterative method can be written in the general form

$$\mathbf{x}^{k+1} = \mathbf{B}\mathbf{x}^{k} + \mathbf{c}$$

- ✓A relation between the errors in successive approximation can be derived by subtracting the equation $\mathbf{x}=\mathbf{B}\mathbf{x}+\mathbf{c}$:

$$\mathbf{x}^{k+1} - \mathrm{x} = \mathbf{B}(\mathbf{x}^{k} - \mathbf{x}) = \ldots = \mathbf{B}^{k+1}(\mathbf{x}^{0} - \mathbf{x})$$

- ✓Now, let **B** have eigenvalues $\lambda_1,\lambda_2,\ldots,\lambda_n$, and assume that the corresponding eigenvectors $\mathbf{u}_1,\mathbf{u}_2,\ldots,\mathbf{u}_n$ are linearly independent. Then we can expand the initial error as

$$\mathbf{x}^{0} - \mathbf{x} = \alpha\mathbf{u}_1 + \alpha\mathbf{u}_2 + \ldots + \alpha\mathbf{u}_n$$

Convergence (cont'd)

and thus $\mathbf{x}^k - \mathbf{x} = \alpha_1 \lambda_1^k \mathbf{u}_1 + \alpha_2 \lambda_2^k \mathbf{u}_2 + \ldots + \alpha_n \lambda_n^k \mathbf{u}_n$.

This means that the process converges from an arbitrary approximation if and only if $|\lambda_i|<1$, $i=1,2,\ldots,n$.

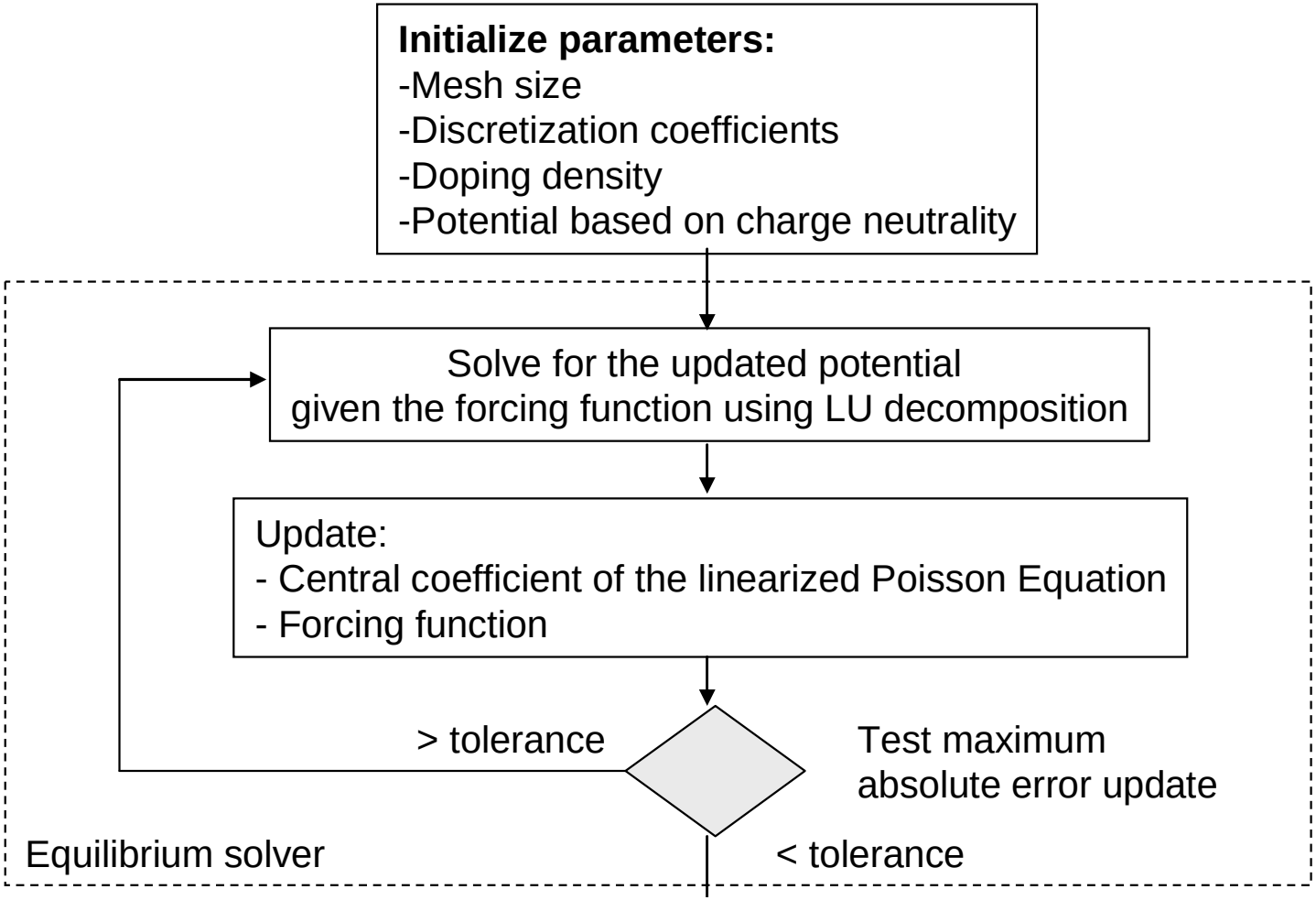
Theorem: A necessary and sufficient condition for a stationary iterative method $\mathbf{x}^{k+1} = \mathbf{B}\mathbf{x}^k + \mathbf{c}$ to converge for an arbitrary initial approximation $\mathbf{x}^0$ is that

$$\rho(\mathbf{B}) = \max_{1 \le i \le n} |\lambda_i(\mathbf{B})| < 1 ,$$

where $\rho(\mathbf{B})$ is called the spectral radius of **B**. For uniform mesh and Dirichlet boundary conditions, one has for the SOR method

$$\omega = \frac{2}{1 + \sqrt{1 - \rho^2}}$$

Flow Chart of Equilibrium Poisson Solver



Criterion for Convergence

There are several criteria for the convergence of the iterative procedure when solving the Poisson equation, but the simplest one is that nowhere on the mesh the absolute value of the potential update is larger than 1E-5 V.

This criterion has shown to be sufficient for all device simulations that have been performed within the Computational Electronics community.

Classification of EM Problems, Finite Difference Discretization Scheme and Solution of Sparse Matrix Equation Ax=b

Dragica Vasileska, Professor
Arizona State University