①

Ch-6 - lec -9 ?

FEM & Diffusion theory so far.

(conservation)

Continuity equation: $\nabla \cdot \underline{J}(\underline{r}) + \Sigma_a(\underline{r})\phi(\underline{r}) = S(\underline{r})$

⇓ diffusion approximation (fick's law) + 1D

Solve

$$\begin{cases} -\dfrac{\partial}{\partial x} D(x) \dfrac{\partial \phi}{\partial x} + \Sigma_a \phi = S \\[2ex] \text{BC: Reflecting: } \left.\dfrac{\partial \phi}{\partial x}\right|_{x = x_L \text{ or } x_R} = 0 \\[2ex] \text{Vacuum: } \left. D\dfrac{\partial \phi}{\partial x}\right|_{x_L} = \left.\dfrac{\phi}{2}\right|_{x_L} \quad \left| \quad \left. -D\dfrac{\partial \phi}{\partial x}\right|_{x_R} = \left.\dfrac{\phi}{2}\right|_{x_R} \right. \end{cases}$$

FEM

① Discretize: $E_1 \; E_2 \; E_3$; $x_1 \; x_2 \; x_3 \; x_4$

② Get weak form:

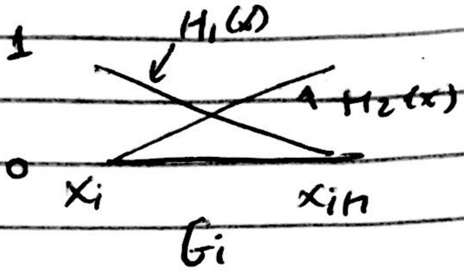
Weighted residual: $\displaystyle\int_{x_i}^{x_{i+1}} dx \, \omega \left(-\frac{\partial}{\partial x} D \frac{\partial \phi}{\partial x} + \Sigma_a \phi - S \right) = 0$

arbitrary weight fn — residual to be minimized

weak form: $\displaystyle D_i \int_{x_i}^{x_{i+1}} \frac{\partial \omega}{\partial x} \frac{\partial \phi}{\partial x} dx - \left[\omega D_i \frac{\partial \phi}{\partial x} \right]_{x_i}^{x_{i+1}} + \Sigma_{a_i} \int_{x_i}^{x_{i+1}} \omega \phi \, dx = \int_{x_i}^{x_{i+1}} \omega S \, dx.$

③ Choose weight and trial fn. We do Galerkin.

Over each element we have



$$\phi_{e_i} = H_1(x)\phi_i + H_2(x)\phi_{i+1}$$

$$H_1(x) = \frac{x_{i+1}-x}{x_{i+1}-x_i} \qquad H_2(x) = \frac{x-x_i}{x_{i+1}-x_i}$$

$$= \frac{x_{i+1}-x}{h_i} \quad / \qquad = \frac{x_2-x_i}{h_i} \quad /$$

$$\phi_{e_i} = \begin{bmatrix} H_1(x) & H_2(x) \end{bmatrix} \begin{bmatrix} \phi_i \\ \phi_{i+1} \end{bmatrix} = [H][\tilde{\phi}] \quad //$$

Galerkin so

$$w = w^T = \begin{bmatrix} w_1 & w_2 \end{bmatrix} \begin{bmatrix} H_1(x) \\ H_2(x) \end{bmatrix} = [\tilde{w}]^T [H]^T \quad //$$

④ Equation for each element

$$\begin{bmatrix} D_i/h_i + \Sigma a_i h_i/3 & -D_i/h_i + \Sigma a_i h_i/6 \\ -D_i/h_i + \Sigma a_i h_i/6 & D_i/h_i + \Sigma a_i h_i/3 \end{bmatrix} \begin{bmatrix} \phi_i \\ \phi_{i+1} \end{bmatrix} - \begin{bmatrix} -D_i \left.\frac{d\phi}{dx}\right|_{x_i} \\ D_i \left.\frac{d\phi}{dx}\right|_{x_{i+1}} \end{bmatrix}$$

$$= S_i \begin{bmatrix} h_i/2 \\ h_i/2 \end{bmatrix} \quad //$$

Scanned with CamScanner

Boundary conditions:

Vacuum: left: $\begin{bmatrix} -\phi(x_l)/2 \\ D_i \dfrac{d\phi}{dx}\Big|_{x_{i+1}} \end{bmatrix}$

right: $\begin{bmatrix} -D_i \dfrac{d\phi}{dx}\Big|_{x_i} \\ -\dfrac{\phi(x_r)}{2} \end{bmatrix}$

Reflecting: left: $\begin{bmatrix} 0 \\ D_i \dfrac{d\phi}{dx}\Big|_{x_{i+1}} \end{bmatrix}$

right: $\begin{bmatrix} -D_i \dfrac{d\phi}{dx}\Big|_{x_i} \\ 0 \end{bmatrix}$

5. Global matrix assembly:

let $\hat{E}_1$, $\hat{E}_2$ & $\hat{E}_3$ represent the 2 by 2 matrices of element 1, 2, 3 respectively. The global matrix looks like

$$\begin{bmatrix} \hat{E}_1^{1,1} + 1/2 & \hat{E}_1^{1,2} & 0 & 0 \\ \hat{E}_1^{2,1} & \hat{E}_1^{2,2} + \hat{E}_2^{1,1} & \hat{E}_2^{1,2} & 0 \\ 0 & \hat{E}_2^{2,1} & \hat{E}_2^{2,2} + \hat{E}_3^{1,1} & \hat{E}_3^{1,2} \\ 0 & 0 & \hat{E}_3^{2,1} & \hat{E}_3^{2,2} + 1/2 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{bmatrix} = \begin{bmatrix} S_1 h_1/2 \\ S_1 h_1/2 + S_2 h_2/2 \\ S_2 h_2/2 + S_3 h_3/2 \\ S_3 h_3/2 \end{bmatrix}$$

Scanned with CamScanner

Call the 4×4 matrix A.

$\Rightarrow \quad [A][\Phi] = [S]$

or simply $A\Phi = S$

(6) Solve the linear system of equations.

Gauss Jacobi:

First thing we do is divide matrix A into an upper triangular matrix (U), lower (L) & a diagonal (D) matrix.

$\hookrightarrow \quad A = D + U + L.$

$\Rightarrow \quad A\Phi = S$

$\hookrightarrow \quad (D + U + L)\Phi = S$

$\Rightarrow \quad D\Phi + (U+L)\Phi = S$

$\Rightarrow \quad D\Phi = -(U+L)\Phi + S.$

Let's call $-(U+L) \rightarrow T.$

$\Rightarrow \quad D\Phi = T\Phi + S$

$\Rightarrow \quad D^{-1}D\Phi = D^{-1}T\Phi + D^{-1}S$

$\Rightarrow \quad \boxed{\Phi = D^{-1}T\Phi + D^{-1}S}$

Now, in order to get our iterative scheme, we lag the Φ on rhs. So:

$$\boxed{\Phi^{n} = D^{-1}T\Phi^{n-1} + D^{-1}S}$$

Scanned with CamScanner

Algorithm

1. Decompose $A \rightarrow L+U+D$.
   so ~~find~~ identify L, U, D.

2. find D^{-1}, $\Rightarrow$ identify T.

3. Guess ϕ (for lagged part on rhs) $\rightarrow \phi^{n-1}$ to same vector.

4. find $\phi^{n} \rightarrow D^{-1}T\phi^{n-1} + D^{-1}s$

5. ~~find~~ check for convergence. Stop iteration if

$$\frac{\|\phi^{n}-\phi^{n-1}\|}{\|\phi^{n}\|} < \text{tolerance}$$

otherwise $\phi^{n-1} = \phi^{n}$ &
repeat 4

Gauss Seidel:

We begin the same way as with Gauss Jacobi.

$A = L+U+D$

$\Rightarrow (L+U+D)\phi = s$. (Now we do something different

$\Rightarrow (L+D)\phi + U\phi = s$

$\Rightarrow (L+D)\phi = -U\phi + s$.

call $-U \rightarrow T$

$\Rightarrow \phi = (L+D)^{-1}T\phi + (L+D)^{-1}s$.

Scanned with CamScanner

2) $\boxed{\phi^{n} = (L+D)^{-1}\hat{T}\,\phi^{n-1} + (L+D)^{-1}S}$

~~HW~~

Write down algorithm for Gauss Seidel ...

Multigrid.

In order to ~~g.~~ do multigrid, lets look at ~~[illegible]~~ ~~[illegible]~~ ~~[illegible]~~. error & residual.

Equation $A\phi = S$ ← when we have soln.

But if our solution method is iterative, the soln $\phi \rightarrow \tilde{\phi}$ which is ~~just~~ ~~[illegible]~~ not necessarily the soln.

So call $A\tilde{\phi} = S$. ← Approximation after iteration.

Algebraic error → $e = \phi - \tilde{\phi}$

Residual: $r = S - A\tilde{\phi}$

Scanned with CamScanner

Since we ~~to~~ are dealing with iterative methods, residual changes upon iteration.

Moreover, by definition of error

$$e = \phi - \tilde{\Phi}$$
$$\Rightarrow \quad \phi = \tilde{\Phi} + e$$

- So if we have some handle on e ... some way to approximate e a get e, we could get to ϕ faster.

- If we have exact e, we would get sol^n in one iteration.

- But e is usually unknown... hold this thought.

- We will try to approximate error at every step & iterate using things we know.

we know $e = \phi - \tilde{\Phi} \Rightarrow \tilde{\Phi} = \phi - e$

Also $r = S - A\tilde{\Phi} \Rightarrow r = S - A(\phi - e)$

$$\Rightarrow r = S - A\phi + Ae$$

$S - A\phi = 0$ because given eqn. & ϕ is sol^n.

$$\Rightarrow \boxed{r = Ae.}$$

$$\Rightarrow e = A^{-1} r$$

Scanned with CamScanner

- ~~Again~~ $A^{-1} r$ returns exact error but requires inversion of A^{-1} so it basically wants us to solve problem before solving the problem to get sol^n.

- So then we choose an approximation to A^{-1} at each iterate & estimate e at each iterate and correct $\tilde{\phi}$

- This is equivalent to preconditioning

⇒ ~~we~~ In practice we do

General synthetic acceleration procedure

1. One iteration of iteration method. find $\tilde{\phi}^{m+1/2}$
2. find residual $r^{m+1} = S - A\tilde{\phi}^{m+1/2}$
3. find $\tilde{e}^{m+1} \approx \hat{A}^{-1} r^{m+1}$
4. $\tilde{\phi}^{m+1} = \tilde{\phi}^{m+1/2} + \tilde{e}^{m+1}$
5. Check for convergence if not converged go back to 1.

Scanned with CamScanner

Multigrid method builds on this error correction idea.

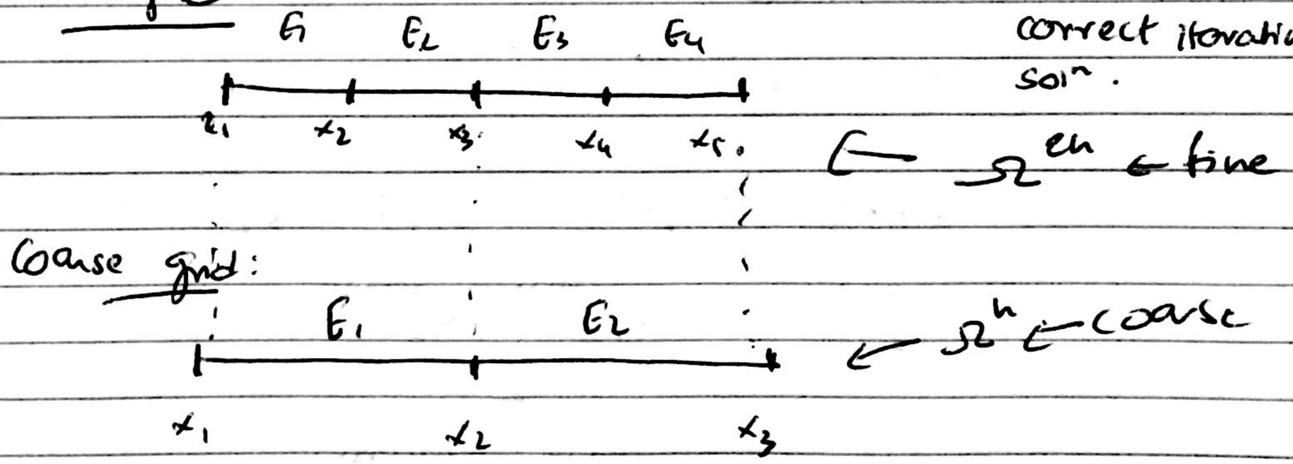
Our approximation to A^{-1} here comes from a coarse mesh.

Multigrid method - (2 grids, V-cycle)

We consider 2 grids: fine grid ← where we want soln

coarse grid ← where we find error to correct iterative soln.

fine grid:



method: ① Do one iteration (GJ, GS, ...)

find residual after one iterate. $r^n = S - A\tilde{\phi}^n$

② Transfer residual onto coarse grid (relaxation) (otherwise interpolate)

There will be coincident ~~points~~ nodes.

r_{fine} Just transfer residual from fine mesh there

so $r_1^{fine} = r_1^{coarse}$; $r_3^{fine} = r_2^{coarse}$; $r_5^{fine} = r_3^{coarse}$

Scanned with CamScanner

(3) Now, get e at those points $(x_1^{coarse}, x_2^{coarse}, x_3^{coarse})$

by doing $e^n = \hat{A}^{-1} r^n$

where $\hat{A}$ is the discretization matrix

~~of~~ we get for coarse grid.

(4) Once we have e on coarse grid, we transfer it back to the fine grid. (prolongation)

$$e_1^{fine} = e_1^{coarse}$$

$$e_3^{fine} = e_2^{coarse}$$

$$e_5^{fine} = e_3^{coarse}$$

We ~~always~~ interpolate to get e_2^{fine} & e_4^{fine}

$$e_2^{fine} = \frac{e_1^{coarse} + e_2^{coarse}}{2}$$

$$e_4^{fine} = \frac{e_2^{coarse} + e_3^{coarse}}{2}$$

Obviously this can be generalized to any # of elements.

Moreover relaxation & prolongation methods can be more advanced.

Scanned with CamScanner

5. Once we have $\hat{e}_1^{fine} \ldots e_5^{fine}$, we correct $\hat{\phi}^n$

where $\hat{\phi}_i^{n+1} = \hat{\phi}_i^n + e_i^n \qquad i = 1..5.$

6. Now check for convergence $\frac{\| \hat{\phi}^{n+1} - \hat{\phi}^n \|}{\| \phi^n \|} < \varepsilon$.

L_2 norm usually

ε = tolerance.

if ~~ow s tot~~ tolerance condition is met, stop otherwise go back to 1 and keep iterating.

Aside: This 2-grid iteration can be extended to an arbitrary # of grids (usually 5-6 grids taken) & different schemes used.