Numerical PDE

D.Y. Kwak

February 11, 2008

Chapter 2

Finite Element Methods

2.1 Model examples

One dim'l problem

$$-u'' = f \text{ on } (0,1), \text{ with B.C. } u(0) = u(1) = 0.$$

Multiply a test function $v \in H_0^1(I)$ and integrate

$$\begin{aligned}(-u'', v) &= -\int_0^1 u''v dx \\ &= -[u'v]_0^1 + \int_0^1 u'v' dx = \int fv dx.\end{aligned}$$

Thus we have

$$(u', v') = (f, v), \quad v \in V = H_0^1(I).$$

If the space $H_0^1(I)$ is replaced by a finite dimensional space $S_h(I)$ of piecewise continuous, linear functions on I with uniform spacing, then we get

$$Au_h = f_h$$

where a typical row is $[\cdots, 0, -1, 2, -1, 0 \cdots]$. FDM gives rise to similar matrix equation.

Two dimensional case - Poisson problem

$$\begin{aligned}-\Delta u &= f \text{ in } \Omega \\ u &= g \text{ on } \partial\Omega\end{aligned}$$

FDM

Assume uniform meshes with $h_x = h_y = h$. The stencil for x direction is $[-1, 2, -1]$. Likewise the stencil for the y direction is also $[-1, 2, -1]$. Hence the 2-D stencil is

$$\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}$$

FEM

Let $S_h^0(\Omega)$ be the space of continuous, piecewise linear on each element satisfying zero boundary condition. Let $u_h = \sum u_i \phi_i$ where ϕ_i is the tent shape basis function satisfying $\phi_i(x_j) = \delta_{ij}$. Then by multiplying ϕ_j and integrate by part to get

$$\int_\Omega \sum_i u_i \nabla\phi_i \cdot \nabla\phi_j \, dxdy = \int_\Omega f\phi_j \, dx, \text{ for each } j = 1, 2, \cdots$$

Writing $a(\phi_i, \phi_j) = \int_\Omega \nabla\phi_i \cdot \nabla\phi_j dxdy$ we get

$$\sum_i a(\phi_i, \phi_j) u_i = (f, \phi_j)$$

In matrix form, it is

$$Au = f, \quad A_{ij} = a(\phi_j, \phi_i)$$

A is called a 'stiffness' matrix.

$$\begin{array}{rcl} -\Delta u + u & = & f \text{ in } \Omega \\ \frac{\partial u}{\partial \mathbf{n}} & = & g \text{ on } \Gamma \end{array} \tag{2.1}$$

Note in this case u is unknown at the boundary. So $V = H^1(\Omega)$ not $H_0^1(\Omega)$. If $g = 0$, we have an insulated boundary.

$$(-\Delta u, v) + (u, v) = (\nabla u, \nabla v) + (u, v) - < g, v >_\Gamma = (f, v)$$

So the variational problem is: (V) Find $u \in H^1$ such that

$$a(u, v) = (f, v) + < g, v >, \quad v \in H^1$$

where $a(u, v) = (\nabla u, \nabla v) + (u, v)$. This is equivalent to (M) Find $u \in H^1$ such that $F(u) \le F(v)$ for all v where $F(v) = \frac{1}{2} a(v, v) - (f, v) - < g, v >$. Also, if $u \in C^2$ the solution of (M) and (V) are the solution of (2.1)

Proof. Let u be the solution of (V). Then

$$\begin{aligned} a(u,v) &= (-\Delta u, v) = \int_\Gamma \frac{\partial u}{\partial n} v + (u,v) = (f,v) + < g, v > \\ \int (-\Delta u + u - f)v &= \int_\Gamma (g - \frac{\partial u}{\partial n}) v ds, \quad v \in H^1 \end{aligned}$$

Restrict to $v \in H_0^1$. Then we get

$$-\Delta u + u - f = 0 \text{ in } \Omega$$

Hence we have

$$\int_\Gamma (g - \frac{\partial u}{\partial n}) v ds = 0, \quad v \in H^1$$

which proves $g = \frac{\partial u}{\partial n}$. The condition $\frac{\partial u}{\partial n} = g$ is called the natural boundary condition. (Look at (V), we did not impose any condition, but we got B.C naturally from the variational formulation.) □

2.2 Remarks on Programming

We consider Neumann problem (2.1). Let $\mathcal{T}_h = \{K_\ell\}$be a triangulation of the domain Ω and let V_h be the space of continuous, piecewise linear functions.

$$A\xi = b, \quad a_{ij} = \sum_K a_{ij}^K, b_i = \sum_K b_i^K$$

$a_{ij}^K = \int_K (\nabla \phi_j \cdot \nabla \phi_i + \phi_j \phi_i)$, $b_i^K = \int_K f\phi_i + \int_{K \cap \Gamma} g\phi_i \, ds$.

(1) Input data : f, g, Ω and coefficients.

(2) Construction and representation of $\mathcal{T}_h$

(3) Computation of element stiffness matrix a^K and b^K

(4) Assembly of global stiffness matrix A, b

(5) Solution of $A\xi = b$

(6) Presentation of result.

Remark on (2): quasi uniform—essentially the same size. Or often desirable to vary the size of triangle—adaptive or successive refinement. Conforming: vertex should not lie in the interior of an edge.

Matrix assembly

We need to compute $a(\phi_i, \phi_j)$

Compute

$$a(\phi_i, \phi_j) = \int_\Omega \nabla\phi_i \cdot \nabla\phi_j \, dxdy = \sum_K \int_K \nabla\phi_i \cdot \nabla\phi_j \, dxdy$$

where the summation runs through the common support of ϕ_i and ϕ_j. We find this by computing the contribution of $a_K(\phi_i, \phi_j) := \int_K \nabla\phi_i \cdot \nabla\phi_j \, dxdy$ called the *element stiffness matrix*. Let us divide a unit square by 4×4 uniform meshes where each small rectangle is cut through a line of slope 1. Label all the vertex nodes $1, 2, 3, \cdots, 25$ lexicographically. Label the element as K_1/K_2, $K_3/K_4, \cdots, K_7/K_8, \cdots$

Let us list some notations:

- L number of elements
- M number of nodes including the Dirichlet boundary
- N number of nodes excluding the Dirichlet boundary(same as number of unknowns)
- Γ_0 the part of boundary where Dirichlet condition is imposed
- Γ_1 the part of boundary where Neumann condition is imposed
- J_0 the index set where Dirichlet condition is imposed
- J_1 the index set where Neumann condition is imposed

Compute element stiffness matrix for each triangle, and add all the contribution to three vertices as K runs through all element. If $i = j$, K runs through all element having the node i as a vertex. If $i \neq j$, K runs through all element having $\bar{ij}$ as an edge. In this way, we assemble the global matrix A all together(not for each entry)

Consider $K = K_{12}$. Its vertices are $7, 8$ and 13. For the element matrix we need to compute $a_K(\phi_i, \phi_j)$ for $i, j = 7, 8, 13$.

$$\begin{aligned} a_K(\phi_7, \phi_7) &= \int_K (-\frac{1}{h}, 0) \cdot (-\frac{1}{h}, 0) = \frac{1}{2} \\ a_K(\phi_7, \phi_8) &= \int_K (-\frac{1}{h}, 0) \cdot (-\frac{1}{h}, -\frac{1}{h}) = -\frac{1}{2} \\ a_K(\phi_7, \phi_{13}) &= \int_K (-\frac{1}{h}, 0) \cdot (0, \frac{1}{h}) = 0 \end{aligned}$$

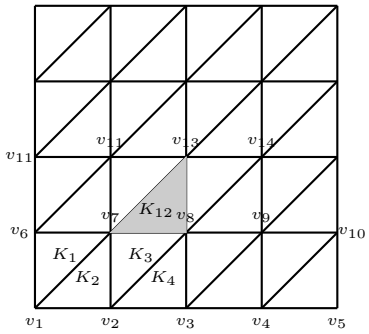

Figure 2.1: Label of elements and vertices

$$\begin{aligned} a_K(\phi_8,\phi_8) &= \int_K (\frac{1}{h},-\frac{1}{h})\cdot(\frac{1}{h},-\frac{1}{h}) = 1 \\ a_K(\phi_8,\phi_{13}) &= \int_K (\frac{1}{h},-\frac{1}{h})\cdot(0,\frac{1}{h}) = -\frac{1}{2} \\ a_K(\phi_{13},\phi_{13}) &= \int_K (0,\frac{1}{h})\cdot(0,\frac{1}{h}) = \frac{1}{2} \end{aligned}$$

The element stiffness matrix $A_{K_{12}}$ (corresponds to the vertices 7,8 and 13) is

$$\begin{bmatrix} \frac{1}{2}, & -\frac{1}{2}, & 0 \\ -\frac{1}{2}, & 1, & -\frac{1}{2} \\ 0, & -\frac{1}{2} & ,\frac{1}{2} \end{bmatrix} = a^{12}_{\alpha,\beta}$$

Generate element matrices for all element $K = 1, 2\cdots$, add its contribution to all pair of vertices (i, j). Let L be the number of elements and let T be the $3 \times L$ matrix whose ℓ-th column denotes the three numbers of vertices of ℓ-th element. For example, $T(\cdot, 12) = [7, 8, 13]^t$. Then the assembly of global stiffness matrix is as follows: Let $A(L, L), b(L)$ be arrays.

Set $A(i, j) = 0, b(i) = 0, i, j = 1, \cdots, L$.
For each $\ell = 1, 2, \cdots, L$
$A(T(\alpha, \ell), T(\beta, \ell)) += a^{\ell}_{\beta\alpha} \quad \alpha, \beta = 1, 2, 3$
$b^{\ell}(T(\alpha, \ell)) += (b^{\ell}_{\alpha}), \alpha = 1, 2, 3.$

Comment on solving algebraic equation.

(1) Use banded storage

(2) Iterative method or direct method ?

(3) Use as many modules as possible.

(4) Input appropriate data. $(0 < c_0 \le p(x,y) < c_1)$

(5) Check the error by discrete L^2, H^1 inner product.

Some notations:

L: number of triangles(elements)

n: number of nodes

K_ℓ, $\ell = 1, \cdots, L$: the elements

Z: $2 \times n$ matrix, $Z(j,i), j = 1,2$ coordinates of node i.

T: $3 \times L$ matrix, $T(j,\ell), j = 1,2,3$ denotes the global node numbering of ℓ-th triangle

A triangulation may be represented by two matrices such as $2 \times n$ matrix Z, and $3 \times L$ matrix T: $Z(j,i), j = 1,2$ represents the (x,y)-coordinate of node N_i, $i = 1, \cdots, n$, while $T(j,\ell), j = 1,2,3$ denote the vertex numbering of the ℓ-th triangle. Node ordering is important if we want to use Gaussian elimination(For instance, we intend to store A as a banded matrix, hopefully with small band)

(3) Computation of element stiffness matrix.

Let $K_\ell \in \mathcal{T}_h$. Then $T(\alpha,\ell)$, $\alpha = 1,2,3$ are the number of vertices of K_ℓ. The x_i-coordinates of vertices are $Z(i, T(\alpha,\ell)), i = 1,2$

$$a^\ell_{\alpha,\beta} = \int_{K_\ell} (\nabla\phi_\alpha \cdot \nabla\phi_\beta + \phi_\alpha\phi_\beta)dx$$

ϕ_α satisfies

$$\phi_\alpha(N_{T(\beta,\ell)}) = \delta_{\alpha,\beta}, \quad \alpha,\beta = 1,2,3$$

$$b^\ell_\alpha = \int_{K_\ell} f\phi_\alpha \, dx + \int_{\Gamma\cap K_\ell} g\phi_\alpha \, ds, \alpha = 1,2,3.$$

Store $A^\ell = (a^\ell_{\alpha\beta})$, $b^\ell = (b^\ell_\alpha)$ for all K_ℓ in a scratch space.

(4) Assembly of global stiffness matrix: Let $A(n,n), b(n)$ be arrays. Initially set $A(i,j) = 0, b(i) = 0, i,j = 1, \cdots, n$. For each $\ell = 1,2,\cdots,L$, compute $A^\ell = (a^\ell_{\alpha\beta})$ and $b^\ell = (b^\ell_\alpha)$ and set

$$\begin{aligned} A(T(\alpha,\ell), T(\beta,\ell))+ &= a^\ell_{\alpha\beta}, \quad \alpha,\beta = 1,2,3 \\ b(T(\alpha,\ell))+ &= b^\ell_\alpha, \alpha = 1,2,3. \end{aligned}$$

Remark 2.2.1. In practice we do not use the full array $A(n,n)$. Instead use either banded matrix if Gaussian elimination is used(How big is the band and what happens to the band during the elimination ?) or store only nonzero element if iterative methods are used.

For FEM software site; see //free.kaist.ac.kr www.netlib.org, http://gams.nist.gov ( ) http://www.ms.uky.edu/ skim/LectureNotes/

Treatment of nonhomogeneous Boundary Condition

- Dirichlet condition
- Neuman condition

Example 2.2.2.

$$\begin{aligned} -\nabla \cdot p\nabla u &= f, \text{ in } \Omega \\ u &= u_0 \text{ on } \Gamma_1 \\ \frac{\partial u}{\partial n} &= g \text{ on } \Gamma_2 \end{aligned} \tag{2.2}$$

Let $V_1 = \{v \in H^1(\Omega), v|_{\Gamma_1} = 0\}$. If $v \in V_1$

$$\begin{aligned} (-\nabla \cdot p\nabla u, v) &= -\int_\Gamma p\frac{\partial u}{\partial n} v ds + \int_\Omega p\nabla u \cdot \nabla v dxdy \\ &= -\int_{\Gamma_2} p\frac{\partial u}{\partial n} v ds + \int_\Omega p\nabla u \cdot \nabla v \, dxdy \end{aligned}$$

The variational formulation is: Find u satisfying the Dirichle condition such that

$$a(u,v) = \ell^*(v), \forall v \in V_1 \tag{2.3}$$

where $a(u,v) = (p\nabla u \cdot \nabla v)$ and $\ell^*(v) = (f,v) + < pg, v >_{\Gamma_2}$.

Remark 2.2.3. For a minimization formulation, set

$$F(v) = \frac{1}{2}a(v,v) - (f,v) - < pg, v >_{\Gamma_2}$$

Take derivative of $F(u + \epsilon v)$ w.r.t ϵ and set it to 0 at $\epsilon = 0$ to obtain (2.3).

Exercise 2.2.4. Show the solution of (2.3) satisfies $\frac{\partial u}{\partial n} = 0$.

Exercise 2.2.5. 1. Consider solving for $G \in H_0^1(I)$

$$(G', v') = v(x_i), \quad v \in H_0^1(I). \tag{2.4}$$

Let $G = ax$ on $(0, x_i)$ and $G = b(1-x)$ on $(x_i, 1)$.

Above equation is

$$\int_0^{x_i} av'dx + \int_{x_i}^1 -bv'dx = av(x_i) + bv(x_i) = v(x_i)$$

Thus $a + b = 1$ and by continuity $ax_i = b(1 - x_i)$. so $b = x_i$, $a = 1 - x_i$. Integrating (2.4) by part formally,

$$-(G'', v) = v(x_i).$$

This means

$$-G''(x) = \delta(x_i)$$

So we have a Green's function.

Choosing $v = u - u_h$, where u_h finite element approximation of u. we see

$$(u - u_h)(x_i) = (G', (u - u_h)') = 0.$$

So in one dimensional case, $u_h(x_i) = u(x_i)$ for each node. This is no longer true for higher dimensional case.

2. For a unit square Ω show that

$$\left(\int_\Gamma v^2 ds\right)^{1/2} \leq C\|v\|_{H^1}, \quad \forall v \in H^1(\Omega).$$

Proof. (Assume the existence of trace in L^2 space) Choose any point (x, y) on a vertical line $x = 1$, we see

$$\begin{aligned}
v(1,y) &= \int_x^1 v_t(t,y)dt + v(x,y) \\
v^2(1,y) &\leq 2\left(\int_x^1 v_t^2(t,y)dt\right) + 2v^2(x,y) \\
\int_0^1 v^2(1,y)dy &\leq 2\int_0^1\int_0^1 v_t^2(t,y)dtdy + 2\int_0^1 v^2(x,y)dy \\
\int_0^1 v^2(1,y)dy &\leq 2\int_0^1\int_0^1 v_x^2(t,y)dxdy + 2\int_0^1\int_0^1 v^2(x,y)dxdy
\end{aligned}$$

Other integrals are easily treated. □

3. Show that the solution of the variational problem :Find $u \in V_1$, such that

$$a(u, v) = (f, v), \forall v \in v \in V_1 \tag{2.5}$$

solves (2.2) with $u_0 = g = 0$.

2.3 Iterative method for SPD system

We consider solving the following equation

$$Ax = b \tag{2.6}$$

by an iterative method, where $A : \mathbb{R}^n \to \mathbb{R}^n$ is SPD. Let $\sigma(A) \subset [\lambda_0, \lambda_N]$ be the spectrum of A. Let R be another SPD matrix and define

$$M = I - R^{-1}A.$$

Let $\rho(M) = \max_{\sigma(M)} |\lambda|$ and suppose $\rho(M) < 1$. We may rewrite (2.6) as

$$x = Mx + R^{-1}b.$$

Define

$$x^{k+1} = Mx^k + \tilde{b}, \quad \tilde{b} = R^{-1}b,$$

or

$$x^{k+1} = x^k + R^{-1}(b - Ax^k),$$

where x^0 is arbitrary.

Set $e^k = x^k - x$. Then

$$e^k = Me^{k-1} = M^k e^0.$$

Now M is symmetric with respect to the inner product $[\cdot, \cdot]$ which is either $(A\cdot, \cdot)$ or $(R\cdot, \cdot)$. Hence for $\|\cdot\| = \|\cdot\|_A$ or $\|\cdot\| = \|\cdot\|_R$, we have

$$\|e^k\| = \|M^k e^0\| \le \rho(M)^k \|e^0\| \to 0.$$

Example 2.3.1 (Choice of R). (1) $R = \lambda_N I$ (Richardson)

(2) $R = D$ (diagonal of A)– Jacobi

(3) domain decomposition

(4) multigrid methods

2.3.1 Acceleration of convergence

For $x^0, \cdots, x^k$ defined above, we consider

$$y^m = \sum_{j=0}^{m} \nu_j x^j.$$

Then if $\sum_{j=0}^{m} \nu_j = 1$ we have

$$\tilde{e}^m = y^m - x = \sum_{j=0}^{m} \nu_j e^j = \sum_{j=0}^{m} \nu_j M^j e^0$$

Thus

$$\tilde{e}^m = P_m(M)e^0.$$

where P_m is a polynomial of degree m and $P_m(1) = 1$.(If $M = I, P_m(I) = I$). Let $M\phi_j = \lambda_j \phi_j$ be an orthonormal system of eigenvectors with respect to $[\cdot, \cdot]$. Then

$$\tilde{e}^m = \sum_{j=1}^{N} [\tilde{e}^m, \phi_j]\phi_j = \sum_{j=1}^{N} [P_m(M)e^0, \phi_j]\phi_j \tag{2.7}$$

$$= \sum_{j=1}^{N} [e^0, P_m(M)\phi_j]\phi_j = \sum_{j=1}^{N} [e^0, P_m(\lambda_j)\phi_j]\phi_j \tag{2.8}$$

$$= \sum_{j=1}^{N} P_m(\lambda_j)[e^0, \phi_j]\phi_j \tag{2.9}$$

Thus

$$\|\tilde{e}^m\|^2 = \sum_{j=1}^{N} P_m^2(\lambda_j)[e^0, \phi_j]^2 \leq \max_{\lambda_i \in \sigma(M)} P_m^2(\lambda_j)\|e^0\|^2 \tag{2.10}$$

For the case $P_m(M) = M^m$, we have $\tilde{e}^m = e^m$, and hence

$$\max_{\lambda_j \in \sigma(M)} P_m^2(\lambda_j) = \max_{\lambda_j \in \sigma(M)} \lambda_j^{2m} \leq \rho^{2m}$$

as before. We can do better by making a judicious choice of P_m. Let $\rho(M)$ denote the spectral radius of M and $\sigma(M)$ denote the set of all spectrum. Since

$$\|\tilde{e}^m\| \leq \sup_{|\lambda| \leq \rho(M)} |P_m(\lambda)| \|e^0\|.$$

We have to choose P_m such that $P_m(1) = 1$ and P_m minimizes $\sup_{|\lambda| \leq \rho(M)} |P_m(\lambda)|$. The solution of this is well known and is given by *Chebyshev* polynomials. The Chebyshev polynomials are defined by

$$C_m(x) = \begin{cases} \cos(m \cos^{-1} x), & |x| \leq 1 \\ \cosh(m \cosh^{-1} x), & |x| > 1. \end{cases}$$

Clearly,

$$C_{m+1}(x) = 2xC_m(x) - C_{m-1}(x).$$

Since $C_0(x) = 1,\ C_1(x) = x$, C_m is a polynomial of degree m. Define

$$P_m(x) = C_m(\frac{x}{\rho})/C_m(\frac{1}{\rho})$$

$\rho = \rho(M) < 1$. This is the solution of the minimization problem.

Let $\tilde{e}^m = P^m(M)e^0$. Then

$$\|\tilde{e}^m\| \leq \sup_{|x|\leq\rho} |P_m(x)|\|e^0\| = [C_m(\frac{1}{\rho})]^{-1}\|e^0\|.$$

Let $\sigma = \cosh^{-1}(\frac{1}{\rho})$

$$C_m(\frac{1}{\rho}) = \frac{e^{m\sigma} + e^{-m\sigma}}{2} = e^{m\sigma}\left[\frac{1 + e^{-2m\sigma}}{2}\right]$$

also

$$\sigma = \ln\left(\frac{1}{\rho} + \sqrt{\frac{1}{\rho^2} - 1}\right).$$

Hence

$$C_m(\frac{1}{\rho})^{-1} = e^{-m\sigma}\left[\frac{2}{1 + e^{-2m\sigma}}\right] \leq 2e^{-m\sigma} \tag{2.11}$$

$$= 2\left(\frac{1}{\rho} + \sqrt{\frac{1}{\rho^2} - 1}\right)^{-m} = 2\left(\frac{\rho}{1 + \sqrt{1 - \rho^2}}\right)^m. \tag{2.12}$$

If ρ is near 1 this is better than ρ^m for large m.

Suppose $\rho = \frac{1-\epsilon}{1+\epsilon}$, with ϵ small. Then

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

This means that we have accelerated the convergence. The number of iteration to reduce the initial error by a factor of δ is

$$2\left(\frac{1 - \sqrt{\epsilon}}{1 + \sqrt{\epsilon}}\right)^n \leq \delta$$

from which we see $n \approx \ln\frac{\delta}{2}/\ln\left(\frac{1-\sqrt{\epsilon}}{1+\sqrt{\epsilon}}\right) \approx \ln\frac{\delta}{2}/\sqrt{\epsilon}$.

2.3.2 Computation of the iterates

$$\tilde{e}^m = P_m(M)e^0 \Rightarrow y^m = x + P_m(M)e^0.$$

We note the recurrence relation

$$C_{m+1}(\frac{M}{\rho}) = \frac{2M}{\rho}C_m(\frac{M}{\rho}) - C_{m-1}(\frac{M}{\rho})$$

$$C_{m+1}(\frac{1}{\rho})P_{m+1}(M) = \frac{2M}{\rho}C_m(\frac{1}{\rho})P_m(M) - C_{m-1}(\frac{1}{\rho})P_{m-1}(M)$$

Apply this to e^0, x then subtract to get

$$C_{m+1}(\frac{1}{\rho})\tilde{e}^{m+1} = \frac{2}{\rho}C_m(\frac{1}{\rho})M\tilde{e}^m - C_{m-1}(\frac{1}{\rho})\tilde{e}^{m-1}$$

Noting that $Mx = x - \tilde{b}$ we have

$$C_{m+1}(\frac{1}{\rho})y^{m+1} = \frac{2}{\rho}C_m(\frac{1}{\rho})(My^m + \tilde{b}) - C_{m-1}(\frac{1}{\rho})y^{m-1}$$

Thus

$$y^{m+1} = \frac{2}{\rho}\left(C_m(\frac{1}{\rho})/C_{m+1}(\frac{1}{\rho})\right)(My^m + \tilde{b}) - \left(C_{m-1}(\frac{1}{\rho})/C_{m+1}(\frac{1}{\rho})\right)y^{m-1}$$

Let

$$y^0 = x^0 \tag{2.13}$$

$$y^1 = Mx^0 + \tilde{b} = x^1, \tag{2.14}$$

and set

$$w_{m+1} = \frac{2C_m(\frac{1}{\rho})}{\rho C_{m+1}(\frac{1}{\rho})} = 1 + \frac{C_{m-1}(\frac{1}{\rho})}{C_{m+1}(\frac{1}{\rho})}, \quad w_1 = 2 \tag{2.15}$$

Then the iterative method becomes

$$y^{m+1} = w_{m+1}\{My^m + \tilde{b} - y^{m-1}\} + y^{m-1} \tag{2.16}$$

$$w_{m+1} = \frac{1}{1 - \frac{\rho^2}{4}w_m}. \tag{2.17}$$

Then the Chebyshev iterates are as simple to compute as the iterates $x^{m+1} = Mx^m + \tilde{b}$.

In order to apply this method we need to estimates ρ or λ_0, λ_N.(Depending on the choice of R) If $\sigma(A) \subset [\lambda_0, \lambda_N]$ then we may choose $R = \frac{\lambda_0+\lambda_N}{2} I$. Thus $M = I - \frac{2}{\lambda_0+\lambda_N} A$ and $\sigma(M) \subset [-\rho, \rho]$ where $\rho = \frac{\lambda_N-\lambda_0}{\lambda_N+\lambda_0} < 1$. Since $\rho = \frac{\kappa-1}{\kappa+1}$, the convergence of

$$x^{k+1} = Mx^k + \tilde{b}$$

is determined by

$$\|e^k\| = \|M^k e^0\| \leq \rho^k \|e^0\| = \left(\frac{\kappa-1}{\kappa+1}\right)^k \|e^0\|.$$

But in the Chevysheff iteration, $\tilde{e}^k$ satisfies

$$\|\tilde{e}^k\| \leq 2\left(\frac{\rho}{1+\sqrt{1-\rho^2}}\right)^k \|\tilde{e}^0\|$$

Hence we have the bound

$$\|\tilde{e}^k\| = 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \|\tilde{e}^0\|$$

which is much better for large k.

Solution by Minimization

Definition 2.3.2. A quadratic functional

$$f(x) = \frac{1}{2} x^T A x - b^T x + c.$$

where A is $n \times n$ symmetric positive definite.

Theorem 2.3.3. *x_0 is a stationary point of f if and only if $Ax_0 = b$.*

Proof. Necessity is clear. For sufficiency, using $Ax_0 = b$, we rewrite $f(x) = \frac{1}{2}(x - x_0)^T A(x - x_0) + \hat{c}$. Since A is diagonalizable, there exists an orthogonal matrix V such that $AV = \Lambda V$ or $V^T AV = \Lambda$, where $\Lambda = diag\{\lambda_1, \cdots, \lambda_k\}$ and $\lambda_i > 0$. Then if we let $z = V^T(x - x_0)$,

$$\hat{f}(z) = \frac{1}{2} z^T \Lambda z + \hat{c}$$

$$= \frac{1}{2} \sum \lambda_i z_i^2 + \hat{c}.$$

Thus $z = 0, x = x_0$ is a stationary point.

2.3.3 Method of steepest descent

We would like to find the minimizer of $f(x) = \frac{1}{2}x^T Ax - b^T x + c$. Let x^0 be arbitrary initial guess. We try to find the next approximation in the form

$$x^{k+1} = x^k + \tau_k d^k \tag{2.18}$$

d^k is called search direction where τ_k is chosen to minimize, or reduce $f(x)$ on some interval near x^k in that direction. We need to choose d^k and τ_k.

Definition 2.3.4. If $f(x+\tau d) < f(x), \quad 0 < \tau \leq \tau_0$, then d is a descent direction for f at x.

Theorem 2.3.5. *If $f \in C^1$ and $\nabla f(x)^T \cdot d < 0$, then d is a descent direction for for f at x.*

Proof. $f(x + \tau d) = f(x) + \tau \nabla f(x) \cdot d + o(\tau)$. If τ is small enough, the second term dominates the third term.

Theorem 2.3.6. *$d = -\nabla f(x) = b - Ax$ is the direction of steepset descent.*

To determine the parameter τ_k, we see

$$f(x + \tau d) = \frac{1}{2}\tau^2 d^T Ad + \tau d^T \nabla f(x) + \hat{c} = \frac{1}{2}\tau^2 (Ad, d) + \tau(d, \nabla f(x)) + \hat{c}.$$

$\min_\tau f(x + \tau d)$ is obtained when $\frac{d}{d\tau} f(x + \tau d)|_{\tau=0} = \tau(Ad, d) + (d, \nabla f(x)) = 0$. Thus, $\tau = -\frac{(d, \nabla f(x))}{(Ad, d)}$ and given x^k, the next search direction is given by

$$d^{k+1} = b - Ax^{k+1} = b - A(x^k + \tau_k d^k) = d^k - \tau_k Ad^k.$$

Now the method of steepset desecnt is described as follows:

$$d^k = b - Ax^k \tag{2.19}$$
$$x^{k+1} = x^k + \tau_k d^k \tag{2.20}$$
$$\tau_k = \frac{(d^k, d^k)}{(Ad^k, d^k)} \tag{2.21}$$
$$d^{k+1} = d^k - \tau_k Ad^k. \tag{2.22}$$

Convergence analysis

We introduce a norm $(\cdot,\cdot)_A$ by $(x,y)_A = (Ax,y)$. When A is symmetric, positive definite, $(\cdot,\cdot)_A$ becomes a true norm (called energy norm) on $\mathbb{R}^n$.

Theorem 2.3.7. *We have*

$$\|x^k - x\|_A \leq \left(\frac{\kappa(A)-1}{\kappa(A)+1}\right)^k \|x^0 - x\|_A,$$

where $\kappa(A)$ is the spectral condition number of A. Further the number of iteration to reduce the error by a factor ϵ is

$$N \geq \kappa(A)\ln(1/\epsilon).$$

2.3.4 Conjugate gradient

In the beginning, we start as in method of steepest descent to find the minimizer of $f(x) = \frac{1}{2}x^TAx - b^Tx + c$ or the zero of $-\nabla f(x) = b - Ax = 0$. Let $x^0 = 0$. Then $d^0 = r^0 = b$, and we let

$$x^{k+1} = x^k + \alpha_k d^k \tag{2.23}$$

$$r^k = b - Ax^k. (\text{ this was } d^k \text{ in steepest descent method}) \tag{2.24}$$

We choose, as in the method of steepest descent, α_k so that $f(x^k + \alpha d^k)$ is minimized. Thus,

$$\alpha_k = (d^k, r^k)/(Ad^k, d^k) \tag{2.25}$$

It also makes the residual r^{k+1} to be orthogonal to the search direction d^k,

$$(d^k, r^{k+1}) = (d^k, b - Ax^{k+1}) = (d^k, b - Ax^k - \alpha_k Ad^k) = 0. \tag{2.26}$$

Now try a new search direction :

$$d^{k+1} = r^{k+1} - \beta_k d^k. \tag{2.27}$$

(If $\beta_k = 0$, it is steepest descent.) We choose β_k so that

$$(Ad^j, d^k) = 0, \quad j \leq k-1 \tag{2.28}$$

and we choose d^{k+1} so that

$$(Ad^k, d^{k+1}) = (Ar^{k+1} - \beta_k Ad^k, d^k) = 0. \tag{2.29}$$

Thus

$$\beta_k = (Ad^k, r^{k+1})/(Ad^k, d^k). \tag{2.30}$$

Let

$$V_k = \text{SPAN}\{b, Ab, \cdots, A^{k-1}b\} = \text{SPAN}\{d^0, d^1, \cdots, d^{k-1}\}.$$

Proposition 2.3.8. *We have*

$$(Ad^{k+1}, y) = 0, \quad \forall y \in V_{k+1} \quad \textit{and} \tag{2.31}$$

$$(Ad^j, r^{k+1}) = 0, \quad j \leq k-1. \tag{2.32}$$

i.e.,

$$d^{k+1} \perp V_{k+1} \textit{ with respect to } (A\cdot, \cdot) \tag{2.33}$$

$$r^{k+1} \perp V_k \textit{ with respect to } (A\cdot, \cdot) \tag{2.34}$$

Proof. Use induction. For $k = 0$, we see $(Ad^0, d^1) = (Ad^0, r^1 - \beta_0 d^0) = (Ad^0, r^1) - \beta_0(Ad^0, d^0) = 0$ by choice of β_0. The Hypothesis for (2.32) is empty. So both of the hypothesis holds for $k = 0$.

We shall prove (2.32) first. Consider

$$(Ad^j, r^{k+1}) = (Ad^j, r^k - \alpha_k Ad^k) = (Ad^j, r^k) - \alpha_k(Ad^j, Ad^k), \text{ for } j \leq k-1.$$

The first term is zero by induction and the second term is zero by induction since $Ad^j \in V_k$ for $j \leq k-1$. For (2.31) we see

$$(Ad^j, d^{k+1}) = (Ad^j, r^{k+1}) - \beta_k(Ad^j, d^k) = (Ad^j, r^{k+1}) = 0,$$

by induction again. Hence we proved $(Ad^j, d^{k+1}) = 0$ for $j \leq k-1$. This together with (2.29) shows (2.31). □

Remark 2.3.9. Note that since

$$(d^{k-1}, A(x - x^k)) = A(d^{k-1}, e^k) = 0$$

(2.26) means

$$e^k = x^k - x \perp V_k \text{ w.r.t } (A\cdot, \cdot) \tag{2.35}$$

And (2.28) means $d^k \perp V_k$ with respect to $(A\cdot, \cdot)$. Thus the new search direction is always orthogonal to the previous vectors(directions)

The algorithm is:
Let $x^0 = 0$, $d^0 = r^0 = b$, and

$$r^k = b - Ax^k \tag{2.36}$$

$$x^{k+1} = x^k + \alpha_k d^k, \quad \alpha_k = (d^k, r^k)/(Ad^k, d^k) \tag{2.37}$$

$$d^{k+1} = r^{k+1} - \beta_k d^k, \quad \beta_k = (Ad^k, r^{k+1})/(Ad^k, d^k). \tag{2.38}$$

Note that since $r^{k+1} = r^k - \alpha_k Ad^k$, only one evaluation of A is necessary and no need to estimate β_k.

2.3.5 Error analysis

By (2.35) and Cauchy Schwarz inequality,

$$(Ae^k, e^k) = (Ae^k, y^k - y + y - x) = (Ae^k, y - x),\ \forall y \in V_k \tag{2.39}$$

$$(Ae^k, e^k) \leq (A(x-y), x-y),\ \forall y \in V_k. \tag{2.40}$$

Since $y \in V_k$ is arbitrary, we can set

$$y = P_{k-1}(A)b,$$

for some polynomial $P_{k-1}(t)$ of degree $k-1$. Hence

$$\begin{aligned}(A(x-y), x-y) &= (A^{-1}(I - AP_{k-1}(A))b, (I - AP_{k-1}(A))b) \\ &\leq \|I - AP_{k-1}(A)\|_A^2 (Ax, x).\end{aligned}$$

Since P_{k-1} is arbitrary

$$(Ae^k, e^k) \leq \min_{P_{k-1}} \|I - AP_{k-1}(A)\|_A^2 (Ax, x).$$

Let $Q_k(t) = Q_k((1-\tau)/c)$(change of variable). Then $Q_k(t) = \hat{Q}(\tau)$ Then Q_k is a polynomial of degree k such that $Q_k(0) = 1$ and $\hat{Q}_k$ is a polynomial of degree k such that $\hat{Q}_k(1) = 1$. If $t \in \sigma(A) \subset [\lambda_0, \lambda_N]$, choose $c = \frac{2}{\lambda_0 + \lambda_N}$. Then $\tau \in [-\rho, \rho]$, where $\rho = \frac{\lambda_N - \lambda_0}{\lambda_0 + \lambda_N} = \frac{\kappa - 1}{\kappa + 1}$, $\kappa = \frac{\lambda_N}{\lambda_0}$. So

$$\min_{P_k} \|I - AP_{k-1}(A)\|_A \tag{2.41}$$

$$= \min_{Q_k(0)=1} \|Q_k(A)\|_A \tag{2.42}$$

$$= \min_{Q_k(0)=1,\ \lambda \in \sigma(A)} \max |Q_k(\lambda)| \tag{2.43}$$

$$= \min_{\hat{Q}_k(1)=1} \max_{\lambda \in \sigma(I - cA)} |\hat{Q}_k(\lambda)| \tag{2.44}$$

$$\leq \min_{\hat{Q}_k(1)=1} \max_{\lambda \in [-\rho, \rho]} |\hat{Q}_k(\lambda)| \tag{2.45}$$

$$\leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k. \tag{2.46}$$

The minimum is obtained when $\hat{Q}_k(x) = C_k(\frac{x}{\rho})/C_k(\frac{1}{\rho})$ is Chebyshev polynomial.(By theory in Chebysev acceleration)

Remark 2.3.10. Conjugate gradient method ends in finite steps: If we choose P_N so that $1 - \lambda_j P_N(\lambda_j) = 0, \quad \forall \lambda_j \in \sigma(A)$, then $A(e^N, e^N) = 0$ and hence $x^N = x$.

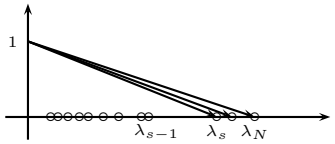

Figure 2.2: Eigenvalues accumulated near λ_0

Accumulated eigenvalues

If the eigenvalues are accumulated near λ_0 so that

$$\lambda_0 < \cdots < \lambda_{s-1} << \lambda_s < \cdots < \lambda_N$$

then one can get better estimate. In fact, typical matrix A from finite element methods satisfies $\sigma(A) \subset [Ch^2, C_1]$. So they are accumulated near λ_0

Let $Q(t) = Q_1(t)Q_2(t)$, where $Q_2(t) = \prod_{j=s}^{N} \left(\frac{\lambda_j - t}{\lambda_j}\right)$ is a polynomial of lower degree which is bounded by one at all eigenvalues $\lambda_0 < \cdots < \lambda_{s-1}$ and **vanishes** on $\lambda_{s+1} < \cdots < \lambda_N$. Hence

$$\max_{\sigma(A)} |Q| \leq \max_{\sigma(A)} |Q_1(t)| \max_{\sigma(A)} |Q_2| \leq \max_{[\lambda_0, \lambda_{s-1}]} |Q_1(t)|$$

Now take minimum w.r.t. all polynomials of degree $k - k_0 (k_0 = N - s + 1)$ such that $Q_1(1) = 1$. Hence again we choose Chebyshev polynomial to see

$$\min_{Q_1} \max_{[\lambda_0, \lambda_{s-1}]} |Q_1(t)| \leq \left(\frac{\sqrt{\kappa_0} - 1}{\sqrt{\kappa_0} + 1}\right)^{k-k_0}$$

where $\kappa_0 = \lambda_{s-1}/\lambda_0 << \kappa$. This is less than $\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k$ for large k.

2.3.6 Preconditioning

Consider $Ax = b$ which is equivalent to

$$R^{-1}Ax = R^{-1}b. \tag{2.47}$$

we introduce an inner product $[\cdot, \cdot]$ as either $(A\cdot, \cdot)$ or $(R\cdot, \cdot)$. Then the operator $R^{-1}A$ is symmetric with respect to $[\cdot, \cdot]$,i.e.

$$[R^{-1}Ax, y] = [x, R^{-1}Ay].$$

R^{-1} is called a preconditioner for A. Two properties of preconditioner is desirable:

(1) The action of R^{-1} on an arbitrary vector is in some sense "easy" to compute.

(2) Since A and R are both SPD, there exist $\tilde{\lambda}_0, \tilde{\lambda}_N$ such that

$$\tilde{\lambda}_0(Rx, x) \leq (Ax, x) \leq \tilde{\lambda}_N(Rx, x).$$

The condition number of $R^{-1}A = \tilde{\lambda}_N/\tilde{\lambda}_0$ should be smaller than that of A.

With $M = I - R^{-1}A$ we can change (2.47) into the form

$$x^{k+1} = Mx^k + R^{-1}Ax, \tag{2.48}$$

Lemma 2.3.11. *Suppose $\sigma(M) \subset (-\rho_0, \rho_1), \rho = \max(\rho_0, \rho_1) < 1$. Then R^{-1} is a preconditioner for A with condition number $\kappa(R^{-1}A) = \frac{1+\rho_0}{1-\rho_1}$.*

Proof. Since

$$-\rho_0(Ry, y) \leq (RMy, y) \leq \rho_1(Ry, y)$$

Then

$$-\rho_1(Ry, y) \leq (Ay, y) - (Ry, y) \leq \rho_0(Ry, y)$$
$$(1 - \rho_1)(Ry, y) \leq (Ay, y) \leq (1 + \rho_0)(Ry, y)$$

□

Remark 2.3.12. Suppose $\sigma(A) \subset [\lambda_0, \lambda_N]$. Then Choose c so that $\sigma(I - cA) \subset [1 - c\lambda_N, 1 - c\lambda_0]$. Choose c so that the spectral radius of M becomes smallest. We choose c so that

$$1 - c\lambda_N = 1 - c\lambda_0.$$

This gives the choice $c = 2/(\lambda_N + \lambda_0)$. Typically, $\lambda_N \approx$ diagonal of $2A$. So damped Jacobi is good preconditioner.

2.3.7 Application to Conjugate Gradient Method

With $z^0 = R^{-1}r^0 = d^0 = \tilde{b} - R^{-1}Ax^0$, the algorithm is

$$x^{k+1} = x^k + \alpha_k d^k, \alpha_k = (d^k, r^k)/(Ad^k, d^k) \tag{2.49}$$
$$r^{k+1} = r^k - \alpha_k Ad^k \tag{2.50}$$
$$z^{k+1} = R^{-1}r^{k+1} \tag{2.51}$$
$$d^{k+1} = z^{k+1} - \beta_k d^k, \beta_k = [R^{-1}Ad^k, z^{k+1}]/[R^{-1}Ad^k, d^k] \tag{2.52}$$

With $[\cdot, \cdot] = (R\cdot, \cdot)$, we obtain a preconditioned conjugate gradient method:

Let $z^0 = R^{-1}r^0 = d^0 = \tilde{b} - R^{-1}Ax^0$

$$x^{k+1} = x^k + \alpha_k d^k, \alpha_k = (d^k, r^k)/(Ad^k, d^k) \tag{2.53}$$
$$r^{k+1} = r^k - \alpha_k Ad^k \tag{2.54}$$
$$z^{k+1} = R^{-1}r^{k+1} \tag{2.55}$$
$$d^{k+1} = z^{k+1} - \beta_k d^k, \beta_k = (Ad^k, z^{k+1})/(Ad^k, d^k) \tag{2.56}$$

Remark 2.3.13. (1) To determine α_k we do not use weighted inner product.

(2) It is easy to show $(d^k, r^k) = (r^k, r^k)$.

(3) Alternatively, α_k, β_k can be computed as

$$\begin{aligned} \alpha_k &= (z^k, r^k)/(Ad^k, d^k) \\ \beta_k &= -(r^{k+1}, z^{k+1})/(r^k, z^k) \end{aligned}$$