Linear algebra

torsdag 26. oktober 2017 10:50

1. <u>Direct methods of solution:</u> $\mathcal{O}(n^3)$

→ Gaussian elimination / LU

→ Cholesky

2. <u>Iterative methods:</u> the simplest strategy is to split the matrix in the form

$A = P - N$, P non-singular

then the iteration proceeds by solving

$$P x^{(k+1)} = N x^{(k)} + b$$

(for the solution of $Ax = b$)

we then hope that $x^{(k)} \longrightarrow x$, the solution of $Ax = b$

(NB: P must be cheaply invertible or we would be better off solving directly)

i) <u>Analysis of convergence:</u>

The standard manipulation is to write the iteration in the form

$$x^{(k+1)} = \underbrace{(I - P^{-1}A)}_{B} x^{(k)} + P^{-1} b.$$

define the error $e^{(k)} = x - x^{(k)}$,

then $e^{(k+1)} = B e^{(k)}$

as can be seen by writing

$$e^{(k+1)} = x - Bx^{(k)} - P^{-1}b$$

$$= (Bx + P^{-1}b) - Bx^{(k)} - P^{-1}b.$$

$$= B(x - x^{(k)}) = Be^{(k)}$$

using $x = Bx + P^{-1}b$

as $Bx = (I - P^{-1}A)x = x - P^{-1}b.$

We therefore have $e^{(k)} \to 0$ as $k \to \infty$ if the spectral radius $\rho(B) < 1$

<u>ii) Richardson iteration</u>:

The previous iterations can be written

$$x^{(k+1)} = x^{(k)} - P^{-1} r^{(k)},$$

where $r^{(k)} = b - Ax^{(k)} = Ae^{(k)}$ is the residual

more generally, we consider

$$x^{(k+1)} = x^{(k)} - \alpha P^{-1} r^{(k)},$$ where α is some parameter.

We find the following

a) the method converges iff

$$\frac{2\,\mathrm{Re}(\lambda_i)}{\alpha |\lambda_i|^2} > 1, \quad \lambda_i \text{ eigenvalues of } P^{-1}A$$

b) the optimal value of α is

$$\alpha_0 = \frac{2}{\lambda_1 + \lambda_n}, \qquad \left.\begin{array}{ll} \lambda_1 & \text{largest} \\ \lambda_n & \text{smallest} \end{array}\right\} \text{eigenvalue}$$

for which

$$\rho_0 = \frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n},$$

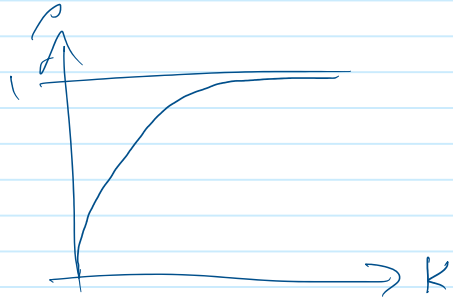
where ρ_0 is the spectral radius of the associated method.

c) Conditioning and convergence:

$$\frac{\frac{\lambda_1}{\lambda_n} - 1}{\frac{\lambda_1}{\lambda_n} + 1} = \frac{\frac{\lambda_1 - \lambda_n}{\lambda_n}}{\frac{\lambda_1 + \lambda_n}{\lambda_n}} = \frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}$$

i.e. where P, A are both symmetric, positive definite, we have

$$\rho_0 = \frac{K_2(P^{-1}A) - 1}{K_2(P^{-1}A) + 1}$$



where K_2 is the condition number.

d) Need for preconditioning: if $P = A$

$$K_2(A) \sim h^{-2}.$$

$$\frac{h^{-2} - 1}{h^{-2} + 1} = \frac{1 - h^2}{1 + h^2}$$

$$= (1-h^2)(1-h^2+h^3-\dots)$$

$$= 1 - 2h^2 + \dots$$

i.e. very slow convergence, even with optimal α.

e) Preconditioners: e.g., incomplete / sparse Cholesky, etc.

(i.e. approximations of a Cholesky factorization).

f) Classical methods: $P = D(A)$, diagonal part of A $\Rightarrow$ Jacobi

(or JOR w/ $\alpha \neq 1$)

$P = L(A)$, lower triangular part of A

$\Rightarrow$ Gauss-Seidel (SOR w/ $\alpha \neq 1$)

3. <u>Minimization methods</u>: Suppose A is symmetric, positive definite. We can then consider iterative schemes based on minimization of the quadratic form

$$\phi(y) = \frac{1}{2} y^T A y - y^T b$$

Indeed,

$$\nabla \phi = \frac{1}{2}(A^T + A) y - b = Ay - b$$

so $\nabla \phi = 0 \iff y$ solves equation

Moreover,

$$\phi(y) = \phi(x) + \tfrac{1}{2}(y-x)^T A (y-x)$$

where x solves the equation, so

$\phi(y) > \phi(x) \quad \forall y \neq x$, i.e. the solution minimizes ϕ.

So, iterative minimization schemes can be employed.

i) <u>Gradient descent</u>: here we move in the direction of steepest descent,

$$x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)},$$

where $d^{(k)} = \nabla\phi(x^{(k)})$. In fact, $\nabla\phi(x^{(k)}) = Ax^{(k)} - b = -r^{(k)}$, so this is really a method of the previous type. The line search gives

$$\alpha_k = \frac{r^{(k)T} r^{(k)}}{r^{(k)T} A r^{(k)}}$$

(found be writing

$$\phi(x^{(k+1)}) = \tfrac{1}{2}(x^k + \alpha r^{(k)})^T A (x^{(k)} + \alpha r^{(k)}) - (x^{(k)} + \alpha r^{(k)})^T b$$

and differentiating wrt α).

The convergence rate is roughly

$$\frac{K_2(A) - 1}{K_2(A) + 1}$$, in line with previous results.

typically very slow.

ii) <u>Conjugate gradient</u>: (a better method!)

Here we generate a sequence of search directions $p^{(1)}, \dots, p^{(k)}, \dots$

such that
$$(p^{(i)})^T A p^{(j)} = 0 \quad \forall i, j$$

Then we move $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$

where α_k can be found as before:

$$\alpha_k = -\frac{r_k^T p^{(k)}}{(p^{(k)})^T A p^{(k)}},$$ where $r_k = b - A x^{(k)}$ is the residual.

a) <u>Theorem</u>: a conjugate gradient method in $\mathbb{R}^n$ terminates in the exact solution after at most n steps.

The point is, $p^{(i)}$ must be linearly independent, as if $\sum a_i p^{(i)} = 0$

$$\sum a_i (p^{(i)})^T A p^{(j)} = a_j (p^{(j)})^T A p^{(j)} = 0$$

$\Rightarrow a_i = 0$ as A positive

definite

holds $\forall j \Rightarrow$ lin. indep.

therefore, we write

(*) $x - x_0 = \sum \sigma_k p^{(k)}$, aim to show

$\sigma_k = \alpha_k$. Now

$$\sigma_k = \frac{p^{(k)T} A (x - x_0)}{p^{(k)T} A p^{(k)}}$$

(multiply (*) on left by $p^{(k)T}A$ and use conjugacy)

Now

$$x_k = x_0 + \alpha_1 p^{(1)} + \dots + \alpha_{k-1} p^{(k-1)}$$

$$\Rightarrow \quad p^{(k)T} A (x_k - x_0) = 0$$

hence

$$p^{(k)T} A (x - x_0) = p^{(k)T} A (x - x_k) = - p^{(k)T} r_k,$$

i.e. $\sigma_k = \alpha_k$ and we're done.

b) <u>Conjugate gradient algorithm</u>:

Many possible choices (e.g. $p^{(k)}$ eigenbasis)

However, the following algorithm allows us to generate $p^{(k+1)}$ from $p^{(k)}$ alone. (i.e. do not require $p^{(i)}$ for $i < k$, saves

time and space).

Set $r^{(0)} = b - Ax^{(0)}$, and $p^{(0)} = r^{(0)}$

then

- $\alpha_k = \dfrac{p^{(k)T} r^{(k)}}{p^{(k)T} A p^{(k)}} = \dfrac{\| r_k \|^2}{p^{(k)T} A p^{(k)}}$

- $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$

- $r^{(k+1)} = r^{(k)} - \alpha_k A p^{(k)}$

- $\beta_k = \dfrac{(Ap^{(k)})^T r^{(k+1)}}{(Ap^{(k)})^T p^{(k)}} = \dfrac{\| r_{k+1} \|^2}{\| r_k \|^2}$

- $p^{(k+1)} = r^{(k+1)} - \beta_k p^{(k)}$

c) <u>Rate of convergence</u>: in practice, we do not use CG as an exact solver, but terminate before we have performed n steps. We can show

$$\| e^{(k)} \|_A \leq 2 \left(\frac{\sqrt{K_2(A)} - 1}{\sqrt{K_2(A)} + 1} \right)^k \| e^{(0)} \|_A$$

where $\| v \|_A = x^T A x$ is the energy norm.

Note the improvement $K_2(A) \to \sqrt{K_2(A)}$ as before.

d) <u>Preconditioned CG</u>: the CG

d) <u>Preconditioned CG</u>: the CG method may also be performed with a preconditioner, resulting in convergence rates with $\sqrt{\kappa_2(P^{-1}A)}$

Here the scheme is:

$$z^{(0)} = P^{-1} r^{(0)}, \qquad p^{(0)} = z^{(0)}$$

$$\alpha_k = \frac{p^{(k)T} r^{(k)}}{p^{(k)T} A p^{(k)}}$$

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$$

$$r^{(k+1)} = r^{(k)} - \alpha_k A p^{(k)}$$

$$P z^{(k+1)} = r^{(k+1)}$$ (i.e. solve for $z^{(k+1)}$)

$$\beta_k = \frac{(Ap^{(k)})^T z^{(k+1)}}{(Ap^{(k)})^T p^{(k)}}$$

$$p^{(k+1)} = z^{(k+1)} - \beta_k p^{(k)}$$

(error estimates come from showing this is equivalent to CG on

$$(P^{-1/2} A P^{1/2}) y = P^{-1/2} b,$$

$$y = P^{1/2} x$$).

Next:

→ Minimization schemes for non-symmetric matrices

→ Multigrid methods.