Computational methods-Lecture 9

Iterative Methods for linear systems

1 Perturbation in b and A

In practice, there are always errors in determining A and b. We would like to know how these errors affect the computed solutions.

Theorem 1. *If* $\|\delta A\| < \frac{1}{\|A^{-1}\|}$, *then*

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\kappa(A)}{1-\kappa(A)\|\delta A\|/\|A\|}\left(\frac{\|\delta b\|}{\|b\|}+\frac{\|\delta A\|}{\|A\|}\right)$$

Proof. We have

$$(A+\delta A)(x+\delta x) = b+\delta b.$$

Using $Ax = b$, one has

$$(A+\delta A)\delta x = \delta b - \delta A x.$$

Hence,

$$\begin{aligned}\|\delta x\| &\leq \|A^{-1}(I+A^{-1}\delta A)^{-1}(\delta b-\delta A x)\| \\ &\leq \|A^{-1}\|\|(I+A^{-1}\delta A)^{-1}\|(\|\delta b\|+\|\delta A\|\|x\|) \\ &\leq \frac{\|A^{-1}\|}{1-\|A^{-1}\delta A\|}(\|\delta b\|+\|\delta A\|\|x\|)\end{aligned}$$

Hence,

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\|A^{-1}\|}{1-\|A^{-1}\|\|\delta A\|}\left(\frac{\|\delta b\|}{\|x\|}+\|\delta A\|\right)$$

Moreover, since

$$\frac{1}{\|x\|} \leq \|A\|\frac{1}{\|b\|},$$

one has

$$\begin{aligned}\frac{\|\delta x\|}{\|x\|} &\leq \frac{\|A^{-1}\|}{1-\|A^{-1}\|\|\delta A\|}\left(\frac{\|A\|\|\delta b\|}{\|b\|}+\|\delta A\|\right) \\ &= \frac{\kappa}{1-\kappa\|\delta A\|/\|A\|}\left(\frac{\|\delta b\|}{\|b\|}+\frac{\|\delta A\|}{\|A\|}\right)\end{aligned}$$

□

If $\delta A = 0$, this reduces to what we have seen $\kappa \frac{\|\delta b\|}{\|b\|}$. From this bound, it is clear that if the condition number is big, the error bound is big and potentially, the relative error can be very large.

Example Consider the *Hilbert matrix* we have seen in the least square chapter.

2 Iterative methods

When the matrix is of large size and is sparse, it might be better to use iterative methods to solve the linear systems. We will look at several classical iterative methods, and then move to conjugate gradient descent.

The idea is similar as we do for finding roots of the nonlinear scalar equations. We rewrite

$$Ax = b$$

into

$$x = Bx + f.$$

Then, we construct sequences

$$x^{(k+1)} = Bx^{(k)} + f.$$

We hope that $\{x^{(k)}\}$ converges to some x^*.

It is clear that the error $\varepsilon^{(k)}$ satisfies

$$\varepsilon^{(k+1)} = B\varepsilon^{(k)} = B^k \varepsilon^{(0)}.$$

Hence, we need $B^k \to 0$ in some sense.

In general, write

$$A = M - N$$

such that

$$Mx = Nx + b \Rightarrow x = M^{-1}Nx + M^{-1}b.$$

3 The Jacobi iterative method

The idea is to look at the equation

$$\sum_j a_{ij} x_j = b_i.$$

Then, one moves the a_{ii} term to left:

$$x_i = (b_i - \sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^{n} a_{ij}x_j)/a_{ii}.$$

With this, we obtain the iterative method:

$$x_i^{(k+1)} = (b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)})/a_{ii} = (b_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j^{(k)})/a_{ii}.$$

This is the Jacobi iterative method.

In matrix form, this is equivalent to taking

$$M = D = \text{diag}(a_{ii}).$$

We decompose

$$A = D - L - U,$$

where L, U are lower and upper triangular matrices with diagonal elements being zero. In the decomposition

$$N = D - A = L + U,$$

and then

$$B = M^{-1}N = D^{-1}(L + U) =: J.$$

Then, the iterative method is given by

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

4 Gauss-Seidel iterative method

Consider the Jacobi iteration

$$x_i^{(k+1)} = (b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)})/a_{ii} = (b_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j^{(k)})/a_{ii}.$$

We solve this from $i = 1$ to n. However, as can be seen, if we solve $x_2^{(k+1)}$, x_1 has been updated already. Natural question: why do we have to use the old $x_1^{(k)}$? Maybe, using the newest data can speed up the convergence. This idea in fact leads to the Gauss-Seidel iterative method. This has been used in the so-called fast sweeping method for Eikonal equations.

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) / a_{ii}.$$

This corresponds to

$$Dx^{(k+1)} = Lx^{(k+1)} + Ux^{(k)} + b.$$

Hence, $M = D - L$ which is the lower triangular part of A. $N = M - A = U$.

Note that, the Gauss-Seidel iteration is convenient in computer because we can update the computed values in place at the location of the variable.

```
for i=1 to n
    x_i ← (b_i − Σ_{j=1}^{i−1} a_ij x_j − Σ_{j=i+1}^{n} a_ij x_j)/a_ii.
end
```

5 Successive Over-Relaxation

Consider the vector has been updated to $(x_1^{(k+1)}, \cdots, x_{i-1}^{(k+1)}, x_i^{(k)}, \cdots, x_n^{(k)})$. The residual for the ith component is

$$r_{ii}^{(k+1)} = b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} - a_{ii} x_i^{(k)}.$$

The Gauss-Seidel iteration chooses $x_i^{(k+1)}$ such that

$$a_{ii} x_i^{(k+1)} = a_{ii} x_i^{(k)} + r_{ii}^{(k+1)}.$$

In other words, the Gauss-Seidel is kind of doing a greedy strategy: it minimizes the residual for the current component so that

$$b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} - a_{ii} x_i^{(k+1)} = 0.$$

This local optimial choice, however, cannot lead to the best convergence because the residual for other components may grow. Hence, the idea is to go in that "good" direction with some amount:

$$x_i^{(k+1)} = x_i^{(k)} + \Delta_i^{(k)},$$

instead of choosing $\Delta_i^{(k)} = \frac{r_{ii}^{(k+1)}}{a_{ii}}$, we do

$$x_i^{(k+1)} = x_i^{(k)} + \omega \frac{r_{ii}^{(k+1)}}{a_{ii}}.$$

If $\omega > 1$, this is called the over-relaxation method. If the Gauss-Seidel converges, doing this can speed up the convergence. The whole method is called the SOR method.

Clearly, the iteraction can be written as

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \omega[x_i^{(k)} + \frac{r_{ii}^{(k+1)}}{a_{ii}}].$$

Hence, SOR, is to obtain a linear combination of the Gauss-Seidel result and the old $x^{(k)}$.

Clearly, the method is equivalent to

$$(D - \omega L)x^{(k+1)} = [(1-\omega)D + \omega U]x^{(k)} + \omega b.$$

Hence,

$$M = \frac{1}{\omega}(D - \omega L),$$

for the iterative method.

6 Convergence of the iterative methods

We have the following important theorem for a matrix

Theorem 2. *For a square matrix B, $\lim_{k\to\infty} B^k = 0$ if and only if $\|B^k\| \to 0$ for any operator norm, if and only if the spectral radius satisfies*

$$\rho(B) < 1.$$

It is obvious that the first two are equivalent. For the last claim, one makes use of the Jordan decomposition.

Take the Jordan block and compute...

Remark 1. *In fact, from functional analysis, there is a more general result: for an operator from a Banach space into itself*

$$\rho(T) = \lim_{m\to\infty} \sqrt[m]{\|T^m\|}.$$

Hence, if $\rho(T) < 1$, the Neumann series $\sum_{m=0}^{\infty} T^m$ converges.

Theorem 3. *The iterative method $x^{(k+1)} = Bx^{(k)} + f$ converges for any initial vector $x^{(0)}$ and f, if and only if $\rho(B) < 1$.*

Proof. First of all, assume $\rho(B) < 1$. Consider the equation

$$(I - B)x^* = f.$$

Since $\rho(B) < 1$, the eigenvalues of $I - B$ are all nonzero. Hence, the matrix is invertible. Define the error

$$e^k = x^{(k)} - x^*.$$

Then,

$$e^k = Be^{k-1} = B^k e^0 \to 0.$$

For the reverse, consider a convergent sequence, then the limit z satisfies

$$z = Bz + f.$$

Since this holds for any f, then $(I - B)z = f$ has a solution for any f, so $I - B$ is invertible. In other words, for a given f, the limit is unique.

Hence, given f, z is determined and $(x^{(k+1)} - z) = B(x^{(k)} - z)$ converges to 0 for any $x^{(0)}$ (since the limit is unique for a given f). We must have then $\lim_{k\to\infty} B^k = 0$. □

There are many consequences of this theorem, which we do not prove

- If some operator norm $\|B\| < 1$, then the iterative method converges.
- If A is strictly diagonally dominate, then the Jacobi, and the Gauss-Seidel iterations converge. The SOR converges for $\omega \in (0, 1]$.
- The SOR method can converge only if $\omega \in (0, 2)$.
- If A is positive definite, then SOR converges for any $\omega \in (0, 2)$.