MSDM5004
Numerical Methods and Modeling in Science
Spring 2024

Lecture 6

Prof Yang Xiang
Hong Kong University of Science and Technology

3. Gauss-Seidel method

Iteration algorithm:

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

Already obtained

Linear system $A\mathbf{x} = \mathbf{b}$

write $A = D - L - U$ $\quad a_{ii} \neq 0$, for each $i = 1, 2, \ldots, n$.

$$(D - L)\mathbf{x}^{(k)} = U\mathbf{x}^{(k-1)} + \mathbf{b}$$

$$\mathbf{x}^{(k)} = (D - L)^{-1}U\mathbf{x}^{(k-1)} + (D - L)^{-1}\mathbf{b},$$

Form of iteration: $\mathbf{x}^{(k)} = T_g\mathbf{x}^{(k-1)} + \mathbf{c}_g$

Gauss-Seidel

This formulation based on D, L, U is only for analysis.
Use the formulation on the previous page in codes.

Example

Use the Gauss-Seidel iterative technique to find approximate solutions to

$$\begin{aligned}
10x_1 - x_2 + 2x_3 &= 6, \\
-x_1 + 11x_2 - x_3 + 3x_4 &= 25, \\
2x_1 - x_2 + 10x_3 - x_4 &= -11, \\
3x_2 - x_3 + 8x_4 &= 15
\end{aligned}$$

starting with $\mathbf{x} = (0, 0, 0, 0)^t$ and iterating until

$$\frac{\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_\infty}{\|\mathbf{x}^{(k)}\|_\infty} < 10^{-3}.$$

Solution

For the Gauss-Seidel method we write the system, for each $k = 1, 2, \ldots$ as

$$
\begin{aligned}
x_1^{(k)} &= \qquad\qquad \frac{1}{10}x_2^{(k-1)} - \frac{1}{5}x_3^{(k-1)} \qquad\qquad + \frac{3}{5},\\
x_2^{(k)} &= \frac{1}{11}x_1^{(k)} \qquad\qquad + \frac{1}{11}x_3^{(k-1)} - \frac{3}{11}x_4^{(k-1)} + \frac{25}{11},\\
x_3^{(k)} &= -\frac{1}{5}x_1^{(k)} + \frac{1}{10}x_2^{(k)} \qquad\qquad + \frac{1}{10}x_4^{(k-1)} - \frac{11}{10},\\
x_4^{(k)} &= \qquad\qquad - \frac{3}{8}x_2^{(k)} + \frac{1}{8}x_3^{(k)} \qquad\qquad + \frac{15}{8}.
\end{aligned}
$$

When $\mathbf{x}^{(0)} = (0, 0, 0, 0)^t$, we have $\mathbf{x}^{(1)} = (0.6000, 2.3272, -0.9873, 0.8789)^t$.

Subsequent iterations give the values in Table

k	0	1	2	3	4	5
$x_1^{(k)}$	0.0000	0.6000	1.030	1.0065	1.0009	1.0001
$x_2^{(k)}$	0.0000	2.3272	2.037	2.0036	2.0003	2.0000
$x_3^{(k)}$	0.0000	−0.9873	−1.014	−1.0025	−1.0003	−1.0000
$x_4^{(k)}$	0.0000	0.8789	0.9844	0.9983	0.9999	1.0000

Because

$$\frac{\|\mathbf{x}^{(5)} - \mathbf{x}^{(4)}\|_\infty}{\|\mathbf{x}^{(5)}\|_\infty} = \frac{0.0008}{2.000} = 4 \times 10^{-4}$$

$\mathbf{x}^{(5)}$ is accepted as a reasonable approximation to the solution.

4. SOR method

Successive Over-Relaxation

Idea

$$D\mathbf{x}^{(k)} = (1-\omega)D\mathbf{x}^{(k-1)} + \omega\big(L\mathbf{x}^{(k)} + U\mathbf{x}^{(k-1)} + \mathbf{b}\big)$$

for certain choices of positive ω

Linear combination of the result of Gauss-Seidel iteration

$$D\mathbf{x}^{(k)} = L\mathbf{x}^{(k)} + U\mathbf{x}^{(k-1)} + \mathbf{b}$$

and

$$D\mathbf{x}^{(k-1)}$$

$$D\mathbf{x}^{(k)} = (1-\omega)D\mathbf{x}^{(k-1)} + \omega\left(L\mathbf{x}^{(k)} + U\mathbf{x}^{(k-1)} + \mathbf{b}\right)$$

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

The SOR method

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

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

The SOR method can be written in the form

$$\mathbf{x}^{(k)} = T_\omega \mathbf{x}^{(k-1)} + \mathbf{c}_\omega$$

with

$$T_\omega = (D - \omega L)^{-1}[(1-\omega)D + \omega U]$$

$$\mathbf{c}_\omega = \omega(D - \omega L)^{-1}\mathbf{b}$$

Example

- The linear system $A\mathbf{x} = \mathbf{b}$ given by

$$\begin{aligned} 4x_1 + 3x_2 \qquad\quad &= 24 \\ 3x_1 + 4x_2 - \quad x_3 &= 30 \\ - \quad x_2 + 4x_3 &= -24 \end{aligned}$$

- Compare the iterations from the Gauss-Seidel method and the SOR method with $\omega = 1.25$ using $\mathbf{x}^{(0)} = (1, 1, 1)^t$ for both methods.

For each $k = 1, 2, \ldots,$ the equations for the Gauss-Seidel method are

$$
\begin{aligned}
x_1^{(k)} &= -0.75x_2^{(k-1)} + 6 \\
x_2^{(k)} &= -0.75x_1^{(k)} + 0.25x_3^{(k-1)} + 7.5 \\
x_3^{(k)} &= 0.25x_2^{(k)} - 6
\end{aligned}
$$

and the equations for the SOR method with $\omega = 1.25$ are

$$
\begin{aligned}
x_1^{(k)} &= -0.25x_1^{(k-1)} - 0.9375x_2^{(k-1)} + 7.5 \\
x_2^{(k)} &= -0.9375x_1^{(k)} - 0.25x_2^{(k-1)} + 0.3125x_3^{(k-1)} + 9.375 \\
x_3^{(k)} &= 0.3125x_2^{(k)} - 0.25x_3^{(k-1)} - 7.5
\end{aligned}
$$

Gauss-Seidel Iterations

k	0	1	2	3	$\cdots$	7
$x_1^{(k)}$	1	5.250000	3.1406250	3.0878906		3.0134110
$x_2^{(k)}$	1	3.812500	3.8828125	3.9267578		3.9888241
$x_3^{(k)}$	1	−5.046875	−5.0292969	−5.0183105		−5.0027940

SOR Iterations ($\omega = 1.25$)

k	0	1	2	3	$\cdots$	7
$x_1^{(k)}$	1	6.312500	2.6223145	3.1333027		3.0000498
$x_2^{(k)}$	1	3.5195313	3.9585266	4.0102646		4.0002586
$x_3^{(k)}$	1	−6.6501465	−4.6004238	−5.0966863		−5.0003486

<u>Remark</u>: The exact solution is $x=(3,4,-5)^t$.

For the iterates to be accurate to 7 decimal places,

- the Gauss-Seidel method requires 34 iterations,
- as opposed to 14 iterations for the SOR method with $\omega = 1.25$.

5. Convergence of the iterative methods

Theorem

For any $\mathbf{x}^{(0)} \in \mathbb{R}^n$, the sequence $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ defined by

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}, \quad \text{for each } k \geq 1,$$

converges to the unique solution of $\mathbf{x} = T\mathbf{x} + \mathbf{c}$ if and only if $\rho(T) < 1$.

(similar to fixed-point iteration)

The **spectral radius** $\rho(A)$ of a matrix A is defined by

$$\rho(A) = \max |\lambda|, \quad \text{where } \lambda \text{ is an eigenvalue of } A.$$

(For complex $\lambda = \alpha + \beta i$, we define $|\lambda| = (\alpha^2 + \beta^2)^{1/2}$.)

Diagonally dominant matrices

Definition

The $n \times n$ matrix A is said to be **diagonally dominant** when

$$|a_{ii}| \geq \sum_{\substack{j=1,\\ j\neq i}}^{n} |a_{ij}| \quad \text{holds for each } i = 1, 2, \cdots, n.$$

A diagonally dominant matrix is said to be **strictly diagonally dominant** when the inequality in (6.10) is strict for each n, that is, when

$$|a_{ii}| > \sum_{\substack{j=1,\\ j\neq i}}^{n} |a_{ij}| \quad \text{holds for each } i = 1, 2, \cdots, n.$$

Theorem A strictly diagonally dominant matrix A is nonsingular.

Convergence of the Jacobi and Gauss-Seidel methods

Theorem

If A is strictly diagonally dominant, then for any choice of $\mathbf{x}^{(0)}$, both the Jacobi and Gauss-Seidel methods give sequences $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ that converge to the unique solution of $A\mathbf{x} = \mathbf{b}$. ■

Convergence of the SOR method

Theorem (Kahan)

If $a_{ii} \neq 0$, for each $i = 1, 2, \ldots, n$, then $\rho(T_\omega) \geq |\omega - 1|$. This implies that the SOR method can converge only if $0 < \omega < 2$. ■

Theorem (Ostrowski-Reich)

If A is a positive definite matrix and $0 < \omega < 2$, then the SOR method converges for any choice of initial approximate vector $\mathbf{x}^{(0)}$. ■

Theorem

If A is positive definite and tridiagonal, then $\rho(T_g) = [\rho(T_j)]^2 < 1$, and the optimal choice of ω for the SOR method is

$$\omega = \frac{2}{1 + \sqrt{1 - [\rho(T_j)]^2}}.$$

With this choice of ω, we have $\rho(T_\omega) = \omega - 1$. ■

6. Other iterative methods

Linear system $A\mathbf{x} = \mathbf{b}$

Conjugate gradient (CG) method

Matrix *A* is symmetric and positive definite

The solution of the linear system is the minimizer of the quadratic function

$$g(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{b}^T\mathbf{x}$$

Large linear systems with symmetric, positive definite matrices are commonly solved using the CG method.

The set of vectors $\{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \cdots, \mathbf{v}^{(n)}\}$ are A-conjugate if

$$\langle \mathbf{v}^{(i)}, A\mathbf{v}^{(j)} \rangle = 0, \quad \text{if} \quad i \neq j.$$

The CG algorithm

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + t_k \mathbf{v}^{(k)}$$

$$t_k = \frac{\langle \mathbf{v}^{(k)}, \mathbf{b} - A\mathbf{x}^{(k-1)} \rangle}{\langle \mathbf{v}^{(k)}, A\mathbf{v}^{(k)} \rangle}$$

The set of vectors $\{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \cdots, \mathbf{v}^{(n)}\}$ are generated from the residual $\mathbf{r} = \mathbf{b} - A\mathbf{x}$: $\{\mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \cdots, \mathbf{r}^{(n)}\}$

$$\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}$$

Generalized minimal residuals (GMRES)

Linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$

Iterative method that minimize the residual in Krylov subspaces

$$\{\mathrm{b}, A\mathrm{b}, A^2\mathrm{b}, \cdots, A^{(k-1)}\mathbf{b}\}$$

Large linear systems with nonsymmetric matrices are commonly solved using the GMRES method.

Chapter 8

Numerical Differentiation

1. Numerical differentiation

Review: The derivative of the function f at x_0 is

$$f'(x_0) = \lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}.$$

An approximation of the derivative

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h}$$

where h is a small number.

Finite difference

Error analysis

$$g(x) = O(h(x)), \quad x \to x_0$$

$$\Longleftrightarrow |g(x)| \leq M|h(x)|, \text{ near } x_0,$$

for some constant M.

Using Taylor expansion

$$f(x_0 + h) = f(x_0) + f'(x_0)h + \frac{1}{2}f''(\xi)h^2,$$

where ξ is between x_0 and $x_0 + h$.

We have

$$f'(x_0) = \boxed{\frac{f(x_0 + h) - f(x_0)}{h}} - \frac{h}{2}f''(\xi).$$

The error of this approximation is $\frac{1}{2}f''(\xi)h$, which is bounded by $\boxed{\frac{M}{2}h}$ with M being the bound of $f''(x)$ over the interval $[x_0, x_0 + h]$. The error is $O(h)$. First order method.

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h}$$

forward-difference formula if $h > 0$

backward-difference formula if $h < 0$.

An example

Use the forward-difference formula to approximate the derivative of $f(x) = \ln x$ at $x_0 = 1.8$ using $h = 0.1$, $h = 0.05$, and $h = 0.01$

Solution The forward-difference formula

$$f'(1.8) \approx \frac{f(1.8+h) - f(1.8)}{h}$$

h	$\dfrac{f(1.8+h) - f(1.8)}{h}$	$\dfrac{\lvert h\rvert}{2(1.8)^2}$ (bound of error)
0.1	0.5406722	0.0154321
0.05	0.5479795	0.0077160
0.01	0.5540180	0.0015432

Note: The exact value of $f'(1.8) = \dfrac{1}{1.8} = 0.55555\cdots$.

Central difference formula

$$f'(x_0) \approx \frac{1}{2h}[f(x_0+h) - f(x_0-h)]$$

Using Taylor expansion

$$f'(x_0) = \frac{1}{2h}[f(x_0+h) - f(x_0-h)] - \frac{h^2}{6} f^{(3)}(\xi_1),$$

where ξ_1 lies between $x_0 - h$ and $x_0 + h$.

The error is $O(h^2)$. Second order method.

Second Derivative

$$f''(x_0) \approx \frac{1}{h^2}[f(x_0 - h) - 2f(x_0) + f(x_0 + h)] \cdot$$

Central difference formula

Using Taylor expansion

$$f''(x_0) = \frac{1}{h^2}[f(x_0 - h) - 2f(x_0) + f(x_0 + h)] - \frac{h^2}{12} f^{(4)}(\xi),$$

for some ξ, where $x_0 - h < \xi < x_0 + h$.

The error is $O(h^2)$. Second order method.

Constructing finite difference schemes using Taylor expansion

e.g. Construct a finite difference scheme for the derivative of $f(x)$ at the point x_0, using values of $f(x_0)$, $f(x_0+h)$, $f(x_0+2h)$.

Using Taylor expansion

$$f(x_0+h) = f(x_0) + hf'(x_0) + \frac{h^2}{2}f''(x_0) + \frac{h^3}{3!}f'''(x_0) + \cdots$$

$$f(x_0+2h) = f(x_0) + 2hf'(x_0) + \frac{(2h)^2}{2}f''(x_0) + \frac{(2h)^3}{3!}f'''(x_0) + \cdots$$

Assume

$$f'(x_0) = a_0 f(x_0) + a_1 f(x_0 + h) + a_2 f(x_0 + 2h) + O(?)$$

Coefficients of $f(x_0)$: $$a_0 + a_1 + a_2 = 0. \tag{1}$$

Coefficients of $f'(x_0)$: $$a_1 h + a_2 2h = 1. \tag{2}$$

Coefficients of $f''(x_0)$: $$a_1 \frac{h^2}{2} + a_2 \frac{(2h)^2}{2} = 0. \tag{3}$$

$$f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2} f''(x_0) + \frac{h^3}{3!} f'''(x_0) + \cdots$$

$$f(x_0 + 2h) = f(x_0) + 2hf'(x_0) + \frac{(2h)^2}{2} f''(x_0) + \frac{(2h)^3}{3!} f'''(x_0) + \cdots$$

Solving this linear system,

$$a_0 = -\frac{3}{2h}, \quad a_1 = \frac{2}{h}, \quad a_2 = -\frac{1}{2h}.$$

The numerical scheme is

$$f'(x_0) \approx \frac{-3f(x_0) + 4f(x_0+h) - f(x_0+2h)}{2h}.$$

The error is $O(h^2)$.

Check order of a numerical scheme

Compute derivative of $f(x) = \ln x$ at $x_0 = 1.8$.

Numerical approximation $\phi_h(x_0) = \dfrac{f(x_0 + h) - f(x_0)}{h}$

$$\begin{aligned}\phi_h(x_0) &= f'(x_0) + \frac{h}{2} f''(\xi), \quad \xi \in [x_0, x_0 + h] \\ &= f'(x_0) + \frac{h}{2} f''(x_0) + O(h^2).\end{aligned}$$

$$f''(\xi) = f''(x_0) + f'''(\xi_1)(\xi - x_0) = f''(x_0) + O(h)$$

$$\phi_h = f'(x_0) + h \cdot \frac{1}{2} f''(x_0) + O(h^2)$$

$$\phi_{\frac{h}{2}} = f'(x_0) + \frac{h}{2} \cdot \frac{1}{2} f''(x_0) + O(h^2)$$

$$\phi_{\frac{h}{4}} = f'(x_0) + \frac{h}{4} \cdot \frac{1}{2} f''(x_0) + O(h^2)$$

$$\frac{\phi_h - \phi_{\frac{h}{2}}}{\phi_{\frac{h}{2}} - \phi_{\frac{h}{4}}} \approx \frac{\left(h - \frac{h}{2}\right) \frac{1}{2} f''(x_0)}{\left(\frac{h}{2} - \frac{h}{4}\right) \frac{1}{2} f''(x_0)} = 2^1$$

"1" is the order of accuracy

Matlab code

```
x0=1.8;
h=0.1;
n=10;
format long
for i=1:n
    phi(i)=(log(x0+h)-log(x0))/h;
    h=h/2;
end;
for i=1:(n-2)
    r=(phi(i)-phi(i+1))/(phi(i+1)-phi(i+2))
end;
phi(n)
```

```
r =1.946839686322930
r =1.972834761917758
r =1.986266023312467
r =1.993094509786362
r =1.996537545297094
r =1.998266333041434
r =1.999132550471774
r =1.999566142373304

ans =

  0.555525416917817
```

Differentiation via polynomial interpolation

suppose that $\{x_0, x_1, \ldots, x_n\}$ are $(n+1)$ distinct numbers in some interval I and that $f \in C^{n+1}(I)$.

$$f(x) \approx P_n(x)$$

Lagrange polynomial of degree *n*

$$P_n(x) = \sum_{k=0}^{n} f(x_k) L_k(x)$$

$$L_k(x) = \frac{(x - x_0)\cdots(x - x_{k-1})(x - x_{k+1})\cdots(x - x_n)}{(x_k - x_0)\cdots(x_k - x_{k-1})(x_k - x_{k+1})\cdots(x_k - x_n)}.$$

$$f'(x) \approx P_n'(x)$$

Polynomial interpolation with error is

$$f(x) = \sum_{k=0}^{n} f(x_k)L_k(x) + \frac{(x-x_0)\cdots(x-x_n)}{(n+1)!} f^{(n+1)}(\xi(x)),$$

for some $\xi(x)$ in I, where $L_k(x)$ denotes the kth Lagrange coefficient polynomial for f at $x_0, x_1, \ldots, x_n$.

Differentiating this expression gives

$$f'(x) = \sum_{k=0}^{n} f(x_k)L_k'(x) + D_x\left[\frac{(x-x_0)\cdots(x-x_n)}{(n+1!)}\right] f^{(n+1)}(\xi(x))$$

$$+ \frac{(x-x_0)\cdots(x-x_n)}{(n+1)!} D_x[f^{(n+1)}(\xi(x))].$$

$$\boxed{D_x \Longleftrightarrow \frac{d}{dx}}$$

When x is one of the numbers x_j,

$$f'(x_j) = \sum_{k=0}^{n} f(x_k)L_k'(x_j) + \frac{f^{(n+1)}(\xi(x_j))}{(n+1)!}\prod_{\substack{k=0\\k\neq j}}^{n}(x_j - x_k),$$

(*n* + 1)-point formula to approximate $f'(x_j)$

All the formulas shown previously can also be obtained using this approach.