

Chapter 4

Iterative methods for solving linear system of equations

The Gaussian elimination method for solving $Ax = b$ is quite efficient if the size of A is small to medium (in reference the available computers) and dense matrices (most of entries of the matrix are non-zero numbers). But for several reasons, sometimes an iterative method may be more efficient

- For sparse matrices, the Gaussian elimination method may destroy the structure of the matrix and cause '*fill-in*'s, see for example,

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 1 & 0 & 0 & 0 \\ 4 & 0 & 0 & 1 & 0 & 0 \\ 5 & 0 & 0 & 0 & 1 & 0 \\ 6 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \implies \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & -1 & -2 & -2 & -2 & -2 \\ 0 & -3 & -2 & -3 & -3 & -3 \\ 0 & -4 & -4 & -3 & -4 & -4 \\ 0 & -5 & -5 & -5 & -4 & -5 \\ 0 & -6 & -6 & -6 & -6 & -5 \end{bmatrix}$$

Obviously, the case discussed above can be generalized to a general n by n matrix with the same structure.

- Large sparse matrices for which we may not be able to store all the entries of the matrix. Below we show an example in two dimensions.

4.0.5 The central finite difference method with five point stencil for Poisson equation.

Consider the Poisson equation

$$u_{xx} + u_{yy} = f(x, y), \quad (x, y) \in \Omega = (a, b) \times (c, d), \quad (4.0.1)$$

$$u(x, y)|_{\partial\Omega} = u_0(x, y), \quad \text{Dirichlet BC.} \quad (4.0.2)$$

If $f \in L^2(\Omega)$, then the solution exists and it is unique. Analytic solution is rarely available. Now we discuss how to use the finite difference equation to solve the Poisson equation.

- Step 1: Generate a grid. A uniform Cartesian grid can be used:

$$x_i = a + ih_x, \quad i = 0, 1, 2, \dots, m, \quad h_x = \frac{b-a}{m}, \quad (4.0.3)$$

$$y_j = c + jh_y, \quad j = 0, 1, 2, \dots, n, \quad h_y = \frac{d-c}{n}. \quad (4.0.4)$$

We want to find an approximate solution U_{ij} to the exact solution at all the grid points (x_i, y_j) where $u(x_i, y_j)$ is unknown. So there are $(m-1)(n-1)$ unknown for Dirichlet boundary condition.

- Step 2: Substitute the partial derivatives with a finite difference formula in terms of the function values at grid points to get.

$$\begin{aligned} & \frac{u(x_{i-1}, y_j) - 2u(x_i, y_j) + u(x_{i+1}, y_j))}{(h_x)^2} + \frac{u(x_i, y_{j-1}) - 2u(x_i, y_j) + u(x_i, y_{j+1}))}{(h_y)^2} \\ & = f_{ij} + T_{ij}, \quad i = 1, \dots, m-1, \quad j = 1, \dots, n-1, \end{aligned}$$

where $f_{ij} = f(x_i, y_j)$. The local truncation error satisfies

$$T_{ij} \sim \frac{(h_x)^2}{12} \frac{\partial^4 u}{\partial x^4} + \frac{(h_y)^2}{12} \frac{\partial^4 u}{\partial y^4}. \quad (4.0.5)$$

Define

$$h = \max\{h_x, h_y\} \quad (4.0.6)$$

The finite difference discretization is consistent if

$$\lim_{h \rightarrow 0} \|\mathbf{T}\| = 0. \quad (4.0.7)$$

Therefore the discretization is consistent and second order accurate.

If we remove the error term in the equation above, and replace the exact solution $u(x_i, y_j)$ with the approximate solution U_{ij} which is the solution of the linear system of equations

$$\frac{U_{i-1,j} + U_{i+1,j}}{(h_x)^2} + \frac{U_{i,j-1} + U_{i,j+1}}{(h_y)^2} - \left(\frac{2}{(h_x)^2} + \frac{2}{(h_y)^2} \right) U_{ij} = f_{ij} \quad (4.0.8)$$

The finite difference scheme at a grid point (x_i, y_j) involves five grid points, east, north, west, south, and the center. The center is called the master grid point.

- Solve the linear system of equations to get an approximate solution at grid points (how?).
- Error analysis, implementation, visualization etc.

4.0.6 Matrix-vector form of the finite difference equations.

Generally, if one wants to use a direct method such as Gaussian elimination method or sparse matrix techniques, then one needs to find out the matrix structure. If one use an iterative method, such as Jacobi, Gauss Seidel, SOR(ω) methods, then it may be not necessarily to have the matrix and vector form.

In the matrix vector form $A\mathbf{U} = \mathbf{F}$, the unknown is a one dimensional array. For the two dimensional Poisson equations, the unknowns U_{ij} are a two dimensional array. Therefore we need to order it to get a one dimensional array. We also need to order the finite difference equations. It is common practice that we use the same ordering for the equations and for the unknowns.

There are two commonly used ordering. One is called the *natural ordering* that fits sequential computers. The other one is called the *red and black ordering* that fits parallel computers.

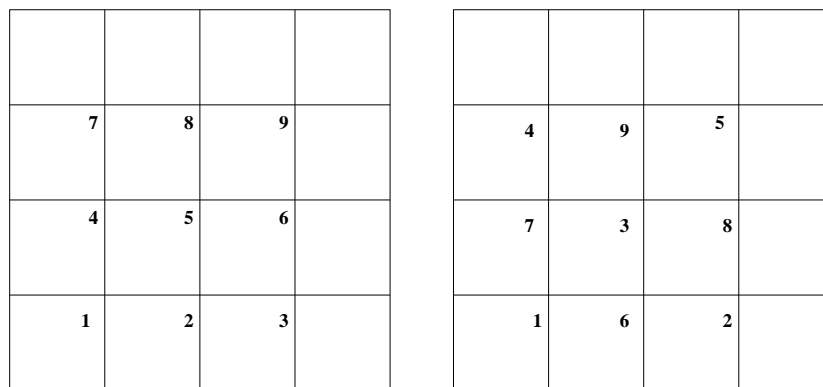


Figure 4.1: The natural ordering (left) and the red-black ordering (right).

The natural row ordering.

In the natural row ordering, we order the unknowns/equations row-wise, therefore the k -th equation corresponding to (i, j) with the following relation

$$k = i + (m - 1)(j - 1), \quad i = 1, 2, \dots, m - 1, \quad j = 1, 2, \dots, n - 1. \quad (4.0.9)$$

We use the following example to verify the matrix-vector form of the finite difference equations.

Assume that $h_x = h_y = h$, $m = n = 4$, so we will have nine equations and nine unknowns. The coefficient matrix is 9 by 9! To write down the matrix-vector form, we use a one-dimensional array \mathbf{x} to express the unknown U_{ij} .

$$\begin{aligned} x_1 = U_{11}, \quad x_2 = U_{21}, \quad x_3 = U_{31}, \quad x_4 = U_{12}, \quad x_5 = U_{22}, \\ x_6 = U_{32}, \quad x_7 = U_{13}, \quad x_8 = U_{23}, \quad x_9 = U_{33}. \end{aligned} \quad (4.0.10)$$

If we order the equations the same way as we order the unknowns, then the nine equations from the standard central finite difference scheme using the five point stencil are

$$\begin{aligned} \frac{1}{h^2} (-4x_1 + x_2 + x_4) &= f_{11} - \frac{u_{01} + u_{10}}{h^2}, \\ \frac{1}{h^2} (x_1 - 4x_2 + x_3 + x_5) &= f_{21} - \frac{u_{20}}{h^2} \\ \frac{1}{h^2} (x_2 - 4x_3 + x_6) &= f_{31} - \frac{u_{30} + u_{41}}{h^2} \\ \frac{1}{h^2} (x_1 - 4x_4 + x_5 + x_7) &= f_{12} - \frac{u_{02}}{h^2} \\ \frac{1}{h^2} (x_2 + x_4 - 4x_5 + x_6 + x_8) &= f_{22} \\ \frac{1}{h^2} (x_3 + x_5 - 4x_6 + x_9) &= f_{32} - \frac{u_{42}}{h^2} \\ \frac{1}{h^2} (x_4 - 4x_7 + x_8) &= f_{13} - \frac{u_{03} + u_{14}}{h^2} \\ \frac{1}{h^2} (x_5 + x_7 - 4x_8 + x_9) &= f_{23} - \frac{u_{24}}{h^2} \\ \frac{1}{h^2} (x_6 + x_8 - 4x_9) &= f_{33} - \frac{u_{34} + u_{43}}{h^2}. \end{aligned}$$

Now we can write down the coefficient matrix easily. It is *block tridiagonal* and has the following form:

$$A = \frac{1}{h^2} \begin{bmatrix} B & I & 0 \\ I & B & I \\ 0 & I & B \end{bmatrix} \quad (4.0.11)$$

where I is a 3×3 identity matrix:

$$B = \begin{bmatrix} -4 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & -4 \end{bmatrix}$$

For a general n by n grid, we will have

$$A = \frac{1}{h^2} \begin{bmatrix} B & I & & & \\ I & B & I & & \\ & & \ddots & \ddots & \ddots \\ & & & I & B \end{bmatrix}, \quad B = \begin{bmatrix} -4 & 1 & & & \\ & 1 & -4 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & -4 \end{bmatrix}.$$

Note that $-A$ is a symmetric positive definite matrix and it is weakly diagonally dominant. Therefore A is non-singular and there is a unique solution.

The matrix-vector form is useful to understand the structure of the linear system of equations, and it may be necessary if a direct method (such as Gaussian elimination) or sparse matrix techniques are used for solving the system. However, it is more convenient sometimes to use the two parameters system (i, j) , especially if an iterative method is used to solve the system. It is more intuitive and useful to visualize the data using two index system.

The eigenvalues and eigenvectors of A can be indexed by two parameters p and k corresponding to wave numbers in the x and y directions. The (p, k) -th eigenvector $u^{p,k}$ has n^2 elements for a n by n matrix of the form above:

$$u_{ij}^{p,k} = \sin(p\pi ih) \sin(k\pi jh), \quad i, j = 1, 2, \dots, n \tag{4.0.12}$$

for $p, k = 1, 2, \dots, n$. The corresponding eigenvalues are

$$\lambda^{p,k} = \frac{2}{h^2} \left(\cos(p\pi h) - 1 + \cos(k\pi h) - 1 \right). \tag{4.0.13}$$

The least dominant eigenvalue (the smallest in the magnitude) is

$$\lambda^{1,1} = -2\pi + O(h^2). \tag{4.0.14}$$

The dominant eigenvalue (the largest in the magnitude) is

$$\lambda^{n/2,n/2} \sim -\frac{4}{h^2}. \tag{4.0.15}$$

Therefore we have the following estimates:

$$\begin{aligned} \|A\|_2 &\sim \max |\lambda^{p,k}| = \frac{4}{h^2}, & \|A^{-1}\|_2 &= \frac{1}{\min |\lambda^{p,k}|} \sim \frac{1}{2\pi}, \\ \text{cond}_2(A) &= \|A\|_2 \|A^{-1}\|_2 \sim \frac{2}{\pi h^2} = O(n^2). \end{aligned} \tag{4.0.16}$$

Since the condition number is considered to be large, we should use double precision to reduce the effect of round off errors.

4.1 Basic iterative methods for solving linear system of equations

The idea of iterative methods is to start with an initial guess, then improve the solution iteratively. The first step is to re-write the original equation $f(\mathbf{x}) = 0$ to an equivalent form $\mathbf{x} = g(\mathbf{x})$ and then we can form an iteration: $\mathbf{x}^{(k+1)} = g(\mathbf{x}^{(k)})$. For example, to find $\sqrt[3]{5}$ is equivalent to solving the equation $x^3 - 5 = 0$. This equation can be written as $x = 5/x^2$ or $x = x - \frac{x^3 - 5}{3x^2}$. For the second one, the iteration

$$x^{(k+1)} = x^{(k)} - \frac{(x^{(k)})^3 - 5}{3(x^{(k)})^2}, \quad k = 0, 1, \dots$$

is called the Newton's iterative method. The mathematical theory behind this is the *fixed point theory*.

For a linear system of equations $A\mathbf{x} = \mathbf{b}$, we hope to re-write it as an equivalent form $\mathbf{x} = R\mathbf{x} + \mathbf{c}$ so that we can form an iteration $\mathbf{x}^{(k+1)} = R\mathbf{x}^{(k)} + \mathbf{c}$ given an initial guess $\mathbf{x}^{(0)}$. We want to choose such a R and \mathbf{c} that $\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \mathbf{x}_e = A^{-1}\mathbf{b}$. A common method is called the splitting approach in which we re-write the matrix A as

$$A = M - K, \quad \det(M) \neq 0. \quad (4.1.1)$$

Then $A\mathbf{x} = \mathbf{b}$ can be written as $(M - K)\mathbf{x} = \mathbf{b}$, or $M\mathbf{x} = K\mathbf{x} + \mathbf{b}$, or $\mathbf{x} = M^{-1}K\mathbf{x} + M^{-1}\mathbf{b}$, or $\mathbf{x} = R\mathbf{x} + \mathbf{c}$, where $R = M^{-1}K$ is called the iteration matrix and $\mathbf{c} = M^{-1}\mathbf{b}$ is a constant vector. The iterative process is then given an initial guess $\mathbf{x}^{(0)}$, we can get a sequence of $\{\mathbf{x}^{(k)}\}$ according to

$$\mathbf{x}^{(k+1)} = R\mathbf{x}^{(k)} + \mathbf{c} \quad (4.1.2)$$

We first discuss three basic iterative methods for solving $A\mathbf{x} = \mathbf{b}$. To derive the three methods, we re-write the matrix A as

$$A = D - L - U = \begin{bmatrix} a_{11} & & & & \\ & a_{22} & & & \\ & & a_{33} & & \\ & & & \ddots & \\ & & & & a_{nn} \end{bmatrix} - \begin{bmatrix} 0 & & & & \\ -a_{21} & 0 & & & \\ -a_{31} & -a_{32} & 0 & & \\ \vdots & \ddots & \ddots & \ddots & \\ -a_{n1} & -a_{n2} & \cdots & -a_{n,n-1} & 0 \end{bmatrix} \\ - \begin{pmatrix} 0 & -a_{12} & \cdots & \cdots & -a_{1n} \\ & 0 & \cdots & \cdots & -a_{2n} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & \vdots \\ & & & & 0 \end{pmatrix}$$

4.2 The Jacobi iterative method: Solve for the diagonals

The matrix vector form of the The Jacobi iterative method can be derived as follows:

$$\begin{aligned} (D - L - U)\mathbf{x} &= \mathbf{b} \\ D\mathbf{x} &= (L + U)\mathbf{x} + \mathbf{b} \\ \mathbf{x} &= D^{-1}(L + U)\mathbf{x} + D^{-1}\mathbf{b} \\ \mathbf{x}^{(k+1)} &= D^{-1}(L + U)\mathbf{x}^{(k)} + D^{-1}\mathbf{b}. \end{aligned}$$

The component form can be written as

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1, j \neq i}^n a_{ij}x_j^{(k)} \right) / a_{ii}, \quad i = 1, 2, \dots, n. \quad (4.2.1)$$

The component form is useful for implementation while the matrix-vector form is good for convergence analysis.

4.3 The Gauss-Seidel iterative method: Use the most updated

In the Jacobi iterative method, when we compute x_2^{k+1} , we have already computed x_1^{k+1} . Assume that x_1^{k+1} is a better approximation than x_1^k , why can not we use x_1^{k+1} when we update x_2^{k+1} instead of x_1^k ? With this idea, we get a new iterative method which is the Gauss-Seidel iterative method for solving $A\mathbf{x} = \mathbf{b}$. The component form is

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right) / a_{ii}, \quad i = 1, 2, \dots, n. \quad (4.3.1)$$

To derive the matrix-vector form of the Gauss-Seidel iterative method, we write the component form above to a form $(\)^{(k+1)} = (\)^{(k)} + (\)$. The component form above is equivalent to

$$\sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} + a_{ii}x_i^{(k+1)} = b_i - \sum_{j=i+1}^n a_{ij}x_j^{(k)}, \quad i = 1, 2, \dots, n, \quad (4.3.2)$$

which is the component form of the following system of equations

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

Thus the iteration matrix of the Gauss-Seidel iterative method is $(D - L)^{-1}U$, and the constant vector is $\mathbf{c} = (D - L)^{-1}\mathbf{b}$.

4.3.1 Implementation details

The iterative methods are an infinity process. However, when we implement on a computer, we have to stop it in finite time. One of several of the following stopping criteria are used

- $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| \leq tol.$
- $\frac{\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|}{\|\mathbf{x}^{(k)}\|} \leq tol.$
- $\|r(\mathbf{x}^{(k)})\| \leq tol.$
- $k \geq k_{max},$

where tol and k_{max} are two given parameters.

4.3.2 Pseudo-code of the Gauss-Seidel iterative method:

```
[x,k] = my_gs(n,a,b,x0,tol)
error = 1e5; x=x0; k=0;
while error > tol
for i=1:n
    x(i) = b(i);
    for j=1:n
        if j~= i
            x(i) = x(i) - a(i,j)*x(j);
        end
    end
    x(i) = x(i)/a(i,i);
end
error = norm(x-x0); %default is 2-norm
x0=x; k = k+1; % replace the old value, add the counter.
end %end while
```

4.3.3 The Gauss-Seidel iterative method for 2-point boundary value problem

Following section 1.3, we have

$$\frac{U_{i-1} - 2U_i + U_{i+1}}{h^2} = f(x_i), \quad i = 1, 2, \dots, n-1.$$

If we use the same ordering for the equations and unknowns, then the diagonals are always $-2/h^2$, the Jacobi iteration is simply

$$U_i^{(k+1)} = \frac{U_{i-1}^{(k)} + U_{i+1}^{(k)}}{2} + \frac{h^2}{2} f(x_i, y_j), \quad i = 1, 2, \dots, n-1.$$

No matrix is formed (or only matrix-vector multiplication is needed), no ordering is necessary (assuming that the equations and unknowns have the same ordering). For the Gauss-Seidel iteration, from k -th to $k + 1$ -th iteration, we can use the following

$$U_i^{(k+1)} = U_i^{(k)}, \quad i = 1, 2, \dots, n - 1,$$

$$U_i^{(k+1)} = \frac{U_{i-1}^{(k+1)} + U_{i+1}^{(k+1)}}{2} + \frac{h^2}{2} f(x_i), \quad i = 1, 2, \dots, n - 1.$$

If $U_i^{(k+1)}$ has not been updated, then it use the value from k -th iteration, otherwise it uses the most updated one.

4.3.4 The Gauss-Seidel iterative method for the finite difference method for Poisson equation

For the Poisson equation $u_{xx} + u_{yy} = f(x, y)$, if we use the standard 5-point central finite difference scheme

$$\frac{U_{i-1,j} + U_{i+1,j} - 4U_{ij} + U_{i,j-1} + U_{i,j+1}}{h^2} = f(x_i, y_j)$$

and the same ordering for the equations and unknowns, then the Jacobi iteration is

$$U_{ij}^{(k+1)} = \frac{U_{i-1,j}^{(k)} + U_{i+1,j}^{(k)} + U_{i,j-1}^{(k)} + U_{i,j+1}^{(k)}}{4} + \frac{h^2}{4} f(x_i, y_j),$$

$$i = 1, 2, \dots, n - 1, \quad j = 1, 2, \dots, n - 1,$$

if the solution is prescribed along the boundary (Dirichlet BC). Again, no matrix is needed, no ordering is necessary. We do not need to transform the two dimensional array to a one dimensional one. The implementation is rather simple.

4.4 The successive over-relaxation (SOR(ω)) iterative method

The Jacobi and Gauss-Seidel methods can be quite slow. The SOR(ω) iterative method is an acceleration method by choosing appropriate parameter ω . The SOR(ω) iterative method is

$$\mathbf{x}^{(k+1)} = (1 - \omega)\mathbf{x}^k + \omega\tilde{\mathbf{x}}_{GS}^{k+1}. \tag{4.4.1}$$

The component form is

$$x_i^{k+1} = (1 - \omega)x_i^k + \omega \left(\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right) / a_{ii} \right). \tag{4.4.2}$$

Note that, it is incorrect to get G-S result first, then do the linear interpolation.

The idea of the SOR method is to interpolate \mathbf{x}^k and \mathbf{x}_{GS}^{k+1} to get a better approximation. When $\omega \leq 1$, the new point of $(1 - \omega)\mathbf{x}^k + \omega\tilde{\mathbf{x}}_{GS}^{k+1}$ is between \mathbf{x}^k and \mathbf{x}_{GS}^{k+1} , and this it is called interpolation. The iterative method is called under relaxation. When $\omega > 1$, the new point of $(1 - \omega)\mathbf{x}^k + \omega\tilde{\mathbf{x}}_{GS}^{k+1}$ is outside \mathbf{x}^k and \mathbf{x}_{GS}^{k+1} , and this it is called extrapolation. The iterative method is called over relaxation. Since the approach is used at every iteration, it is called successive over relaxation (SOR) method.

To derive the matrix-vector form of SOR(ω) method, we write its component form as

$$a_{ii}x_i^{k+1} + \omega \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} = a_{ii}(1 - \omega)x_i^k + \omega b_i - \omega \sum_{j=i+1}^n a_{ij}x_j^{(k)}. \quad (4.4.3)$$

This is equivalent to

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

Thus the iteration matrix and constant vector of the SOR(ω) method are

$$R_{SOR}(\omega) = (D - \omega L)^{-1}((1 - \omega)D + \omega U), \quad \mathbf{c}_{SOR} = \omega(D - \omega L)^{-1}\mathbf{b}. \quad (4.4.4)$$

4.5 Convergence of basic iteration methods $\mathbf{x}^{(k+1)} = R\mathbf{x}^{(k)} + \mathbf{c}$

Using an iterative method, we will get a vector sequence of $\{\mathbf{x}^{(k)}\}$ and we know how to tell whether it is convergent or not. However, for an iterative method, we need to consider all possible initial guesses and constant vector \mathbf{c} .

If the vector sequence of $\{\mathbf{x}^{(k)}\}$ converges to \mathbf{x}^* , then by taking limit on both sides of the iterative scheme, we have

$$\mathbf{x}^* = R\mathbf{x}^* + \mathbf{c}. \quad (4.5.1)$$

The above equality is called the consistency condition.

Definition 4.5.1 *The iteration methods $\mathbf{x}^{(k+1)} = R\mathbf{x}^{(k)} + \mathbf{c}$ is convergent if for any initial guess $\mathbf{x}^{(0)}$ and constant vector \mathbf{c} , the vector sequence of $\{\mathbf{x}^{(k)}\}$ converges to the solution of the system of equations $\mathbf{x}^* = R\mathbf{x}^* + \mathbf{c}$.*

Now we discuss a few sufficient conditions that guarantee convergence of a basic iterative method.

Theorem 4.5.1 *If there is an associated matrix norm such that $\|R\| < 1$, then the iteration method $\mathbf{x}^{(k+1)} = R\mathbf{x}^{(k)} + \mathbf{c}$ converges.*

Proof: Let $\mathbf{e}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^*$, from the iterative method $\mathbf{x}^{(k+1)} = R\mathbf{x}^{(k)} + \mathbf{c}$ and the consistency condition $\mathbf{x}^* = R\mathbf{x}^* + \mathbf{c}$, we have

$$\mathbf{e}^{(k+1)} = R\mathbf{e}^{(k)}$$

$$0 \leq \|\mathbf{e}^{(k+1)}\| = \|R\mathbf{e}^{(k)}\| \leq \|R\| \|\mathbf{e}^{(k)}\| \leq \|R\| \|R\| \|\mathbf{e}^{(k-1)}\| \leq \dots \leq \|R\|^{k+1} \|\mathbf{e}^{(0)}\|$$

Thus we conclude that $\lim_{k \rightarrow \infty} \|\mathbf{e}^{(k)}\| = 0$, or equivalently, $\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \mathbf{x}^*$.

Example: Given

$$R = \begin{pmatrix} 0.9 & 0.05 \\ -0.8 & -0.1 \end{pmatrix}$$

Does the iterative method $\mathbf{x}^{(k+1)} = R\mathbf{x}^{(k)} + \mathbf{c}$ converge?

We can easily get $\|R\|_1 = 1.7$, which leads to no conclusion; and $\|R\|_\infty = 0.95 < 1$, which lead to the conclusion that the iterative method converges.

4.5.1 Convergence speed

In the theorem above, the k -th error depends on the initial one that we do not know. The following error estimate does not need the initial error.

Theorem 4.5.2 *If there is an associated matrix norm such that $\|R\| < 1$, we have the following error estimate for the iteration method $\mathbf{x}^{(k+1)} = R\mathbf{x}^{(k)} + \mathbf{c}$.*

$$\|\mathbf{e}^{(k+1)}\| \leq \frac{\|R\|^k}{1 - \|R\|} \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|. \quad (4.5.2)$$

Proof: From the iterative method $\mathbf{x}^{(k+1)} = R\mathbf{x}^{(k)} + \mathbf{c}$, we also have $\mathbf{x}^{(k)} = R\mathbf{x}^{(k-1)} + \mathbf{c}$. Subtracting the two, we get

$$\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} = R(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}) = R^2(\mathbf{x}^{(k-1)} - \mathbf{x}^{(k-2)}) = \dots = R^k(\mathbf{x}^{(1)} - \mathbf{x}^{(0)}).$$

Since

$$\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(*)} + \mathbf{x}^* - \mathbf{x}^{(k)} = \mathbf{e}^{(k+1)} - \mathbf{e}^{(k)} = (R - I)\mathbf{e}^{(k)}.$$

Combining the two equalities above we get

$$-(I - R)\mathbf{e}^{(k)} = R^k (\mathbf{x}^{(1)} - \mathbf{x}^{(0)}).$$

This leads to

$$\|\mathbf{e}^{(k)}\| = \|(I - R)^{-1} R^k (\mathbf{x}^{(1)} - \mathbf{x}^{(0)})\|.$$

Finally from the Banach's lemma, we have

$$\|\mathbf{e}^{(k)}\| \leq \frac{\|R\|^k}{1 - \|R\|} \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|.$$

4.5.2 Other sufficient conditions using the original matrix A

Theorem 4.5.3 *If A is strictly row diagonally dominant matrix, then both Jacobi and Gauss-Seidel methods converge. The Gauss-Seidel method converges faster in the sense that*

$$\|R_{GS}\|_{\infty} \leq \|R_J\|_{\infty} < 1$$

Proof: The proof of the first part is easy. For the Jacobi method, we have $R = D^{-1}(L + U)$, thus

$$\|R_J\|_{\infty} = \max_i \sum_{j=1, j \neq i}^n \left| \frac{a_{ij}}{a_{ii}} \right| < 1.$$

The proof for the Gauss-Seidel method is not trivial and long, we refer the readers to the book [J. W. Demmel] on page 287-288.

For general matrices, it is unclear whether the Jacobi or Gauss-Seidel method converges faster even if they both converge.

Theorem 4.5.4 *If A is a symmetric positive definite (SPD) matrix, then the $SOR(\omega)$ method converges for $0 < \omega < 2$.*

Again we refer the readers to the book [J. W. Demmel] on page 290-291.

Theorem 4.5.5 *If A is a weakly row diagonally dominant matrix,*

$$\sum_{j=1, j \neq i}^n |a_{ij}| \leq |a_{ii}|, \quad , i = 1, 2, \dots, n,$$

with at least on inequality is strictly and A is irreducible, that is, there is no permutation matrix such that

$$P^T A P = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}$$

then both Jacobi and Gauss-Seidel methods converge. The Gauss-Seidel method converges faster in the sense that

$$\|R_{GS}\|_{\infty} \leq \|R_J\|_{\infty} < 1.$$

Lemma 4.5.1 *A is irreducible if the graph of the matrix is strongly connected.*

4.5.3 A sufficient and necessary condition

The spectral radius of a matrix A is defined as

$$\rho(A) = \max_i |\lambda_i(A)| \tag{4.5.3}$$

Note that from $\rho(A) \leq \|A\|$ from any associated matrix norms. The proof is quite simple. Let λ_{i^*} be the eigenvalue of A such that $\rho(A) = |\lambda_{i^*}|$ and $\mathbf{x}^* \neq \mathbf{0}$ is the corresponding eigenvector, then we have $A\mathbf{x}^* = \lambda_{i^*}\mathbf{x}^*$. Thus we have $|\lambda_{i^*}|\|\mathbf{x}^*\| \leq \|A\|\|\mathbf{x}^*\|$. Since $\|\mathbf{x}^*\| \neq 0$, we get $\rho(A) \leq \|A\|$.

Theorem 4.5.6 *An iteration method $\mathbf{x}^{(k+1)} = R\mathbf{x}^{(k)} + \mathbf{c}$ converges for arbitrary $\mathbf{x}^{(0)}$ and \mathbf{c} if and only if $\rho(R) < 1$.*

Proof: Part A: If the iterative method converge, then $\rho(R) < 1$. This can be done using counter proof method. Assume that $\rho(R) > 1$, then let $A\mathbf{x}^* = \lambda_{i^*}\mathbf{x}^*$, with $\rho(A) = |\lambda_{i^*}| > 1$ and $\|\mathbf{x}^*\| \neq 0$. If we set $\mathbf{x}^{(0)} = \mathbf{x}^*$ and $\mathbf{c} = \mathbf{0}$, then we have $\mathbf{x}^{(k+1)} = \lambda_{i^*}^{k+1}\mathbf{x}^*$ which do not have a limit since $\lambda_{i^*}^{k+1} \rightarrow \infty$. The case of $\rho(R) = 1$ is left as an exercise.

Proof: Part B: If $\rho(R) < 1$, then the iterative method converges for arbitrary $\mathbf{x}^{(0)}$ and \mathbf{c} . The key in the proof is to find a matrix norm such that $\|R\| < 1$.

From linear algebra Jordan's theorem we know that for any square matrix R , it is similar to a Jordan canonical form, that is, there is a nonsingular matrix S such that

$$S^{-1}RS = \begin{pmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & \ddots \\ & & & & J_p \end{pmatrix}, \quad \begin{pmatrix} \lambda_i & 1 & & & \\ & \lambda_i & 1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 \\ & & & & \lambda_i \end{pmatrix},$$

Note that $\|S^{-1}RS\|_\infty = \rho(R) < 1$ if all Jordan blocks are 1 by 1 matrix, otherwise $\|S^{-1}RS\|_\infty \leq \rho(R) + 1$. Since $\rho(R) < 1$, we can find $\epsilon > 0$ such that $\rho(R) + \epsilon < 1$, say $\epsilon = (1 - \rho(R))/2$. Consider a particular Jordan Block J_i and assume it is a k by k matrix. Let

$$D_i(\epsilon) = \begin{pmatrix} 1 & & & \\ & \epsilon & & \\ & & \ddots & \\ & & & \ddots \\ & & & & \epsilon^{k-1} \end{pmatrix}, \quad D_i(\epsilon)J_iD_i^{-1}(\epsilon) = \begin{pmatrix} \lambda_i & \epsilon & & & \\ & \lambda_i & \epsilon & & \\ & & \ddots & \ddots & \\ & & & \ddots & \epsilon \\ & & & & \lambda_i \end{pmatrix},$$

Using this diagonal matrix, we can get

$$D(\epsilon) = \begin{pmatrix} D_1(\epsilon) & & & & \\ & D_1(\epsilon) & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & D_p(\epsilon) \end{pmatrix},$$

Thus $\|D^{-1}(\epsilon)S^{-1}RSD\|_\infty = \rho(R) + \epsilon < 1$.

Finally we define a new matrix norm as

$$\|R\|_{new} = \|D^{-1}(\epsilon)S^{-1}RSD\|_\infty \quad (4.5.4)$$

It can easily show that the definition above is indeed an associate matrix norm and $\|R\|_{new} = \rho(R) + \epsilon < 1$, we conclude that the iterative method converges for any initial guess and vector \mathbf{c} .

4.6 Discussion for the Poisson equations, what is the best ω ?

For the finite difference methods for Poisson equations in 1D and 2D, we can find the eigenvalues of the coefficient matrix, which also leads to eigenvalues of the iteration matrix (Jacobib, Gauss Seidel, SOR ω). For 1D model problem $u''(x) = f(x)$ with the Dirichlet boundary condition $u(0)$ and $u(1)$ are prescribed, the coefficient matrix is

$$A_{FD} = \frac{1}{h^2} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & -2 \end{bmatrix}, \quad A = \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & -2 \end{bmatrix}.$$

The matrix is weakly row diagonally dominant (not strictly) and irreducible; $-A$ is symmetric positive definite; or A is symmetric negative definite.

Theorem 4.6.1 *For the above n by n matrix, we have*

$$\rho(R_J) = \max_{1 \leq i \leq n} \left| 1 + \frac{\lambda_i(A)}{2} \right|, \quad (4.6.1)$$

where $\lambda_i(A)$, $i = 1, 2, \dots, n$ are the eigenvalues of the matrix A , R_J is the iteration matrix of the Jacobi method.

Proof: We know that $R_j = D^{-1}(L + U)$ and $D = -2I$. Let λ be an eigenvalue of J_R , then $\det(\lambda I - J_R) = 0$, or $\det(\lambda I - D^{-1}(L + U)) = 0$, or $\det(D^{-1}(\lambda D - (L + U))) = 0$, or $\det(D)\det(\lambda D - (L + U)) = 0$. Since $\det(D) \neq 0$, we have $\det(\lambda D - (L + U)) = 0$, or $\det((\lambda - 1)D + D - (L + U)) = 0$, or $\det((\lambda - 1)D + A) = 0$, or $\det((1 - \lambda)D - A) = 0$, or $\det(-2(1 - \lambda)I - A) = 0$ since $D = -2I$. Thus $-2(1 - \lambda)$ is an eigenvalue of A , or

$$\lambda_i(R_J) = 1 + \frac{\lambda_i(A)}{2}, \quad i = 1, 2, \dots, n.$$

From the relation above, we have the theorem right way.

The following lemma gives the eigenvalues of a tri-diagonal matrix.

Lemma 4.6.1 *Let A be the following n by n matrix*

$$A = \begin{bmatrix} \alpha & \beta & & & \\ \beta & \alpha & \beta & & \\ & \ddots & \ddots & \ddots & \\ & & & \beta & \alpha \end{bmatrix}.$$

The eigenvalues of A are the following

$$\lambda_k = \alpha + 2\beta \cos \frac{k\pi}{n+1}, \quad k = 1, 2, \dots, n.$$

The eigenvector corresponding to λ_k is

$$\chi_{k,j} = \sin \frac{kj\pi}{n+1}, \quad j = 1, 2, \dots, n.$$

Proof: It is easy to check that $A\chi_k = \lambda_k\chi_k$.

When $\alpha = -2$, $\beta = 1$, we have

$$\lambda_k = -2 + 2 \cos \frac{k\pi}{n+1}, \quad k = 1, 2, \dots, n.$$

Thus the spectral radius of the Jacobi iterative method for 1D Poisson problem is

$$\begin{aligned} \rho(R_J) &= \max_{1 \leq k \leq n} \left| 1 + \frac{\lambda_k(A)}{2} \right| = \max_{1 \leq k \leq n} \left| 1 - \frac{2(1 - \cos(k\pi/(n+1)))}{2} \right| \\ &= \max_{1 \leq k \leq n} \left| \cos \frac{k\pi}{n+1} \right| = \cos \frac{\pi}{n+1} \sim 1 - \frac{1}{2} \left(\frac{\pi}{n+1} \right)^2. \end{aligned}$$

We can see that as n is getting larger, the spectral radius is getting close to unit indicating slower convergence.

Once we know the spectral radius, we also know roughly the number of iterations needed to reach the desired accuracy. For example, if we wish to have roughly six significant digit, the we should set $\rho(R)^k \leq 10^{-6}$ or $k \geq -6 \log_{10}(\rho(R))$.

4.6.1 Finite difference method for the Poisson equation in two dimensions

In two space dimensions, we have parallel results. The eigenvalues for the matrix $A = h^2 A_{FD}$ of N by N matrix $N = n^2$ are

$$\lambda_{i,j} = - \left(4 - 2 \left(\cos \frac{i\pi}{n+1} + \cos \frac{j\pi}{n+1} \right) \right), \quad i, j = 1, 2, \dots, n. \quad (4.6.2)$$

The diagonals of the matrix are -4 and we have

$$\lambda_{i,j}(R_J) = 1 + \frac{\lambda_{i,j}(A)}{4}, \quad i, j = 1, 2, \dots, n.$$

Thus

$$\begin{aligned} \rho(R_J) &= \max_{1 \leq i, j \leq n} \left| 1 + \frac{\lambda_{i,j}(A)}{4} \right| = \max_{1 \leq i, j \leq n} \left| 1 - \frac{2(1 - \cos(i\pi/(n+1)) + \cos(j\pi/(n+1)))}{4} \right| \\ &= \max_{1 \leq i, j \leq n} \left| \cos \frac{i\pi}{n+1} + \cos \frac{j\pi}{n+1} \right| = \cos \frac{\pi}{n+1} \sim 1 - \frac{1}{2} \left(\frac{\pi}{n+1} \right)^2. \end{aligned}$$

We see that the results in 1D and 2D are pretty much the same. To derive the best ω for the $SOR(\omega)$ method, we need to derive the eigenvalue relation between the original matrix and iteration matrix. Note that $R_{SOR} = (D - \omega L)^{-1}((1 - \omega)D + \omega U)$.

Theorem 4.6.2 *The optimal ω for the $SOR(\omega)$ method for the system of equations derived from the finite difference method for the Poisson equation is*

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - (\rho(R_J))^2}} = \frac{2}{1 + \sqrt{1 - \cos^2 \frac{\pi}{n+1}}} = \frac{2}{1 + \sin \frac{\pi}{n+1}} \sim \frac{2}{1 + \frac{\pi}{n+1}} \quad (4.6.3)$$

Below is the sketch of the proof.

- Step 1. Show the eigenvalue relation between R_J and $R_{SOR}(\omega)$

$$\lambda_{SOR}(\omega) = 1 - \omega + \frac{1}{2} \omega^2 \lambda_J^2 \pm \omega \lambda_J \sqrt{1 - \omega + \frac{\omega^2 \lambda_J^2}{4}} \quad (4.6.4)$$

- Step 2. Find the extreme value (minimum) of above as a function of λ_J which leads to the optimal ω .

We refer the readers to the book [J. W. Demmel] on page 292-293 for the proof.

Remark 4.6.1

- The spectral radius of $R_{SOR}(\omega)$ is

$$\rho(R_{SOR}) = \begin{cases} 1 - \omega + \frac{1}{2}\omega^2\rho(R_J)^2 + \omega\rho(R_J)\sqrt{1 - \omega + \frac{\omega^2\rho(R_J)^2}{4}}, & 0 < \omega < \omega_{opt}, \\ \omega - 1, & \omega_{opt} < \omega < 2. \end{cases}$$

If we plot $\rho(R_{SOR})$ against ω , we can see it is a quadratic curve between $0 < \omega < \omega_{opt}$ and it is flat as ω is getting closer to ω_{opt} which means it is less sensitive in the neighborhood of ω_{opt} ; while the second piece is a linear function. Thus, we would rather choose ω larger than smaller.

- The optimal ω is only for the Poisson equation, not for other elliptic problems. However, it gives a good indication of the best ω is the diffusion is dominant in reference to the mesh size.

Chapter 5

Computing algebraic eigenvalues and eigenvectors

5.1 Preliminary

Given a square matrix $A \in R^{n,n}$, if we can find a number $\lambda \in C$ and $\mathbf{x} \neq \mathbf{0}$ such that $A\mathbf{x} = \lambda\mathbf{x}$, then λ is called an *eigenvalue* of A , \mathbf{x} is called a corresponding *eigenvector* to λ . Note that if $A\mathbf{x} = \lambda\mathbf{x}$, then $A(c\mathbf{x}) = \lambda(c\mathbf{x})$ for any non-zero constant c , in other words, eigenvector can differ by a constant. Often we prefer to use an eigenvector with unit length ($\|\mathbf{x}\| = 1$). We call (λ, \mathbf{x}) an *eigen-pair* if $A\mathbf{x} = \lambda\mathbf{x}$ ($\mathbf{x} \neq \mathbf{0}$).

For an eigen-pair (λ, \mathbf{x}) , we have $A\mathbf{x} - \lambda\mathbf{x} = \mathbf{0}$. This means that $(\lambda I - A)\mathbf{x} = \mathbf{0}$ has non-zero (or not unique) solutions. This indicates that (λI_A) is singular, or $\det(\lambda I_A) = 0$. Thus λ must be a root of the *characteristic polynomial* of the matrix A , $\det(\lambda I - A) = \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0$.

There are n eigenvalues for an n by n square matrix. The eigenvalues can be real, complex numbers, repeated roots. If the matrix is real, then the complex eigenvalues are in pairs, that is, if $\lambda = a + bi$ is one eigenvalue, then $\bar{\lambda} = a - bi$ is also an eigenvalue. If A is a real and symmetric matrix, then all the eigenvalues are real numbers.

Different eigenvectors corresponding to different eigenvalues are linear independent. If an eigenvalue λ^* has multiplicity p , which means that characteristic polynomial has the factor $(\lambda - \lambda^*)^p$, but no the factor $(\lambda - \lambda^*)^{p+1}$, the number is independent of eigenvectors corresponding to λ^* is less than or equal to p , recall some examples in class. If an n by n square matrix A has n linear independent eigenvectors, then A is diagonalizable, that is, there is a nonsingular matrix S , such that $S^{-1}AS = D$, where $D = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$ is a diagonal matrix.

For convenience of discussion, we will use the following notations. We arrange eigenval-

ues of a matrix A according to

$$|\lambda_1| \geq |\lambda_2| \geq \cdots |\lambda_i| \geq \cdots \geq |\lambda_n|.$$

Thus $\rho(A) = |\lambda_1|$, λ_1 is called the **dominant eigenvalue** (can be more than one), while λ_n is called the **least dominant eigenvalue**.

There are many applications of eigenvalue problems. Below are a few of them

- Frequencies if vibration, resonates, etc.
- Spectral radius and convergence of iterative method
- Discrete form of continuous eigenvalue problems, for example, the Sturm-Louville problem

$$\begin{aligned} -(p(x)u'(x))' + q(x)u(x) &= \lambda u(x), \quad 0 < x < 1, \\ u(0) = 0, \quad u(1) &= 0. \end{aligned}$$

After applying a finite difference or finite element method, we would have $A\mathbf{x} = \lambda\mathbf{x}$. The solutions are the basis for the Fourier series expansion.

- Stability analysis of dynamic systems, or numerical methods.

5.2 The Power's method

The idea of the power method is: starting from a non-zero vector $\mathbf{x}^{(0)} \neq \mathbf{0}$ (an approximation to an eigenvector), then form an iteration

$$\mathbf{x}^{(k+1)} = A\mathbf{x}^{(k)} = A^2\mathbf{x}^{(k-1)} = \cdots A^{k+1}\mathbf{x}^{(0)}$$

Then under some conditions, we can extract an eigen-pair information from the sequence!

With the assumption that λ_1 satisfies $|\lambda_1| > |\lambda_2|$, which is an essential condition, then we can show that $\mathbf{x}^{(k)} \sim C\mathbf{x}_1$, and $x_p^{(k+1)}/x_p^{(k)} \sim \lambda_1$ as k is very large, where $|x_p^{(k+1)}| = \|\mathbf{x}^{(k+1)}\|$.

Sketch of the proof:

While feasible in theory, the idea is not practical in computation because $\mathbf{x}^{(k)} \rightarrow \mathbf{0}$ if $\rho(A) < 1$ and $\mathbf{x}^{(k)} \rightarrow \infty$ if $\rho(A) > 1$. The solution is to rescale the vector sequence, which leads to the following power method.

Given $\mathbf{x}^{(0)} \neq \mathbf{0}$, form the following iteration

for $k = 1$ until converges

$$\begin{aligned} \mathbf{y}^{(k+1)} &= A\mathbf{x}^{(k)} \\ \mathbf{x}^{(k+1)} &= \frac{\mathbf{y}^{(k+1)}}{\|\mathbf{y}^{(k+1)}\|_2} \\ \mu_{k+1} &= (\mathbf{x}^{(k+1)})^T A\mathbf{x}^{(k+1)} \end{aligned}$$

end

We can use the following stopping criteria: $|\mu_{k+1} - \mu_k| < tol$ or $\|\mathbf{y}^{(k+1)} - \mathbf{y}^{(k)}\| < tol$ or both.

Under some conditions, the sequence of the pair $(\mu_k, \mathbf{x}^{(k)})$ converge to the eigen-pair corresponding to the dominant eigenvalue.

For simplicity of the proof, we assume that A has a complete eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$, $\|\mathbf{v}_i\| = 1$, $A\mathbf{v}_i = \lambda_i\mathbf{v}_i$.

Theorem 5.2.1 *Assume that $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$, and $\mathbf{x}^{(0)} = \sum_{i=1}^n \alpha_i \mathbf{v}_i$ with $\alpha_1 \neq 0$, then the pair $(\mu_k, \mathbf{x}^{(k)})$ the power method converges to the eigen-pair corresponding to the dominant eigenvalue.*

Proof: Note that

$$\begin{aligned} \mathbf{y}^{(k)} &= A\mathbf{x}^{(k-1)} = A \frac{\mathbf{y}^{(k-1)}}{\|\mathbf{y}^{(k-1)}\|_2} = \gamma_k A\mathbf{y}^{(k-1)} \\ &= \gamma_k \gamma_{k-1} A^2 \mathbf{y}^{(k-2)} = \dots \gamma_k \gamma_{k-1} \dots \gamma_1 A^{k-1} \mathbf{y}^{(1)} = C_k A^k \mathbf{x}^{(0)} \end{aligned}$$

where $\gamma_k \gamma_{k-1} \dots \gamma_1$ and C_k are some constants. Since $\mathbf{x}^{(k)}$ is parallel to $\mathbf{y}^{(k)}$ and has unity length in 2-norm, we must have

$$\mathbf{x}^{(k)} = \frac{\mathbf{y}^{(k)}}{\|\mathbf{y}^{(k)}\|_2}.$$

On the other hand, we know we have

$$A^k \mathbf{x}^{(0)} = \lambda_1^k \left(\alpha_1 v_1 + \alpha_2 v_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k + \dots \alpha_n v_n \left(\frac{\lambda_n}{\lambda_1} \right)^k \right)$$

Thus we have

$$\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \lim_{k \rightarrow \infty} \frac{\mathbf{x}^{(k)}}{\|\mathbf{x}^{(k)}\|_2} = \frac{\alpha_1 \mathbf{v}_1}{\|\alpha_1 \mathbf{v}_1\|_2} = \pm \mathbf{v}_1.$$

and

$$\lim_{k \rightarrow \infty} \mu_k = \lim_{k \rightarrow \infty} (\mathbf{x}^{(k)})^T A\mathbf{x}^{(k)} = \lambda_1.$$

5.2.1 The power method using the infinity norm

If we use different scaling, we can get different power method. Using the infinity norm, first we introduce the x_p notation for a vector \mathbf{x} . Given a vector \mathbf{x} , x_p is the first component such that $|x_p| = \|\mathbf{x}\|_\infty$, and p is the index. For example, if $\mathbf{x} = [2 \ -1 \ -5 \ 5 \ -5]^T$, then $x_p = -5$ with $p = 3$.

5.3 The inverse power method for the least dominant eigenvalue

If A is invertible, then $1/\lambda_i$ are the eigenvalues of A^{-1} and $1/\lambda_n$ will be the dominant eigenvalue of A^{-1} . However, the following inverse power method for the least dominant eigenvalue without the need to have the intermediate steps.

Given $\mathbf{x}^{(0)} \neq \mathbf{0}$, form the following iteration

for $k = 1$ until converges

$$\begin{aligned} A\mathbf{y}^{(k+1)} &= \mathbf{x}^{(k)} \\ \mathbf{x}^{(k+1)} &= \frac{\mathbf{y}^{(k+1)}}{\|\mathbf{y}^{(k+1)}\|_2} \\ \mu_{k+1} &= (\mathbf{x}^{(k+1)})^T A\mathbf{x}^{(k+1)}. \end{aligned}$$

end

With similar conditions ($|\lambda_n| < |\lambda_{n-1}| \leq \dots \leq |\lambda_1|$, the essential condition), one can prove that

$$\lim_{k \rightarrow \infty} \mu_k = \lambda_n, \quad \lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \mathbf{v}_n, \quad \text{with } \|\mathbf{v}_n\|_2 = 1.$$

Note that, at each iteration, we need to solve a linear system of equations with the same coefficient matrix. This is the most expensive part of the algorithm. An efficient implementation is to get the matrix decomposition done outside the loop. Let $PA = LU$, the algorithm can be written as follows.

Given $\mathbf{x}^{(0)} \neq \mathbf{0}$, form the following iteration

for $k = 1$ until converges

$$L\mathbf{z}^{(k+1)} = P\mathbf{x}^{(k)}$$

$$U\mathbf{y}^{(k+1)} = \mathbf{z}^{(k+1)}$$

$$\mathbf{x}^{(k+1)} = \frac{\mathbf{y}^{(k+1)}}{\|\mathbf{y}^{(k+1)}\|_2}$$

$$\mu_{k+1} = (\mathbf{x}^{(k+1)})^T A \mathbf{x}^{(k+1)}$$

end

5.4 Gershgorin theorem and the shifted inverse power method

If we know a good approximate σ to an eigenvalue λ_p such that

$$|\lambda_p - \sigma| < \min_{1 \leq i \leq n, i \neq p} |\lambda_i - \sigma|.$$

That is, $\lambda_p - \sigma$ is the least dominant eigenvalue of $A - \sigma I$. We can use the following shifted inverse power method to find the eigenvalue λ_p and its corresponding eigenvector. Given $\mathbf{x}^{(0)} \neq \mathbf{0}$, form the following iteration

for $k = 1$ until converges

$$(A - \sigma I)\mathbf{y}^{(k+1)} = \mathbf{x}^{(k)}$$

$$\mathbf{x}^{(k+1)} = \frac{\mathbf{y}^{(k+1)}}{\|\mathbf{y}^{(k+1)}\|_2}$$

$$\mu_{k+1} = (\mathbf{x}^{(k+1)})^T A \mathbf{x}^{(k+1)}$$

end

Thus if we can find good approximations of any eigenvalue, we can use the shifted inverse power method to compute it. Now the question is how do we roughly locate the eigenvalues of a matrix A . Gershgorin theorem provides some useful hints.

Definition 5.4.1 Given a matrix $A \in C^{n \times n}$, the circle (all points within the circle on the complex plane)

$$|\lambda - a_{ii}| \leq \sum_{j=1, j \neq i}^n |a_{ij}| \tag{5.4.1}$$

is called the i -th Gershgorin circle.

Theorem 5.4.1 *Gershgorin Theorem.*

1. Any eigenvalue have to be in one of Gershgorin circles.
2. The union of k Gershgorin circles, which do not intersect with other $n - k$ circles, contains precisely k eigenvalues of A .

Proof: For any eigen-pair (λ, \mathbf{x}) , $A\mathbf{x} = \lambda\mathbf{x}$. Consider the p -th component of \mathbf{x} such that $|x_p| = \|\mathbf{x}\|_\infty$, we have

$$\sum_{j=1}^n a_{pj}x_j = \lambda_p x_p$$

or

$$(\lambda - a_{pp})x_p = \sum_{j=1, j \neq p}^n a_{pj}x_j.$$

From the expression above we get

$$|\lambda - a_{pp}| \leq \sum_{j=1, j \neq p}^n |a_{pj}| \left| \frac{x_j}{x_p} \right| \leq \sum_{j=1, j \neq p}^n |a_{pj}|, \quad \text{since } |x_j/x_p| \leq 1.$$

Thus λ is in the p -th Gershgorin circle and the first part of the theorem is complete.

The proof for the second part is based continuation theory that roots of a polynomial are continuous functions of the coefficients of the polynomials. The theorem is obviously true for the diagonal matrix. If the radius of the Gershgorin circles increase continuously as we change the zero off diagonal entries from 0 to a_{ij} , the eigenvalues will move among union of the Gershgorin circles but can not cross to the disjoint ones.

Example: Let A be the following matrix

$$A = \begin{pmatrix} -5 & -1 & 0 \\ -1 & 2 & -1/2 \\ 0 & -1 & 8 \end{pmatrix}$$

Use the Gershgorin theorem to roughly locate the eigenvalues.

The three Gershgorin circles are

- $R_1 : |z + 5| \leq 1.$
- $R_2 : |z - 2| \leq 1.5.$
- $R_3 : |z - 8| \leq 1.$

The do not interset with each other, so each circle has one eigenvalue. Since the matrix is a real matrix, and complex eigenvalues have to be in pair, we conclude that all the eigenvalues are real. Thus we get

- the dominant eigenvalue satisfies $7 \leq \lambda_1 \leq 9$,
- the least dominant eigenvalue satisfies $0.5 \leq \lambda_3 \leq 3.5$,
- the middle eigenvalue satisfies $-6 \leq \lambda_2 \leq -4$.

If we wish to find the middle eigenvalue λ_2 we should choose $\sigma = -5$. Even for the dominant eigenvalue, we would get faster convergence if we shift the matrix by taking $\sigma = 8$ and then apply the shifted power method.

5.5 How to find a few or all eigenvalues?

If we know an eigenpair $(\lambda_1, \mathbf{x}_1)$ of a matrix A , we can use the **deflation method** to reduce A to a one-dimensional lower matrix whose eigenvalues are the same as the rest eigenvalues of A . The process is follows. Assume that $\|\mathbf{x}_1\|_2 = 1$, we can form expand \mathbf{x}_1 to form an orthogonal basis of R^n : $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ with $\mathbf{x}_i^T \mathbf{x}_j = \delta_{ij}$. Note that $\delta_{ij} = 0$ if $i \neq j$ and $\delta_{ii} = 1$. Let $Q = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$, then Q is an orthogonal matrix ($Q^T Q = Q Q^T = I$). We can get

$$\begin{aligned} Q^T A Q &= Q^T [A \mathbf{x}_1, A \mathbf{x}_2, \dots, A \mathbf{x}_n] \\ &= Q^T [\lambda_1 \mathbf{x}_1, A \mathbf{x}_2, \dots, A \mathbf{x}_n] = \begin{pmatrix} \lambda_1 & * \\ 0 & A_1 \end{pmatrix} \end{aligned}$$

Thus the eigenvalues of A_1 are also those of A , but A_1 is a one-dimensional matrix compare with the original matrix A . The deflation method is only used if we wish to find a few eigenvalues.

To find all eigenvalues, the QR method for eigenvalues are often used. The idea of the QR method is first to reduce a matrix to a simple form (often upper Hessenberg matrix or tridiagonal matrix) using similarity transformation $S^{-1} A S$ so that the eigenvalues are unchanged. Since the inverses of an orthogonal matrix is its transpose, S is often chosen as orthogonal matrix. Orthogonal matrices also have better stability than other matrixes since $\|Q \mathbf{x}\|_2 = \|\mathbf{x}\|_2$ and $\|Q A\|_2 = \|A\|_2$.

Definition 5.5.1 Given a unit vector \mathbf{w} , $\|\mathbf{w}\|_2 = 1$, the Household matrix is defined as

$$P = I - 2\mathbf{w}\mathbf{w}^T. \quad (5.5.1)$$

It is a simple check that $P = P^T = P^{-1}$. Such a matrix sometimes is also called a reflection matrix or transformation.

Theorem 5.5.1 If $\|\mathbf{x}\|_2 = \|\mathbf{y}\|_2$, then there is a Household matrix P such that $P \mathbf{x} = \mathbf{y}$.

Proof: Assume that $P\mathbf{x} = \mathbf{y}$, then we have

$$\begin{aligned}(I - 2\mathbf{w}\mathbf{w}^T)\mathbf{x} &= \mathbf{y} \\ \mathbf{x} - \mathbf{y} &= 2\mathbf{w}(\mathbf{w}^T\mathbf{x}).\end{aligned}$$

Note that $2(\mathbf{w}^T\mathbf{x})$ is a number, thus \mathbf{w} is parallel to $\mathbf{x} - \mathbf{y}$. Since \mathbf{w} is also a unit vector in 2-norm, we conclude that $\mathbf{w} = (\mathbf{x} - \mathbf{y})/\|\mathbf{x} - \mathbf{y}\|_2$, then it is simple manipulation to show that $P\mathbf{x} = \mathbf{y}$.

Example: Find a Householder matrix P such that

$$P \begin{pmatrix} 3 \\ 0 \\ 4 \end{pmatrix} = \begin{pmatrix} \alpha \\ 0 \\ 0 \end{pmatrix}$$

Note that in this example, we need to find both α and P . Since the orthogonal transformation does not change the 2-norm, we should have

$$\begin{aligned}\sqrt{3^2 + 4^2} &= \alpha^2, & \implies \alpha &= \pm 5, \\ \mathbf{w} = (\mathbf{x} - \mathbf{y})/\|\mathbf{x} - \mathbf{y}\|_2 &= \frac{1}{\|\mathbf{x} - \mathbf{y}\|_2} \begin{pmatrix} 3 \pm \alpha \\ 0 - 0 \\ 4 - 0 \end{pmatrix}\end{aligned}$$

To avoid possible cancellation, we should choose the opposite sign, that $\alpha = -5$, and $\mathbf{w} = [8 \ 0 \ 4]^T/\sqrt{80}$.

5.5.1 The QR decomposition of a matrix A

Start from $A_0 = A$.

for $k = 0, 1, \dots$ until converge

$$A_k = Q_k R_k$$

$$A_{k+1} = R_k Q_k.$$

end

Theorem 5.5.2 *If $A \in R^{n \times n}$, and $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$, then*

- $A_{i+1} \sim A_i \sim A$, that is, all A_k have the same eigenvalues.

•

$$\lim_{k \rightarrow \infty} A_k = R_A = \begin{pmatrix} R_{11} & R_{12} & \cdots & R_{1p} \\ & R_{22} & \cdots & \cdots \\ & & \ddots & \vdots \\ & & & R_{pp} \end{pmatrix}$$

where R_A is a block upper triangular matrix whose diagonals R_{ii} is either a 1 by 1 or 2 by 2 matrix (corresponding to complex eigenvalues in pair).

Proof of the first part: $A_{k+1} = R_k Q_k = Q_k^T A_k R_k$.

Shifted QR method:

Start from $A_0 = A$.

for $k = 0, 1, \dots$ until converge

$$A_k - \sigma_k I = Q_k R_k$$

$$A_{k+1} = R_k Q_k + \sigma_k I.$$

end

Stop criteria: $\max_{3 \leq i \leq n, 1 \leq j \leq i-2} |a_{ij}| < tol$.

Double shifted QR method: If σ is chosen as a complex number, then we should use the Double shifted QR method.

Start from $A_0 = A$.

for $k = 0, 1, \dots$ until converge

$$A_k - \sigma_k I = Q_k R_k$$

$$A_{k+1} = R_k Q_k + \sigma_k I.$$

$$A_{k+1} - \bar{\sigma}_k I = Q_{k+1} R_{k+1}$$

$$A_{k+2} = R_{k+1} Q_{k+1} + \bar{\sigma}_k I.$$

end

QR method for finding all eigenvalues are quite expensive. To reduce the computational cost. Often we use the similarity transformation via Householder matrix first to reduce the original matrix to an upper Hessenberg matrix (tri-diagonal if the matrix is symmetric) first, then apply the QR method. This requires $n-2$ steps: $P_{n-1} P_{n-3} \dots P_2 P_1 A P_1 P_2 \dots P_{n-3} P_{n-2}$, where

$$P_1 = \begin{pmatrix} 1 & 0 \\ 0 & \bar{P}_1 \end{pmatrix}, \quad \bar{P}_1 \begin{pmatrix} a_{21} \\ a_{31} \\ \vdots \\ a_{n1} \end{pmatrix} = \begin{pmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

for example. The reason to choose P_1 to keep the first row of A unchanged when we multiply P_1 from the left is to ensure that the first columns of $P_1 A$ unchanged when we multiply P_1 from the right so that those zeros will remain.

Chapter 6

Least squares and SVD solutions

In this chapter, we discuss numerical method for solving arbitrary linear system of equations.

6.1 Least squares solutions

Consider $A\mathbf{x} = \mathbf{b}$, where $A \in R^{m \times n}$, $m \geq n$, and $rank(A) = n$. One motivation is the curve fitting example in which we have many observed data, but we find a simple polynomial to fit the data. Below are some examples:

$$\begin{pmatrix} 2 \\ 1 \\ 4 \\ -1 \end{pmatrix} \begin{pmatrix} x \end{pmatrix} = \begin{bmatrix} 0 \\ 0 \\ 4 \\ 2 \end{bmatrix}$$

From the first two equations, the solution should be $x = 0$; from the third equation, the solution should be $x = 1$; from the last equation, the solution should be $x = -2$; In other words, we can not find a single x that satisfy all the equations. The system of equations is called ***over-determined***. In general, there is no classical solution to an over-determined system of equation. We need to find the 'best' solution.

By the best solution we mean that to minimize the error in some norm. Since the residual is computable, one approach is to minimize the residual in 2-norm of all possible choices:

The solution \mathbf{x}^* is the 'best solution' in 2-norm if it satisfies

$$\|\mathbf{b} - A\mathbf{x}^*\|_2 = \min_{\mathbf{x} \in R^n} \|\mathbf{b} - A\mathbf{x}\|_2. \quad (6.1.1)$$

If we use different norms rather than 2-norm, it will leads to different algorithm and different applications. But 2-norm is the one that is used the most and it is the simplest one.

Note that, an equivalent definition is the following:

$$\|\mathbf{b} - A\mathbf{x}^*\|_2^2 = \min_{\mathbf{x} \in R^n} \|\mathbf{b} - A\mathbf{x}\|_2^2. \quad (6.1.2)$$

The above definition gives way to compute the 'best solution' as the global minimum of a multi-variable function of its component x_1, x_2, \dots, x_n . This can be seen from the following:

$$\phi(\mathbf{x}) = \|\mathbf{b} - A\mathbf{x}\|_2^2 = (\mathbf{b} - A\mathbf{x})^T (\mathbf{b} - A\mathbf{x}) = \mathbf{b}^T \mathbf{b} - \mathbf{b}^T A\mathbf{x} - \mathbf{x}^T A^T \mathbf{b} + \mathbf{x}^T A^T A\mathbf{x}.$$

Note that $\mathbf{b}^T A\mathbf{x} = \mathbf{x}^T A^T \mathbf{b}$, one can easily get the gradient $\nabla \phi(\mathbf{x})$ which is

$$\nabla \phi(\mathbf{x}) = 2 (A^T A\mathbf{x} - A^T \mathbf{b}).$$

Since the columns of A are linearly independent ($\text{rank}(A) = n$), we have $\mathbf{x}^T A^T A\mathbf{x} = \|A\mathbf{x}\|^2 > 0$ for any non-zero \mathbf{x} , and $A^T A$ is symmetric positive definite. The only critical point of $\phi(\mathbf{x})$ is the solution of the following **normal equation**

$$A^T A\mathbf{x} = A^T \mathbf{b} \quad (6.1.3)$$

whose unique solution is the least squares solution which minimizes the 2-norm of the residual in R^n space.

The normal equation approach not only provides a numerical method, but also shows that the least squares solution is unique under the condition $\text{rank}(A) = n$, that is, the columns of A are linearly independent.

A serious problem with the normal equation approach is the possible ill-conditioned system. Note that if $m = n$, then $\text{cond}_2(A^T) = (\text{cond}_2(A))^2$. A more accurate method is the QR method for the least squares solution. For the following least squares problem

$$\begin{pmatrix} R \\ 0 \end{pmatrix} \mathbf{x} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix}$$

We have

$$\|\mathbf{b} - A\mathbf{x}\|_2^2 = \|\mathbf{b}_1 - R\mathbf{x}\|_2^2 + \|\mathbf{b}_2\|_2^2$$

and

$$\min_{\mathbf{x} \in R^n} \|\mathbf{b} - A\mathbf{x}\|_2^2 = \min_{\mathbf{x} \in R^n} \|\mathbf{b}_1 - R\mathbf{x}\|_2^2 + \|\mathbf{b}_2\|_2^2 = \|\mathbf{b}_2\|_2^2$$

when $\mathbf{b}_1 - R\mathbf{x} = 0$ or $\mathbf{x} = R^{-1}\mathbf{b}_1$. Particularly, if R is an upper triangular matrix, we can use the backward substitution to solve the tri-diagonal system of equations efficiently.

In the QR method for the least squares, the idea is to reduce the original problem to the problem above using orthogonal, particularly, the Householder, transformation which

keeps the least squares solution unchanged. From the process of the QR algorithm, we can apply a sequence of Householder matrices that reduce A to an upper triangular form

$$P_n P_{n-1} \cdots P_1 A = \begin{pmatrix} R \\ 0 \end{pmatrix}$$

or $QA = \begin{pmatrix} R^T & 0 \end{pmatrix}^T$, then from $A\mathbf{x} = \mathbf{b}$, we get $QA\mathbf{x} = Q\mathbf{b}$, or

$$\begin{pmatrix} R \\ 0 \end{pmatrix} \mathbf{x} = \begin{pmatrix} \tilde{\mathbf{b}}_1 \\ \tilde{\mathbf{b}}_2 \end{pmatrix}$$

In practice, we can apply the Householder matrix directly to the augmented matrix $[A : \mathbf{b}]$ as in the Gaussian elimination method.

An example:

6.2 Singular value decomposition (SVD), and SVD solution of $A\mathbf{x} = \mathbf{b}$

Singular value decomposition can be used to solve $A\mathbf{x} = \mathbf{b}$ for arbitrary A and \mathbf{b} .

Theorem 6.2.1 *Given any matrix $A \in C^{m \times n}$, there are two orthogonal matrices $U \in C^{m \times m}$, $UU^H = U^H U = I$ and $V \in C^{n \times n}$, $VV^H = V^H V = I$ such that*

$$A = U \sum V^H, \quad \text{where} \quad \sum = \begin{pmatrix} \sigma_1 & & & & \\ & \sigma_2 & & & \\ & & \ddots & & \\ & & & \sigma_p & \\ & & & & \mathbf{0} \end{pmatrix}_{m \times n} \quad (6.2.1)$$

$\sigma_1, \sigma_2, \dots, \sigma_p > 0$ are called the singular values of A . Note that they are positive numbers, $p = \text{rank}(A)$. Furthermore

- $\sigma_i = \sqrt{\lambda_i(A^H A)} = \sqrt{\lambda_i(AA^H)}$, square root of non-zero eigenvalues of $A^H A$ or AA^H .
- $\|A\|_2 = \max_{1 \leq i \leq p} \sigma_i(A)$.

Note that we often arrange singular values according to $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p > 0$. The proof process also gives a way for such a decomposition (constrictive).

Proof: Let $\sigma_1 = \|A\|_2$, then there is an \mathbf{x} , $\|\mathbf{x}\|_2 = 1$ such that $A^H A \mathbf{x}_1 = \sigma_1^2 \mathbf{x}_1$. Let $\mathbf{y}_1 = A \mathbf{x}_1 / \sigma_1$, we have $\|\mathbf{y}_1\|_2 = \|A \mathbf{x}_1\|_2 / \sigma_1 = 1$.

Next we expand \mathbf{x}_1 to form an orthogonal basis in $R^{n \times n}$ to form an orthogonal matrix V

$$V = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] = [\mathbf{x}_1, V_1], \quad V^H V = V^H V = I.$$

We also expand \mathbf{y}_1 to form an orthogonal basis in $R^{m \times m}$ to form an orthogonal matrix U

$$U = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_m] = \begin{bmatrix} \mathbf{y}_1 & U_1 \end{bmatrix}, \quad U^H U = U^H U = I.$$

Then we have

$$U^H A V = \begin{pmatrix} \mathbf{y}_1^H \\ U_1^H \end{pmatrix} \begin{pmatrix} A \mathbf{x}_1 & A V_1 \end{pmatrix} = \begin{pmatrix} \sigma_1 & \mathbf{0} \\ \mathbf{0} & U_1^H A V_1 \end{pmatrix}$$

This is because $\mathbf{y}_1^H A \mathbf{x}_1 = \mathbf{x}_1^H A^H A \mathbf{x}_1 / \sigma_1 = \mathbf{x}_1^H \mathbf{x}_1 \sigma_1^2 / \sigma_1 = \sigma_1$; $U_1^H A \mathbf{x}_1 = \sigma_1 U_1^H \mathbf{y}_1 = \mathbf{0}$; $\mathbf{y}_1^H A V_1 = (A \mathbf{y}_1)^H V_1 = \mathbf{x}_1^H A^H A V_1 = \sigma_1^2 \mathbf{x}_1^H V_1 = \mathbf{0}$. Thus from the mathematics induction principle, we can continue this process to get the SVD decomposition.

Pseudo-inverse of a matrix A

From the SVD decomposition of a matrix A , we can literally find the 'inverse' of the matrix to get its pseudo-inverse

$$A^+ = V \Sigma^+ U^H, \quad \text{where} \quad \Sigma^+ = \begin{pmatrix} \frac{1}{\sigma_1} & & & & & \\ & \frac{1}{\sigma_2} & & & & \\ & & \ddots & & & \\ & & & & \frac{1}{\sigma_p} & \\ & & & & & \mathbf{0} \end{pmatrix}_{n \times m} \quad (6.2.2)$$

Particularly, if $m = n$ and $\det(A) \neq 0$, then $A^+ = A^{-1}$. The pseudo-inverse matrix A^+ of a matrix A has the following properties:

- $AA^+A = A$, note that $A^+A \neq I$.
- $A^+AA^+ = A^+$.
- $A^+A = (A + A)^H$. If $\text{rank}(A) = n$, then $A^+ = (A^H A)^{-1} A^H$.
- $AA^+ = (AA^+)^H$. If $\text{rank}(A) = m$, then $A^+ = A^H (AA^H)^{-1}$.

Solving $A\mathbf{x} = \mathbf{b}$ of arbitrary matrix A

The SVD solution of $A\mathbf{x} = \mathbf{b}$ is simply $\mathbf{x}^* = A^+\mathbf{b}$. It has the the following properties:

- If $m = n$ and $\det(A) \neq 0$, then $\mathbf{x}^* = A^+ \mathbf{b} = A^{-1} \mathbf{b}$.
- If $\text{rank}(A) = n$, then $\|A\mathbf{x}^* - \mathbf{b}\|_2 = \min_{\mathbf{x} \in \mathbb{R}^n} \|A\mathbf{x} - \mathbf{b}\|_2$, that is, \mathbf{x}^* is the least squares solution.
- If there is more than one classical solution, then \mathbf{x}^* is the one with minimal 2-norm, that is

$$\|\mathbf{x}^*\|_2 = \min_{\|A\mathbf{x} - \mathbf{b}\|_2 = \|A\mathbf{x} - \mathbf{b}\|_2} \{\|\mathbf{x}\|_2\}, \quad \|A\mathbf{x}^* - \mathbf{b}\|_2 = \min_{\mathbf{x} \in \mathbb{R}^n} \|A\mathbf{x} - \mathbf{b}\|_2$$