# MA 7007: Numerical Solution of Differential Equations I
# Iterative Methods for Sparse Linear Systems

Suh-Yuh Yang (楊肅煜)

Department of Mathematics, National Central University
Jhongli District, Taoyuan City 32001, Taiwan

E-mail: syyang@math.ncu.edu.tw
Website: http://www.math.ncu.eud.tw/∼syyang/

## Solving $Ax = b$: direct method vs. iterative method

- Direct methods for solving the matrix problem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$: e.g., Gaussian elimination, $LU$-decomposition.
  - large operation counts
  - hard to do on parallel machines
  - a solution will be found, and we know how long and how much memory it takes
- Iterative methods produce a sequence of vectors that ideally converges to the solution.
  - much smaller operation counts
  - a lot easier to implement on parallel computers
  - not as reliable or predicable (the number of iterations is not known in advance)
- For very large problems (especially in 3D), a direct solver is impractical. e.g., Gaussian elimination is an $O(m^3)$ algorithm.

# Centered difference scheme

As an example, we consider the Poisson equation with the Dirichlet BC:

$$\begin{cases} \nabla^2 u = g & \text{in } \Omega := (0,1) \times (0,1), \\ u = \varphi & \text{on } \partial\Omega. \end{cases}$$

Let $u_{ij}$ represent an approximation to $u(x_i, y_j)$ and $g_{ij} := g(x_i, y_j)$. For simplicity, we set $\Delta x = \Delta y = h$. Then we have

$$\frac{1}{h^2}(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{ij}) = g_{ij}.$$

We can rewrite the above equation as

$$u_{ij} = \frac{1}{4}(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}) - \frac{h^2}{4} g_{ij}.$$

## Jacobi and Gauss-Seidel iterative methods

- **Jacobi iteration:**

$$u_{ij}^{[k+1]} = \frac{1}{4}\left(u_{i-1,j}^{[k]} + u_{i+1,j}^{[k]} + u_{i,j-1}^{[k]} + u_{i,j+1}^{[k]}\right) - \frac{h^2}{4} g_{ij}, \quad k \geq 0.$$

  Jacobi iteration is about the worst possible iterative method. But it's very simple, and useful as a test for parallelization.
- **Gauss-Seidel iteration:** Jacobi iteration is rather slow to converge, and can be made faster by using the updated values of the solution as soon as they are available.

$$u_{ij}^{[k+1]} = \frac{1}{4}\left(u_{i-1,j}^{[k+1]} + u_{i+1,j}^{[k]} + u_{i,j-1}^{[k+1]} + u_{i,j+1}^{[k]}\right) - \frac{h^2}{4} g_{ij}, \quad k \geq 0.$$

- **Important features:**
  - The matrix $A$ is never stored;
  - The storage is optimal, essentially only the $m^2$ solution values are stored;
  - Each iteration requires $O(m^2)$ work.

## Matrix splitting methods

The Jacobi and Gauss-Seidel iterative methods for the linear system $Au = f$ can be analyzed by viewing them as based on a splitting of the matrix $A$ into

$$A = M - N,$$

where $M$ and $N$ are two $m \times m$ matrices. Then the linear system $Au = f$ can be written as

$$Mu - Nu = f \quad \Longrightarrow \quad Mu = Nu + f,$$

which suggests the iterative method

$$Mu^{[k+1]} = Nu^{[k]} + f, \quad k \geq 0.$$

The goal is to choose $M$ so that the following conditions hold:

- The sequence $\{u^{[k]}\}$ is easily computed.
- The sequence $\{u^{[k]}\}$ converges rapidly to the solution.

## Jacobi and Gauss-Seidel iterative methods

Consider the linear system $Au = f$. Let $A = D - L - U$, where $D = \text{diag}(A)$, $L$ is the negative of the strictly lower part of $A$, and $U$ is the negative of the strictly upper part of $A$. Then

- **Jacobi iteration:**

$$\begin{aligned} &M = D, \qquad N = L + U, \\ &Du^{[k+1]} = (L+U)u^{[k]} + f, \quad k \geq 0. \end{aligned}$$

- **Gauss-Seidel iteration:**

$$\begin{aligned} &M = D - L, \qquad N = U, \\ &(D-L)u^{[k+1]} = Uu^{[k]} + f, \quad k \geq 0. \end{aligned}$$

## Convergence analysis

To analyze these methods, we derive from the update formula

$$\begin{aligned} u^{[k+1]} &= M^{-1}Nu^{[k]} + M^{-1}f, \\ &= Gu^{[k]} + c, \end{aligned}$$

where $G := M^{-1}N$ is the *iteration matrix* and $c := M^{-1}f$.

Let $u^*$ represent the true solution to the linear system $Au = f$. Then $u^* = Gu^* + c$. We call $u^*$ a fixed point or an equilibrium of $G(\cdot) + c$. If $e^{[k]} := u^{[k]} - u^*$ represents the error at $k$th step, then we have

$$e^{[k+1]} = Ge^{[k]}.$$

Repeating this process, we obtain

$$e^{[k]} = G^k e^{[0]},$$

From this we can see that the method will converge from any initial guess $u^{[0]}$ if $G^k \to 0$(an $m \times m$ matrix of zeros) as $k \to \infty$.

# A necessary and sufficient condition

For simplicity, assume that $G$ is a diagonalizable matrix, so that we can write

$$G = R\Gamma R^{-1} \qquad \Longleftarrow \quad \boxed{R^{-1}GR = \Gamma} \quad \Longrightarrow \qquad GR = R\Gamma,$$

where $R$ is the matrix of right eigenvectors of $G$ and $\Gamma$ is a diagonal matrix of eigenvalues $\gamma_1, \cdots, \gamma_m$. Then

$$G^k = R\Gamma^k R^{-1},$$

where $\Gamma^k = \mathrm{diag}(\gamma_1^k, \cdots, \gamma_m^k)$. One observe that the $G^k \to 0$ as $k \to 0$ if $|\gamma_p| < 1$ for all $p = 1, 2, \cdots, m$. This is, if $\rho(G) < 1$, then $G^k \to 0$ as $k \to 0$, where $\rho(G)$ is the spectral radius of $G$. In fact, this is a necessary and sufficient condition:

**Theorem:** The iteration formula

$$u^{[k+1]} = Gu^{[k]} + c$$

converges for any initial guess $u^{[0]}$ if and only if the spectral radius of $G$ be less than 1, i.e., $\rho(G) < 1$.

## Spectral radius

- The spectral radius of $A$ is defined by

$$\rho(A) = \max\{|\lambda| : \det(A - \lambda I) = 0\}.$$

  Thus, $\rho(A)$ is the smallest number such that a circle with that radius centered at 0 in the complex plane will contain all the eigenvalues of $A$.

- **Theorem on Spectral Radius:** The spectral radius function satisfies

$$\rho(A) = \inf_{\|\cdot\|} \|A\|,$$

  in which the infimum is taken over all subordinate matrix norms.

- **Corollary on Spectral Radius:**
  - $\rho(A) \leq \|A\|$ for any subordinate matrix norm.
  - If $\rho(A) < 1$ then $\|A\| < 1$ for some subordinate matrix norm.

## Proof of the Theorem ($\Leftarrow$)

Suppose that $\rho(G) < 1$. There is a subordinate matrix norm such that $\|G\| < 1$. From the iteration formula, we have

$$u^{[1]} = Gu^{[0]} + c, \quad u^{[2]} = G^2 u^{[0]} + Gc + c, \quad \cdots, \quad u^{[k]} = G^k u^{[0]} + \sum_{j=0}^{k-1} G^j c.$$

Using the matrix norm and corresponding vector norm, we obtain

$$\|G^k u^{[0]}\| \le \|G^k\| \|u^{[0]}\| \le \|G\|^k \|u^{[0]}\| \to 0 \quad \text{as} \quad k \to \infty.$$

Moreover, by Neumann series we have

$$\sum_{j=0}^{\infty} G^j c = (I - G)^{-1} c.$$

Finally, by letting $k \to \infty$, we obtain

$$\lim_{k\to\infty} u^{[k]} = \lim_{k\to\infty} \left( G^k u^{[0]} + \sum_{j=0}^{k-1} G^j c \right) = (I - G)^{-1} c.$$

## Proof of the Theorem ($\Rightarrow$)

Suppose that $\rho(G) \geq 1$. Select $v$ and $\lambda$ so that $Gv = \lambda v$, where $|\lambda| \geq 1$ and $v \neq 0$. Recall that $u^{[k]} = G^k u^{[0]} + \sum_{j=0}^{k-1} G^j c$. Let $c = v$ and $u^{[0]} = 0$. Then we have

$$u^{[k]} = \sum_{j=0}^{k-1} G^j v = \sum_{j=0}^{k-1} \lambda^j v.$$

- If $\lambda = 1$, $u^{[k]} = kv$, this diverges as $k \to \infty$.
- If $\lambda \neq 1$, $u^{[k]} = (\lambda^k - 1)(\lambda - 1)^{-1} v$, this diverges as $k \to \infty$ and this diverges also because $\lim_{k\to\infty} \lambda^k$ does not exist.

For both cases, $\{u^{[k]}\}$ diverges, a contradiction! Therefore, $\rho(G) < 1$.

# Analysis of Jacobi method

Recall the Jacobi method

$$Du^{[k+1]} = (L+U)u^{[k]} + f = (D-A)u^{[k]} + f.$$

We have $G = D^{-1}(D-A) = I - D^{-1}A$ and $c = D^{-1}f$.

As a simple example, we apply this method to the linear system arising from the centered difference approximation to $u''(x) = g(x)$ with Dirichlet BC,

$$u''(x) = g(x), \quad 0 < x < 1, \quad u(0) = \alpha \text{ and } u(1) = \beta.$$

Then the linear system $Au = f$ is

$$\frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{bmatrix}\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_{m-1} \\ U_m \end{bmatrix} = \begin{bmatrix} g(x_1) - \alpha/h^2 \\ g(x_2) \\ g(x_3) \\ \vdots \\ g(x_{m-1}) \\ g(x_m) - \beta/h^2 \end{bmatrix}.$$

# Analysis of Jacobi method (continued)

The iteration matrix is

$$G = I - D^{-1}A = I + \frac{h^2}{2}A.$$

The eigenvalues of $G$ are

$$\gamma_p = 1 + \frac{h^2}{2}\lambda_p = 1 + \frac{h^2}{2}\left(\frac{2}{h^2}(\cos(p\pi h) - 1)\right) = \cos(p\pi h), \quad p = 1, 2, \cdots, m.$$

So the spectral radius of $G$ is

$$\rho(G) = |\gamma_1| = \cos(\pi h) = \cos\left(\frac{\pi}{m+1}\right) < 1$$

and the Jacobi method converges for any initial guess $u^{[0]}$ for the linear system arising from the centered difference approximation for the 1-D example.

# Analysis of Gauss-Seidel method

Recall the Gauss-Seidel method

$$(D-L)u^{[k+1]} = Uu^{[k]} + f.$$

We have $G = (D-L)^{-1}U$ and $c = (D-L)^{-1}f$.

Let $\lambda$ be a nonzero eigenvalue of $G$ and $v := (v_1, v_2, \cdots, v_m)^\top \neq 0$ be a corresponding eigenvector. Then we have

$$\begin{aligned}
&(D-L)^{-1}Uv = \lambda v \implies Uv = \lambda(D-L)v \implies \lambda Dv = \lambda Lv + Uv \\
&\implies \lambda v_i = \frac{-1}{2}\Big(-\lambda v_{i-1} - v_{i+1}\Big) = \frac{1}{2}\Big(\lambda v_{i-1} + v_{i+1}\Big), 1 \le i \le m,\ v_0 = v_{m+1} = 0.
\end{aligned}$$

Now we set $v_i = \lambda^{i/2} u_i$ for $1 \le i \le m$. Then

$$\lambda^{\frac{i}{2}+1} u_i = \frac{1}{2}\Big(\lambda^{\frac{i-1}{2}+1} u_{i-1} + \lambda^{\frac{i+1}{2}} u_{i+1}\Big).$$

Multiplying $\lambda^{-\frac{i+1}{2}}$ leads to

$$\lambda^{\frac{1}{2}} u_i = \frac{1}{2}\Big(u_{i-1} + u_{i+1}\Big).$$

# Analysis of Gauss-Seidel method (continued)

$$\lambda^{\frac{1}{2}} u_i = \frac{1}{2}\left(u_{i-1} + u_{i+1}\right) \implies \lambda^{\frac{1}{2}}(-2)u_i = -(u_{i-1} + u_{i+1})$$
$$\implies \lambda^{\frac{1}{2}} Du = (L+U)u$$
$$\implies \lambda^{\frac{1}{2}} u = D^{-1}(L+U)u = D^{-1}(D-A)u = (I - D^{-1}A)u.$$

We have already proved that $u = (u_1, u_2, \cdots, u_m)^\top$ is an eigenvector associated with the eigenvalue $\lambda^{\frac{1}{2}}$ of the iteration matrix $I - D^{-1}A$ of the Jacobi method. Moreover, one can check that the inverse process works as well. From the above discussion, we can conclude that the eigenvalues $\lambda_p$ of the iteration matrix $G = (D-L)^{-1}U$ of the Gauss-Seidel method should be

$$\lambda_p = \cos^2(p\pi h), \quad p = 1, 2, \cdots, m,$$

where $\cos(p\pi h)$, $p = 1, 2, \cdots, m$, are the eigenvalues of the iteration matrix $I - D^{-1}A$ of the Jacobi method. It leads to

$$\rho((D-L)^{-1}U) = \cos^2(\pi h) = \cos^2\left(\frac{\pi}{m+1}\right) < 1.$$

Thus, the Gauss-Seidel method converges for any initial guess $u^{[0]}$ for the linear system arising from the 1-D example.

# Successive over-relaxation (SOR) method

The Gauss-Seidel moves $u_i$ in right direction but is far too conservative in the amount it allows $u_i$ to move.

**Successive Over-Relaxation (SOR):** Compute Gauss-Seidel approximation and then go further:

$$u_i^{\mathrm{GS}} = \frac{1}{2}\left(u_{i-1}^{[k+1]} + u_{i+1}^{[k]} - h^2 f_i\right) \quad \text{and} \quad u_i^{[k+1]} = \omega u_i^{\mathrm{GS}} + (1-\omega)u_i^{[k]},$$

can be combined to yield,

$$u_i^{[k+1]} = \frac{\omega}{2}\left(u_{i-1}^{[k+1]} + u_{i+1}^{[k]} - h^2 f_i\right) + (1-\omega)u_i^{[k]}.$$

**Remarks:**

- $0 < \omega < 1$: under-relaxation methods and can be used to obtain convergence of some systems that are not convergent by the GS method.
- $1 < \omega$: over-relaxation methods, which are used to accelerate the convergence for systems that are convergent by the GS method.
- Optimal $\omega$ for the Poisson problem:

$$\omega_{\mathrm{opt}} = \frac{2}{1+\sin(\pi h)} \approx 2 - 2\pi h.$$

# A general theory for SOR

For a general system $Au = f$ with $A = D - L - U$, where $D = \text{diag}(A)$, $L$ is the negative of the strictly lower part of $A$, and $U$ is the negative of the strictly upper part of $A$. Then

**Successive Over-Relaxation (SOR):**

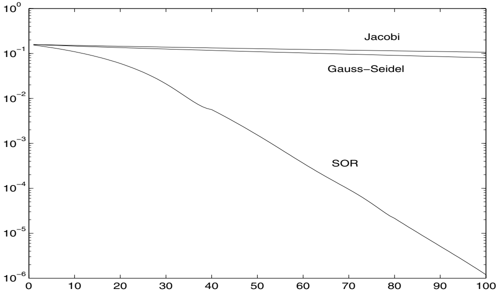
$$Mu^{[k+1]} = Nu^{[k]} + f,$$

where

$$M = \frac{1}{\omega}(D - \omega L), \qquad N = \frac{1}{\omega}((1-\omega)D + \omega U).$$

A theorem of SOR method states that if $A$ is symmetric and positive definite (SPD) and $D - \omega L$ is nonsingular, then SOR method converges for all $0 < \omega < 2$.

# Comparison




Errors versus $k$ for Jacobi, Gauss-Seidel and SOR methods.

(Two-point BVP: $u''(x) = f(x)$, SOR with optimal $\omega_{\text{opt}}$)

## Recall some properties of SPD

- Let $A \in \mathbb{C}^{m\times m}$ be a square matrix and $x, y \in \mathbb{C}^m$. Define $A^* := \overline{A}^\top$, $x^* := \overline{x}^\top$ and $(x, y) := y^* x \in \mathbb{C}$. Then $(Ax, x) = x^* A x$ is called a quadratic form.
- Definition: Let $A \in \mathbb{C}^{m\times m}$. $A$ is positive definite $\Longleftrightarrow (Ax, x) > 0, \forall\, 0 \neq x \in \mathbb{C}^m$.
- Note 1: $A = A^* \Longleftrightarrow (Ax, x) \in \mathbb{R}, \forall\, x \in \mathbb{C}^m$.
- Note 2: If $A \in \mathbb{C}^{m\times m}$ is positive definite, then $A = A^*$. (by Note 1)
- Note 3: Let $A \in \mathbb{R}^{m\times m}$. $A$ is positive definite $\Longleftrightarrow A = A^\top$ and $(Ax, x) > 0, \forall\, 0 \neq x \in \mathbb{R}^m$.
- Note 4: Let $A \in \mathbb{C}^{m\times m}$ and $A = A^*$. Then $A$ is positive definite $\Longleftrightarrow$ all of its eigenvalues are real and positive.

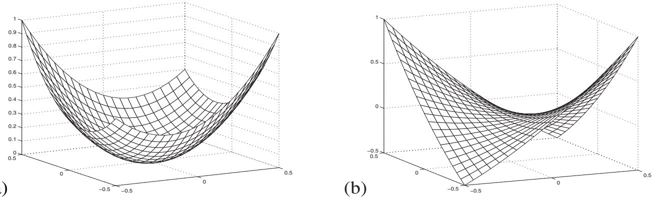
# SPD linear systems

Consider the linear system $Au = f$, where $A \in \mathbb{R}^{m \times m}$ is symmetric (S) and positive definite (PD), or negative definite since negating the system then gives an SPD matrix. Define $\phi : \mathbb{R}^m \to \mathbb{R}$ by

$$\phi(u) = \frac{1}{2} u^\top A u - u^\top f.$$

- Problem (1): Find $u^* \in \mathbb{R}^m$ such that $\phi(u^*) = \min_{u \in \mathbb{R}^m} \phi(u)$.
- Problem (2): Find $u^* \in \mathbb{R}^m$ such that $Au^* = f$.

**Note:** $\exists!$ solution $u^* \in \mathbb{R}^m$ such that $Au^* = f$, since $A$ is SPD.




(a) (b)

$\phi(u)$ for $m = 2$: (a) $A$ is SPD; (b) $A$ is S but indefinite.

## Proof of Problem (1) $\Longleftrightarrow$ Problem (2)

- Problem (1) $\Longrightarrow$ Problem (2):

  Let $u^* \in \mathbb{R}^m$ be such that $\phi(u^*) = \min_{u\in\mathbb{R}^m} \phi(u)$. Given $0 \neq u \in \mathbb{R}^m$. Then

$$\begin{aligned} g(\varepsilon) &:= \phi(u^* + \varepsilon u) = \frac{1}{2}(u^* + \varepsilon u)\cdot A(u^* + \varepsilon u) - f\cdot(u^* + \varepsilon u) \\ &= \frac{1}{2}u^*\cdot Au^* + \frac{1}{2}\varepsilon u^*\cdot Au + \frac{1}{2}\varepsilon u\cdot Au^* + \frac{1}{2}\varepsilon^2 u\cdot Au - f\cdot u^* - \varepsilon f\cdot u \\ &= \frac{1}{2}\varepsilon^2 u\cdot Au + \varepsilon u\cdot Au^* - \varepsilon f\cdot u + \frac{1}{2}u^*\cdot Au^* - f\cdot u^*, \end{aligned}$$

  where we use $u^*\cdot Au = (u^*, Au) = (A^\top u^*, u) = (Au^*, u) = (u, Au^*) = u\cdot Au^*$.

  $\therefore g$ is a quadratic polynomial in $\varepsilon$ with leading coefficient $\frac{1}{2}u\cdot Au > 0$.

  $\because g(0) = \phi(u^*) = \min_{u\in\mathbb{R}^m} \phi(u)$. $\quad\therefore g'(0) = 0$ (by Fermat's Theorem).

  $\therefore 0 = g'(0) = \left(\varepsilon u\cdot Au + u\cdot Au^* - f\cdot u\right)\Big|_{\varepsilon=0} = u\cdot(Au^* - f), \forall\, 0 \neq u \in \mathbb{R}^m$.

  $\therefore Au^* = f$.

# Proof of Problem (1) $\Longleftrightarrow$ Problem (2) (continued)

- Problem (2) $\Longrightarrow$ Problem (1):

  Assume that $Au^* = f$. Let $u \in \mathbb{R}^m$. Define $w := u - u^*$. Then $u = w + u^*$.

  We have

$$\begin{aligned}
\phi(u) &= \frac{1}{2}u \cdot Au - f \cdot u = \frac{1}{2}(w+u^*) \cdot A(w+u^*) - f \cdot (w+u^*) \\
&= \frac{1}{2}w \cdot Aw + w \cdot Au^* + \frac{1}{2}u^* \cdot Au^* - f \cdot w - f \cdot u^* \\
&= \frac{1}{2}w \cdot Aw + w \cdot Au^* - f \cdot w + \phi(u^*) \\
&\geq w \cdot Au^* - f \cdot w + \phi(u^*) \qquad (\because A \text{ is SPD } \therefore \frac{1}{2}w \cdot Aw \geq 0) \\
&= w \cdot f - f \cdot w + \phi(u^*) = \phi(u^*).
\end{aligned}$$

  $\therefore \phi(u^*) = \min_{u \in \mathbb{R}^m} \phi(u)$.

## Minimization algorithms

Given an initial approximation $u^{[0]} \in \mathbb{R}^m$ of the exact solution $u^*$. Find $u^{[k]} \in \mathbb{R}^m$, $k = 1, 2, \ldots$ of the form

$$u^{[k+1]} = u^{[k]} + \alpha_k d^{[k]}, \; k = 0, 1, \ldots,$$

where $d^{[k]} \in \mathbb{R}^m$ is the search direction, $\alpha_k > 0$ is the step size (length). We will focus on two methods:

- The method of steepest descent (also called the gradient method).
- The conjugate-gradient method.

# Some notation

Let $\phi : \mathbb{R}^m \to \mathbb{R}$ be a smooth function and $u \in \mathbb{R}^m$.

- Gradient of $\phi$ at $u = \phi'(u) := \nabla\phi(u) := \left(\frac{\partial\phi}{\partial u_1}(u), \frac{\partial\phi}{\partial u_2}(u), \cdots, \frac{\partial\phi}{\partial u_m}(u)\right)^\top$.
- Hessian of $\phi$ at $u$,

$$
\begin{aligned}
\phi''(u) &= \begin{bmatrix} \frac{\partial^2\phi}{\partial u_1^2}(u) & \frac{\partial^2\phi}{\partial u_1 \partial u_2}(u) & \cdots & \frac{\partial^2\phi}{\partial u_1 \partial u_m}(u) \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial^2\phi}{\partial u_m \partial u_1}(u) & \frac{\partial^2\phi}{\partial u_m \partial u_2}(u) & \cdots & \frac{\partial^2\phi}{\partial u_m^2}(u) \end{bmatrix}_{m\times m} \\
&= \left(\nabla\frac{\partial\phi}{\partial u_1}(u), \cdots, \nabla\frac{\partial\phi}{\partial u_m}(u)\right) \\
&:= \nabla\left(\frac{\partial\phi}{\partial u_1}(u), \cdots, \frac{\partial\phi}{\partial u_m}(u)\right) \\
&= \nabla\left(\phi'(u)^\top\right) \\
&= \nabla\left(\nabla\phi(u)^\top\right).
\end{aligned}
$$

# Example

Assume that $A \in \mathbb{R}^{m\times m}$ is a symmetric matrix, $f \in \mathbb{R}^m$ is a given vector, and $\phi : \mathbb{R}^m \to \mathbb{R}$ is defined by $\phi(u) := \frac{1}{2}u^\top Au - u^\top f$.

Then we can prove that $\forall\, u \in \mathbb{R}^m$,

- $\phi'(u) = Au - f$;
- $\phi''(u) = A$,

by using the following identities:

- $u \cdot Au = u_1(A_{1\cdot} \cdot u) + u_2(A_{2\cdot} \cdot u) + \cdots + u_m(A_{m\cdot} \cdot u)$.
- $\phi''(u) = \nabla(\nabla\phi(u)^\top) = \nabla((Au - f)^\top) = \nabla(A_{1\cdot} \cdot u - f_1, \cdots, A_{m\cdot} \cdot u - f_m)$.

## Taylor's expansion of a smooth function $\phi$ at $u^{[k]}$

Recall that we want to find $u^* \in \mathbb{R}^m$ such that $\phi(u^*) = \min_{u\in\mathbb{R}^m} \phi(u)$ by using the minimization algorithm: $u^{[k+1]} = u^{[k]} + \alpha_k d^{[k]}$, $k \geq 0$, where $\phi$ is a smooth function given by $\phi(u) := \frac{1}{2}u^\top A u - u^\top f$. To determine $\alpha_k$ and $d^{[k]}$, by Taylor's expansion, we have

$$\begin{aligned} \phi(u^{[k+1]}) &= \phi(u^{[k]}) + \nabla\phi(u^{[k]}) \cdot (u^{[k+1]} - u^{[k]}) \\ &\quad + (u^{[k+1]} - u^{[k]}) \cdot \frac{\phi''(\eta)}{2!}(u^{[k+1]} - u^{[k]}), \text{ for some } \eta \in \overline{u^{[k]}u^{[k+1]}} \\ &= \phi(u^{[k]}) + \alpha_k \phi'(u^{[k]}) \cdot d^{[k]} + \frac{\alpha_k^2}{2!} d^{[k]} \cdot \phi''(\eta) d^{[k]}. \end{aligned}$$

$\therefore$ $\phi(u^{[k+1]}) = \phi(u^{[k]}) + \alpha_k \phi'(u^{[k]}) \cdot d^{[k]} + O(\alpha_k^2)$, provided the entries in $\phi''(\eta)$ are bounded in a neighborhood containing $\overline{u^{[k]}u^{[k+1]}}$.

$\therefore$ If $\phi'(u^{[k]}) \cdot d^{[k]} < 0$ and $\alpha_k > 0$ is sufficiently small, then $\phi(u^{[k+1]}) < \phi(u^{[k]})$. In this case, we call $d^{[k]}$ a descent direction.

# The method of steepest descent

Note that $\phi(u) := \frac{1}{2}u^\top Au - u^\top f$ and $A$ is SPD.

If we choose $d^{[k]} = -\phi'(u^{[k]}) = -(Au^{[k]} - f)$ and if $\phi'(u^{[k]}) \neq 0$,

then we have $\phi'(u^{[k]}) \cdot d^{[k]} = -\|\phi'(u^{[k]})\|_2^2 < 0$.

We obtain the so-called steepest descent method or the gradient method.

**Note:** If $\phi'(u^{[k]}) = 0$ then $Au^{[k]} - f = 0 \Longrightarrow Au^{[k]} = f \Longrightarrow u^{[k]}$ is the exact solution.

# How to choose $\alpha_k > 0$ in the method of steepest descent?

Determine optimal $\alpha_k$ such that $\phi(u^{[k]} + \alpha_k d^{[k]}) = \min_{\alpha\in\mathbb{R}} \phi(u^{[k]} + \alpha d^{[k]})$.

Notice that $\phi(u^{[k]} + \alpha d^{[k]})$ can be viewed as a quadratic function in $\alpha$ with positive leading coefficient.

If $\alpha_k$ is optimal, then $\dfrac{d}{d\alpha}\phi(u^{[k]} + \alpha d^{[k]})\Big|_{\alpha=\alpha_k} = 0.$

$$\therefore \phi'(u^{[k]} + \alpha d^{[k]}) \cdot d^{[k]}\Big|_{\alpha=\alpha_k} = 0.$$

$$\therefore \phi'(u^{[k]} + \alpha_k d^{[k]}) \cdot d^{[k]} = 0.$$

$$\begin{aligned}\Longrightarrow 0 \quad &= \quad \phi'(u^{[k]} + \alpha_k d^{[k]}) \cdot d^{[k]} = \Big(A(u^{[k]} + \alpha_k d^{[k]}) - f\Big) \cdot d^{[k]} \\ &= \quad (Au^{[k]} - f) \cdot d^{[k]} + \alpha_k d^{[k]} \cdot Ad^{[k]}.\end{aligned}$$

$$\therefore \alpha_k = -\frac{(Au^{[k]} - f) \cdot d^{[k]}}{d^{[k]} \cdot Ad^{[k]}} = \frac{d^{[k]} \cdot d^{[k]}}{d^{[k]} \cdot Ad^{[k]}},$$

provided $d^{[k]} = -\phi'(u^{[k]}) = -(Au^{[k]} - f) \neq 0$.

$\because A$ is SPD. $\quad \therefore d^{[k]} \cdot Ad^{[k]} > 0$, provided $d^{[k]} = -\phi'(u^{[k]}) = -(Au^{[k]} - f) \neq 0$.

$\therefore \alpha_k > 0$, provided $d^{[k]} = -\phi'(u^{[k]}) = -(Au^{[k]} - f) \neq 0$.

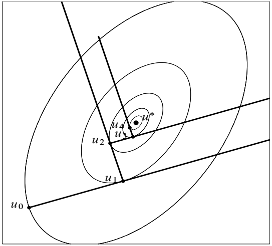
# The method of steepest descent with optimal step length $\alpha_k$

The steepest descent algorithm takes the form, for $k = 0, 1, 2, \ldots$

$$\begin{aligned} u^{[k+1]} &= u^{[k]} + \alpha_k d^{[k]}, \\ \alpha_k &= \frac{d^{[k]} \cdot d^{[k]}}{d^{[k]} \cdot A d^{[k]}}, \end{aligned}$$

where

$$d^{[k]} = -(Au^{[k]} - f).$$




$m = 2$: the concentric ellipses are level sets of $\phi(u)$.
($\because$ $A$ is SPD, the level sets of $\phi$ are always ellipses)

# Remarks

- It appears that in each iteration we must do two matrix-vector multiples, $Au^{[k]}$ to compute $d^{[k]}$ and then $Ad^{[k]}$ to compute $\alpha_k$. However, note that

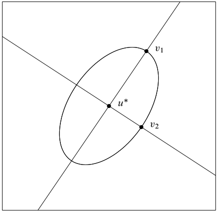
$$\begin{aligned} d^{[k+1]} &= f - Au^{[k+1]} \\ &= f - A(u^{[k]} + \alpha_k d^{[k]}) \\ &= d^{[k]} - \alpha_k A d^{[k]}. \end{aligned}$$

  So once we have computed $Ad^{[k]}$ as needed for $\alpha_k$, we can also use this result to compute $d^{[k+1]}$.
- Since $d^{[k+1]} = d^{[k]} - \alpha_k A d^{[k]}$, we have

$$\begin{aligned} d^{[k+1]} \cdot d^{[k]} &= d^{[k]} \cdot d^{[k]} - \alpha_k A d^{[k]} \cdot d^{[k]} \\ &= d^{[k]} \cdot d^{[k]} - \frac{d^{[k]} \cdot d^{[k]}}{d^{[k]} \cdot A d^{[k]}} A d^{[k]} \cdot d^{[k]} \\ &= 0. \end{aligned}$$

# The major and minor axes of the elliptical level set of $\phi(u)$

Assume that $A$ is a SPD $2 \times 2$ matrix. Let $v_1$ and $v_2$ be the points that the gradient $\nabla\phi(v_j)$ lies in the direction that connects $v_j$ to the center $u^*$, see the figure below.

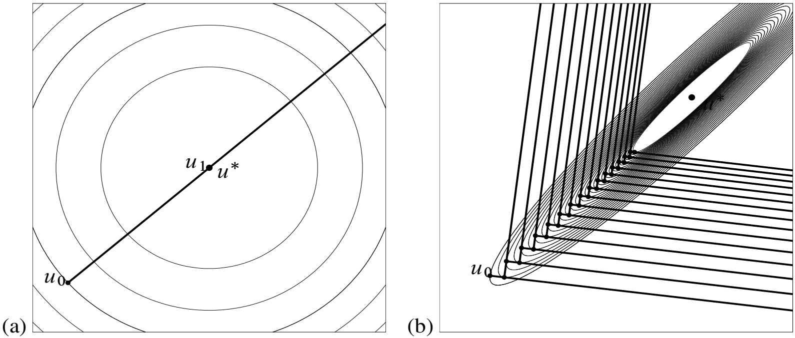



Then for $j = 1, 2$, $\nabla\phi(v_j) = Av_j - f = \lambda_j(v_j - u^*)$, for some $\lambda_j \in \mathbb{R}$.

Since $f = Au^*$, this gives $Av_j - f = A(v_j - u^*) = \lambda_j(v_j - u^*)$.

Hence, each direction $v_j - u^*$ is an eigenvector of $A$ and $\lambda_j$ is an eigenvalue.

## Level sets of $\phi(u)$: $m = 2$




(a) level sets of $\phi(u)$ are circular; (b) level sets of $\phi(u)$ are far from circular.

## The length of the major and minor axes

The length of the major and minor axes is related to the magnitude of $\lambda_1$ and $\lambda_2$. Suppose that $v_1$ and $v_2$ lie on the level set along which $\phi(u) = 1$. Then we have

$$\phi(v_j) = \frac{1}{2} v_j^\top A v_j - v_j^\top f = \frac{1}{2} v_j^\top A v_j - v_j^\top A u^* = 1, \quad j = 1, 2.$$

Taking the inner product of $A(v_j - u^*) = \lambda_j (v_j - u^*)$ with $v_j - u^*$ and combining with $\frac{1}{2} v_j^\top A v_j - v_j^\top A u^* = 1$, we have

$$\|v_j - u^*\|_2^2 = \frac{2 + u^{*\top} A u^*}{\lambda_j}, \quad j = 1, 2.$$

Hence the ratio of the length of the major axis to the length of the minor axis is

$$\frac{\|v_1 - u^*\|_2}{\|v_2 - u^*\|_2} = \sqrt{\frac{\lambda_2}{\lambda_1}} = \sqrt{\kappa_2(A)}.$$

where $\lambda_1 \leq \lambda_2$ and $\kappa_2(A)$ is the 2-norm condition number of $A$.

## The 2-norm condition number of $A$: $\kappa_2(A)$

Let $A \in \mathbb{R}^{m\times m}$ be a SPD matrix.

Let $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_m$ be the eigenvalues of $A$.

Then $0 < \dfrac{1}{\lambda_m} \le \dfrac{1}{\lambda_{m-1}} \le \cdots \le \dfrac{1}{\lambda_1}$ are the eigenvalues of $A^{-1}$.

Let $\rho(A)$ denote the spectral radius of $A$, i.e., the maximum size of the eigenvalues of $A$. That is, $\rho(A) = \max\limits_j |\lambda_j|$.

$$
\begin{aligned}
\kappa_2(A) \quad &:= \quad \|A\|_2\|A^{-1}\|_2 = \sqrt{\rho(A^*A)}\sqrt{\rho((A^{-1})^*A^{-1})} \\
&= \quad \sqrt{\rho(A^\top A)}\sqrt{\rho((A^{-1})^\top A^{-1})} = \sqrt{\rho(A^2)}\sqrt{\rho((A^{-1})^2)} \\
&= \quad \sqrt{\lambda_m^2}\sqrt{\frac{1}{\lambda_1^2}} = \frac{\lambda_m}{\lambda_1} = \frac{\lambda_{\max}}{\lambda_{\min}}.
\end{aligned}
$$

## The $A$-conjugate search direction

The steepest descent direction can be generalized by choosing a search direction $p^{[k]}$ in the $(k+1)$th iteration that might be different from the direction $d^{[k]}$.

We set

$$u^{[k+1]} = u^{[k]} + \alpha_k p^{[k]},$$

where $\alpha_k$ is chosen to minimize $\phi(u^{[k]} + \alpha_k p^{[k]})$ over all scalar $\alpha$. In other words, we perform a line search along the line through $u^{[k]}$ in the direction $p^{[k]}$ and find the minimum of $\phi$ on this line. The solution is at the point where the line is tangent to a contour line of $\phi$, and

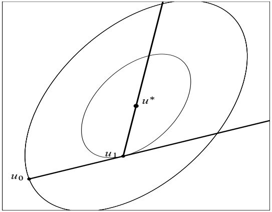
$$\alpha_k = \frac{d^{[k]} \cdot p^{[k]}}{p^{[k]} \cdot Ap^{[k]}}.$$

## The $A$-conjugate search direction (continued)

- A bad choice of search direction $p^{[k]}$ would be a direction orthogonal to $d^{[k]}$, since then $p^{[k]}$ would be tangent to the level set of $\phi$ at $u^{[k]}$, $\phi(u)$ could only increase along this line, and so $u^{[k+1]} = u^{[k]}$. Note that in this case

$$\alpha_k = \frac{d^{[k]} \cdot p^{[k]}}{p^{[k]} \cdot Ap^{[k]}} = \frac{0}{p^{[k]} \cdot Ap^{[k]}} = 0.$$

- But as long as $p^{[k]} \cdot d^{[k]} \neq 0$, the new point $u^{[k+1]}$ will be different from $u^{[k]}$ and will satisfy $\phi(u^{[k+1]}) < \phi(u^{[k]})$.




The two search directions used are $A$-conjugate

# The $A$-conjugate search direction (continued)

Once we obtain $u^{[1]}$ by the formulas

$$u^{[k+1]} = u^{[k]} + \alpha_k p^{[k]} \quad \text{and} \quad \alpha_k = \frac{d^{[k]} \cdot p^{[k]}}{p^{[k]} \cdot Ap^{[k]}},$$

we choose the next search direction $p^{[1]}$ to be a vector satisfying

$$p^{[1]} \cdot Ap^{[0]} = 0.$$

Two vectors $p^{[0]}$ and $p^{[1]}$ that satisfy the above equation are said to be $A$-conjugate.

- For any SPD matrix $A$, the vectors $u$ and $v$ are $A$-conjugate if the inner product of $u$ with $Av$ is zero, i.e., $u \cdot Av = 0$.
- If $A = I$, this just means the vectors are orthogonal, and $A$-conjugate is a natural generalization of the notion of orthogonality.

# The conjugate-gradient algorithm

Given $u^{[0]} \in \mathbb{R}^m$,

$$p^{[0]} := d^{[0]} := -(Au^{[0]} - f).$$

Find $u^{[1]}$ and $p^{[1]}$, $u^{[2]}$ and $p^{[2]}$, $\cdots$, such that for $k = 0, 1, \cdots$,

$$\begin{aligned} u^{[k+1]} &= u^{[k]} + \alpha_k p^{[k]}, \\ \alpha_k &= \frac{d^{[k]} \cdot p^{[k]}}{p^{[k]} \cdot Ap^{[k]}} \quad \text{(optimal step length)}, \\ p^{[k+1]} &= d^{[k+1]} + \beta_k p^{[k]} \quad \text{(for next step)}, \end{aligned}$$

where

$$\begin{aligned} \beta_k &= \frac{-d^{[k+1]} \cdot Ap^{[k]}}{p^{[k]} \cdot Ap^{[k]}}, \\ d^{[k]} &= -(Au^{[k]} - f) \quad (= f - Au^{[k]}, \text{ residual}). \end{aligned}$$

# Some properties

The vectors generated in the CG algorithm have the following properties, provided $d^{[k]} \neq 0$ (if $d^{[k]} = 0$, then we have converged):

- $p^{[k]}$ is $A$-conjugate to all the previous search directions, i.e., $p^{[k]} \cdot Ap^{[j]} = 0$ for $j = 0, 1, \cdots, k-1$.
  *Partial proof:* Note that

$$\beta_k = \frac{-d^{[k+1]} \cdot Ap^{[k]}}{p^{[k]} \cdot Ap^{[k]}} \Rightarrow (d^{[k+1]} + \beta_k p^{[k]}) \cdot Ap^{[k]} = 0 \Rightarrow p^{[k+1]} \cdot Ap^{[k]} = 0.$$

- The residual $d^{[k]}$ is orthogonal to all previous residuals, $d^{[k]} \cdot d^{[j]} = 0$ for $j = 0, 1, \cdots, k-1$.
- The following three subspaces of $\mathbb{R}^m$ are identical:

$$\begin{aligned} &\mathrm{span}(p^{[0]}, p^{[1]}, p^{[2]}, \cdots, p^{[k-1]}), \\ &\mathrm{span}(d^{[0]}, Ad^{[0]}, A^2 d^{[0]}, \cdots, A^{k-1} d^{[0]}), \\ &\mathrm{span}(Ae^{[0]}, A^2 e^{[0]}, A^3 e^{[0]}, \cdots, A^k e^{[0]}) \qquad (e^{[0]} := u^{[0]} - u^*). \end{aligned}$$

# Convergence of conjugate gradient

- There exists $k \leq m$ such that $Au^{[k]} = f$.
- Define the $A$-norm by
$$\|e\|_A := \sqrt{e^\top A e}.$$
Then we have that after $k$ steps of the conjugate gradient method, the iteration error $e^{[k]} := u^{[k]} - u^*$ satisfies the bound
$$\|e^{[k]}\|_A \leq 2\left(\frac{\sqrt{\kappa_2(A)}-1}{\sqrt{\kappa_2(A)}+1}\right)^k \|e^{[0]}\|_A$$
- Thus, for a given $\varepsilon > 0$, to satisfy $\|u^{[k]} - u^*\|_A \leq \varepsilon\|u^{[0]} - u^*\|_A$, it is sufficient to choose $k$ such that
$$2\left(\frac{\sqrt{\kappa_2(A)}-1}{\sqrt{\kappa_2(A)}+1}\right)^k \leq \varepsilon.$$
That is
$$k \geq \frac{1}{2}\sqrt{\kappa_2(A)}\log\frac{2}{\varepsilon} = O(\sqrt{\kappa_2(A)}).$$
In many numerical methods for elliptic PDEs, $\kappa_2(A) = O(h^{-2})$.