Unit 3.2 The Conjugate Gradient Method

Numerical Analysis

Mar. 15, 2017

EE/NTHU

Bilinear Form

- Given a symmetric $n \times n$ positive definite matrix $\mathbf{A}$ and an n-vector $\mathbf{b}$, a bilinear form can be defined as
$$\Phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x}. \tag{3.2.1}$$
- Note that since $\mathbf{A}$ is positive definite, $\mathbf{x}^T\mathbf{A}\mathbf{x} \geq 0$ for any $\mathbf{x} \in \mathbb{R}^n$, and $\mathbf{x}^T\mathbf{A}\mathbf{x} = 0$ if and only if $\mathbf{x} = \mathbf{0}$.
- Thus the bilinear form has a minimum point $\mathbf{x}^*$ such that $\Phi(\mathbf{x}) \geq \Phi(\mathbf{x}^*)$ for any $\mathbf{x} \in \mathbb{R}$.
- The minimum point $\mathbf{x}^*$ satisfies
$$\nabla\Phi(\mathbf{x}^*) = \mathbf{0}.$$
Since
$$\nabla\Phi(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}. \tag{3.2.2}$$
Thus,
$$\begin{aligned} \mathbf{A}\mathbf{x}^* - \mathbf{b} &= \mathbf{0}, \\ \mathbf{A}\mathbf{x}^* &= \mathbf{b}. \end{aligned} \tag{3.2.3}$$
- Finding the solution of the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ is equivalent to finding the minimum point of the bilinear form $\frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x}$.

Steepest Descent Method

- Given any initial guess $\mathbf{x}^{(0)}$ the steepest descent method finds the minimum point of the bilinear form $\Phi(\mathbf{x})$ along the gradient direction.
- Since the gradient $\nabla\Phi(\mathbf{x})$ is the direction of the maximum ascent and $-\nabla\Phi(\mathbf{x})$ is the maximum descent, this method makes sense intuitively. And the search direction is

$$-\nabla\Phi(\mathbf{x}) = -\mathbf{A}\mathbf{x} + \mathbf{b} = \mathbf{r}. \tag{3.2.4}$$

  where $\mathbf{r}$ is the residue of $\mathbf{x}$.
- At iteration k, we have point $\mathbf{x}^{(k)}$ and try to find $\mathbf{x}^{(k+1)}$ along the steepest descent direction.

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{r}^{(k)}. \tag{3.2.5}$$

  to find the minimum of

$$\Phi(\mathbf{x}^{(k+1)}) = \frac{1}{2}(\mathbf{x}^{(k)} + \alpha_k\mathbf{r}^{(k)})^T\mathbf{A}(\mathbf{x}^{(k)} + \alpha_k\mathbf{r}^{(k)}) - \mathbf{b}^T(\mathbf{x}^{(k)} + \alpha_k\mathbf{r}^{(k)})$$

- The minimum $\Phi(\mathbf{x}^{(k+1)})$ must satisfy $\dfrac{d\,\Phi(\mathbf{x}^{(k+1)})}{d\,\alpha_k} = 0$, and

$$\begin{aligned}\frac{d\,\Phi(\mathbf{x}^{(k+1)})}{d\,\alpha_k} &= (\mathbf{r}^{(k)})^T\mathbf{A}\mathbf{x}^{(k)} + \alpha_k(\mathbf{r}^{(k)})^T\mathbf{A}\mathbf{r}^{(k)} - \mathbf{b}^T\mathbf{r}^{(k)} \\ &= \alpha_k(\mathbf{r}^{(k)})^T\mathbf{A}\mathbf{r}^{(k)} - (\mathbf{r}^{(k)})^T\mathbf{r}^{(k)}\end{aligned}$$

Steepest Descent Method, II

- Thus, we have

$$\alpha_k = \frac{(\mathbf{r}^{(k)})^T\mathbf{r}^{(k)}}{(\mathbf{r}^{(k)})^T\mathbf{A}\mathbf{r}^{(k)}}. \tag{3.2.6}$$

- Also,

$$\begin{aligned}\mathbf{r}^{(k+1)} &= \mathbf{b} - \mathbf{A}\mathbf{x}^{(k+1)} \\ &= \mathbf{b} - \mathbf{A}(\mathbf{x}^{(k)} + \alpha_k\mathbf{r}^{(k)}) \\ &= \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)} - \alpha_k\mathbf{A}\mathbf{r}^{(k)} \\ &= \mathbf{r}^{(k)} - \alpha_k\mathbf{A}\mathbf{r}^{(k)}. \end{aligned} \tag{3.2.7}$$

- This leads to

Algorithm 3.2.1. Steepest Descent

Given a linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ with a symmetric and positive definite matrix $\mathbf{A}$ and an initial guess $\mathbf{x}^{(0)}$, let $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$, and repeat for $k \geq 1$

$$\alpha_k = \frac{(\mathbf{r}^{(k)})^T\mathbf{r}^{(k)}}{(\mathbf{r}^{(k)})^T\mathbf{A}\mathbf{r}^{(k)}}, \tag{3.2.8}$$

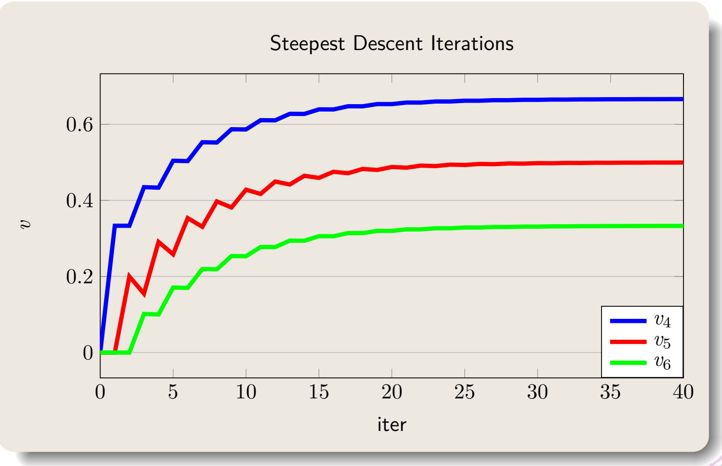
$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k\mathbf{r}^{(k)}, \tag{3.2.9}$$

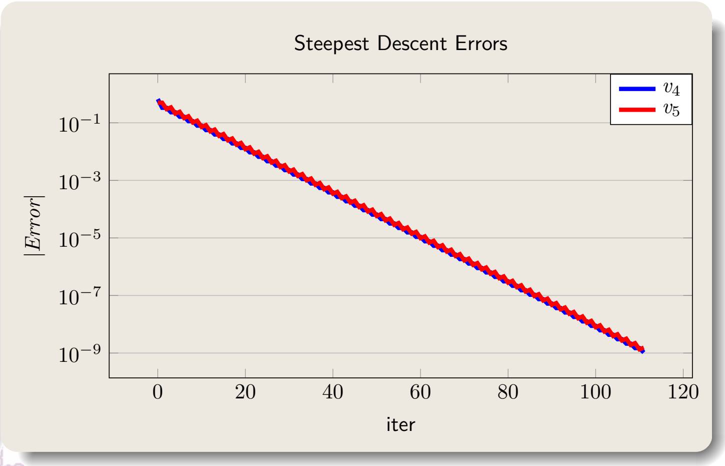
$$\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha_k\mathbf{A}\mathbf{r}^{(k)}. \tag{3.2.10}$$

Steepest Descent Method, III

- In the Steepest Descent algorithm, the computation time is dominated by the matrix-vector multiplication, $\mathbf{Ar}^{(k)}$.
  - This is of $\mathcal{O}(n^2)$.
  - All other operations involves vector operations, of $\mathcal{O}(n)$,
  - Thus, the time complexity is $\mathcal{O}(n^2)$ per iteration.
- Also note that $\mathbf{Ar}^{(k)}$ can be calculated once and applied for both Eqs. (3.2.8) and (3.2.10).
- The overall time complexity depends on the number of iterations needed to get to the desired degree of accuracy.
  - The time complexity if $\mathcal{O}(N_{iter} \times n^2)$.

Steepest Descent Method, IV





- Using the resistor network as an example, it is shown the steepest descent method converges.
  - The average convergence rate is constant, but the convergence is not monotonic.
  - Note that the original 9×9 matrix is transformed into a 7×7 symmetric matrix.

Steepest Descent Method, V

- The convergence property of the steepest descent method is given below.

Theorem. 3.2.2.

Given a symmetric and positive definite matrix $\mathbf{A}$, then the steepest gradient method is convergent for any choice of the initial guess $\mathbf{x}^{(0)}$. Moreover,

$$\|\mathbf{e}^{(k+1)}\|_{\mathbf{A}} \leq \frac{\kappa_2(\mathbf{A})-1}{\kappa_2(\mathbf{A})+1}\|\mathbf{e}^{(k)}\|_{\mathbf{A}}, \qquad k=0,1,\ldots. \tag{3.2.11}$$

with $\|\mathbf{x}\|_{\mathbf{A}}$ defined as

$$\|\mathbf{x}\|_{\mathbf{A}} = (\mathbf{x}^T\mathbf{A}\mathbf{x})^{1/2}. \tag{3.2.12}$$

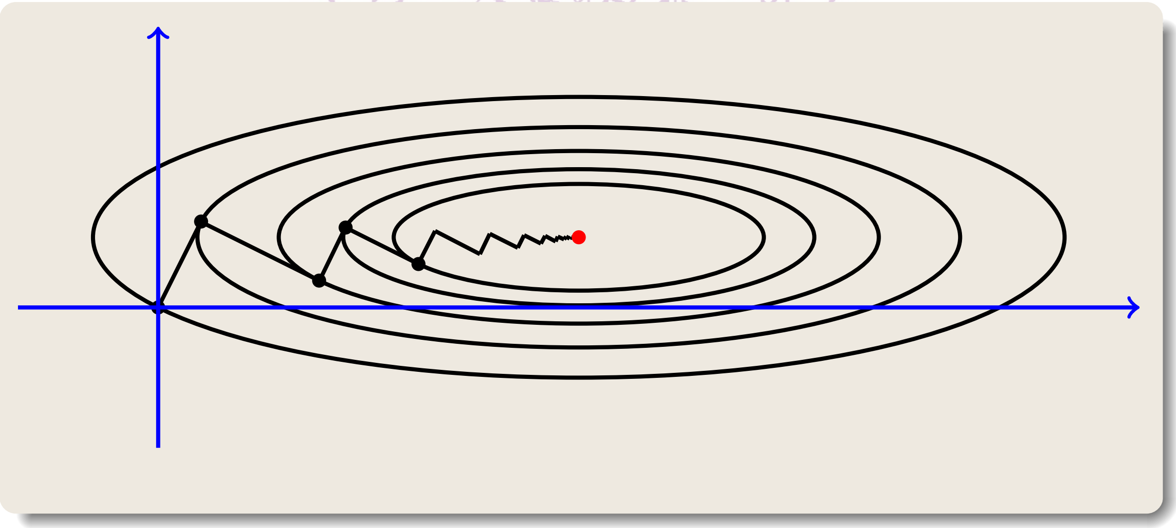
- Note that $\kappa_2(\mathbf{A}) = \|\mathbf{A}\|_2\|\mathbf{A}^{-1}\|_2 = \dfrac{\lambda_{max}}{\lambda_{min}} \geq 1$.
- If $\kappa_2(\mathbf{A}) \gg 1$, then $\|\mathbf{e}^{(k+1)}\|_{\mathbf{A}} \approx \|\mathbf{e}^{(k)}\|_{\mathbf{A}}$ and the convergence rate is slow.
- On the other hand, if $\kappa_2(\mathbf{A}) \approx 1$, then $\|\mathbf{e}^{(k+1)}\|_{\mathbf{A}} \approx 0$ and the convergence rate is fast.

Steepest Descent Method, VI

- Example iteration process for

$$\begin{bmatrix}1 & 0\\0 & 12\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} = \begin{bmatrix}6\\12\end{bmatrix}$$

with $\mathbf{x}_0^T = [0 \quad 0]$.



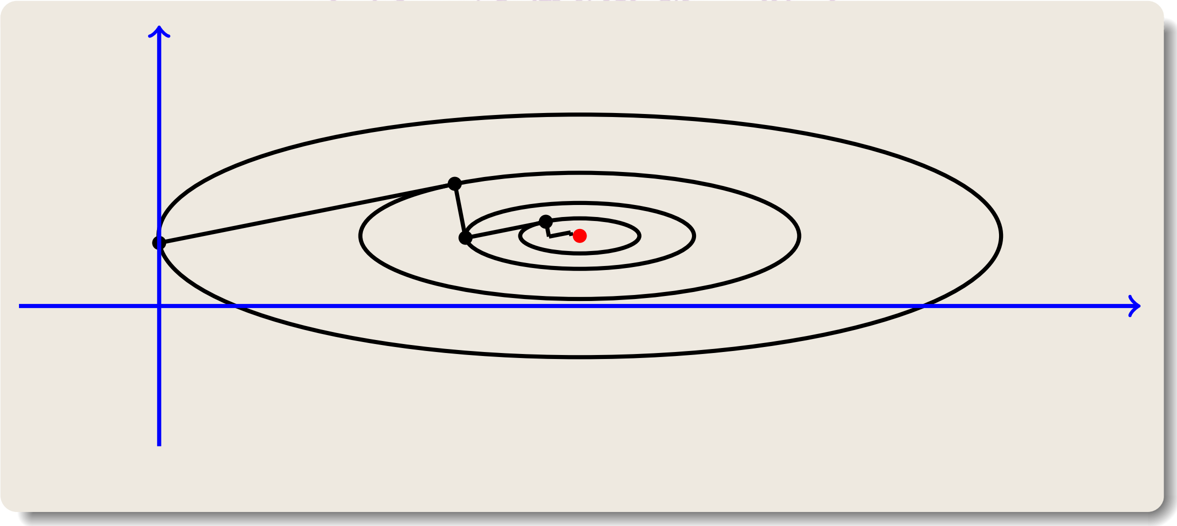
- Steepest descent method is convergent but slow.

Steepest Descent Method, VII

- Example iteration process for

$$\begin{bmatrix} 1 & 0 \\ 0 & 12 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 12 \end{bmatrix}$$

with $\mathbf{x}_0^T = [0 \quad 0.9]$.



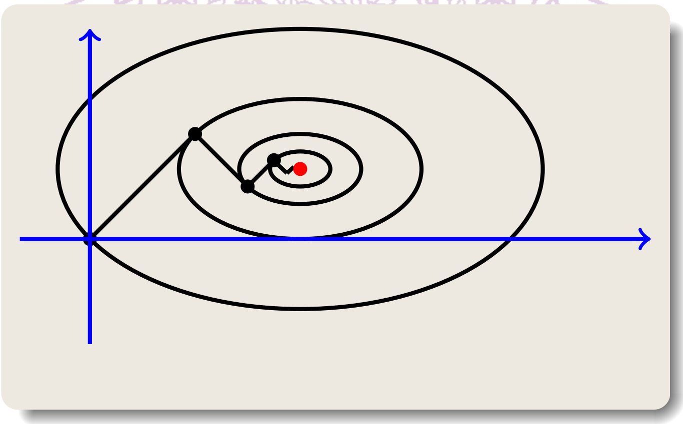
- Convergence rate of steepest descent method depends on the initial guess.

Steepest Descent Method, VIII

- Example iteration process for

$$\begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

with $\mathbf{x}_0^T = [0 \quad 0]$.



- Convergence rate of steepest descent method also depends on matrix $\mathbf{A}$.

The Conjugate Gradient Method

- There are two steps in steepest descent
  - To choose a search path (which is the negative gradient direction)
  - And to find the minimum point along the path
- If, instead of using $\mathbf{r}^{(k)}$ as search path, we choose a different direction $\mathbf{p}^{(k)}$, let

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)} \tag{3.2.13}$$

Then

$$\Phi(\mathbf{x}^{(k+1)}) = \frac{1}{2}(\mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)})^T \mathbf{A}(\mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}) - \mathbf{b}^T(\mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}) \tag{3.2.14}$$

And the minimum point satisfies

$$\begin{aligned} \frac{d\,\Phi(\mathbf{x}^{(k+1)})}{d\,\alpha_k} &= (\mathbf{p}^{(k)})^T \mathbf{A}\mathbf{x}^{(k)} + \alpha_k (\mathbf{p}^{(k)})^T \mathbf{A}\mathbf{p}^{(k)} - \mathbf{b}^T \mathbf{p}^{(k)} \\ &= (\mathbf{p}^{(k)})^T \left[\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}\right] + \alpha_k (\mathbf{p}^{(k)})^T \mathbf{A}\mathbf{p}^{(k)} \\ &= -(\mathbf{p}^{(k)})^T \mathbf{r}^{(k)} + \alpha_k (\mathbf{p}^{(k)})^T \mathbf{A}\mathbf{p}^{(k)} = 0 \end{aligned}$$

Thus,

$$\alpha_k = \frac{(\mathbf{p}^{(k)})^T \mathbf{r}^{(k)}}{(\mathbf{p}^{(k)})^T \mathbf{A}\mathbf{p}^{(k)}}. \tag{3.2.15}$$

The Conjugate Gradient Method, II

- With the new search direction $\mathbf{p}^{(k)}$,

$$\begin{aligned} \mathbf{r}^{(k+1)} &= \mathbf{b} - \mathbf{A}\mathbf{x}^{(k+1)} \\ &= \mathbf{b} - \mathbf{A}(\mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}) \\ &= \mathbf{r}^{(k)} - \alpha_k \mathbf{A}\mathbf{p}^{(k)} \end{aligned} \tag{3.2.16}$$

And

$$\begin{aligned} (\mathbf{p}^{(k)})^T \mathbf{r}^{(k+1)} &= (\mathbf{p}^{(k)})^T \mathbf{r}^{(k)} - \alpha_k (\mathbf{p}^{(k)})^T \mathbf{A}\mathbf{p}^{(k)} \\ &= (\mathbf{p}^{(k)})^T \mathbf{r}^{(k)} - (\mathbf{p}^{(k)})^T \mathbf{r}^{(k)} \\ &= 0. \end{aligned} \tag{3.2.17}$$

Therefore, the new gradient $\mathbf{r}^{(k+1)}$ is always orthogonal to the search direction $\mathbf{p}^{(k)}$.

- The conjugate gradient method defines the search direction as the following:

$$\mathbf{p}^{(0)} = \mathbf{r}^{(0)}, \tag{3.2.18}$$

$$\mathbf{p}^{(k+1)} = \mathbf{r}^{(k+1)} - \beta_k \mathbf{p}^{(k)}, \qquad k = 0, 1, \ldots. \tag{3.2.19}$$

The Conjugate Gradient Method, III

- The parameter β_k is determined by enforcing the **A**-conjugate condition on $\mathbf{p}^{(k+1)}$ and $\mathbf{p}^{(k)}$, that is,

$$(\mathbf{p}^{(k)})^T\mathbf{A}\mathbf{p}^{(k+1)} = 0. \tag{3.2.20}$$

Note that since $\mathbf{A}$ is symmetric,

$$(\mathbf{p}^{(k)})^T\mathbf{A}\mathbf{p}^{(k+1)} = (\mathbf{p}^{(k+1)})^T\mathbf{A}\mathbf{p}^{(k)}. \tag{3.2.21}$$

Since

$$(\mathbf{p}^{(k)})^T\mathbf{A}\mathbf{p}^{(k+1)} = (\mathbf{p}^{(k)})^T\mathbf{A}\mathbf{r}^{(k+1)} - (\beta_k\mathbf{p}^{(k)})^T\mathbf{A}\mathbf{p}^{(k)},$$

we have

$$\beta_k = \frac{(\mathbf{p}^{(k)})^T\mathbf{A}\mathbf{r}^{(k+1)}}{(\mathbf{p}^{(k)})^T\mathbf{A}\mathbf{p}^{(k)}}. \tag{3.2.22}$$

The Conjugate Gradient Method, IV

- Combining equations (3.2.15), (3.2.13), (3.2.16), (3.2.22), and (3.2.19), we have

Algorithm 3.2.3. Conjugate Gradient Method.

Given a linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ with a symmetric positive definite matrix $\mathbf{A}$ and an initial guess $\mathbf{x}^{(0)}$, let $\mathbf{p}^{(0)} = \mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$, for $k = 0, 1, \ldots$

$$\alpha_k = \frac{(\mathbf{p}^{(k)})^T\mathbf{r}^{(k)}}{(\mathbf{p}^{(k)})^T\mathbf{A}\mathbf{p}^{(k)}}, \tag{3.2.23}$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k\mathbf{p}^{(k)}, \tag{3.2.24}$$

$$\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha_k\mathbf{A}\mathbf{p}^{(k)}, \tag{3.2.25}$$

$$\beta_k = \frac{(\mathbf{p}^{(k)})^T\mathbf{A}\mathbf{r}^{(k+1)}}{(\mathbf{p}^{(k)})^T\mathbf{A}\mathbf{p}^{(k)}}, \tag{3.2.26}$$

$$\mathbf{p}^{(k+1)} = \mathbf{r}^{(k+1)} - \beta_k\mathbf{p}^{(k)}. \tag{3.2.27}$$

The Conjugate Gradient Method, V

- It can be shown that $\mathbf{p}^{(k+1)}$ is $\mathbf{A}$-orthogonal to the subspace formed by $\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k)}\}$.
- And, $\mathbf{r}^{(k+1)}$ is orthogonal to the same subspace $\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k)}\}$ and also $\{\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(k)}\}$.

Theorem. 3.2.4.

Let $\mathbf{A}$ be a symmetric and positive definite matrix, then the conjugate gradient algorithm in solving the linear system $\mathbf{Ax} = \mathbf{b}$ terminates at most n iterations, with the exact solution.

Theorem. 3.2.5.

Given the linear system $\mathbf{Ax} = \mathbf{b}$ with a symmetric and positive definite matrix $\mathbf{A}$. The conjugate gradient method converges in at most n iterations, and the error $\mathbf{e}^{(k)}$ at the k-th iteration $(0 \le k < n)$ is orthogonal to $\mathbf{p}^{(j)}, j = 0, \ldots, k-1$ and

$$\|\mathbf{e}^{(k)}\|_{\mathbf{A}} \le \frac{2c^k}{1+c^{2k}} \|\mathbf{e}^{(0)}\|_{\mathbf{A}}, \text{ with } c = \frac{\sqrt{\kappa_2(\mathbf{A})}-1}{\sqrt{\kappa_2(\mathbf{A})}+1}. \tag{3.2.28}$$

- Thus, the conjugate gradient method converges much faster than the steepest descent method.

The Conjugate Gradient Method, VI

- Note that Eq. (3.2.23) is

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

Since

$$\begin{aligned}
(\mathbf{p}^{(k)})^T \mathbf{r}^{(k)} &= (\mathbf{r}^{(k)})^T \mathbf{p}^{(k)} \\
&= (\mathbf{r}^{(k)})^T (\mathbf{r}^{(k)} - \beta_{k-1} \mathbf{p}^{(k-1)}) \\
&= (\mathbf{r}^{(k)})^T \mathbf{r}^{(k)} - \beta_{k-1} (\mathbf{r}^{(k)})^T \mathbf{p}^{(k-1)} \\
&= (\mathbf{r}^{(k)})^T \mathbf{r}^{(k)}
\end{aligned}$$

due to Eq. (3.2.17), thus

$$\alpha_k = \frac{(\mathbf{r}^{(k)})^T \mathbf{r}^{(k)}}{(\mathbf{p}^{(k)})^T \mathbf{A} \mathbf{p}^{(k)}}. \tag{3.2.29}$$

The Conjugate Gradient Method, VII

- To simplify β_k, we'll need the following
  From Eq. (3.2.25),

$$\begin{aligned}(\mathbf{r}^{(k)})^T\mathbf{r}^{(k+1)} &= (\mathbf{r}^{(k)})^T\left[\mathbf{r}^{(k)} - \alpha_k\mathbf{A}\mathbf{p}^{(k)}\right] \\ &= (\mathbf{r}^{(k)})^T\left[\mathbf{r}^{(k)} - \frac{(\mathbf{p}^{(k)})^T\mathbf{r}^{(k)}}{(\mathbf{p}^{(k)})^T\mathbf{A}\mathbf{p}^{(k)}}\mathbf{A}\mathbf{p}^{(k)}\right] \\ &= \mathbf{0}.\end{aligned} \tag{3.2.30}$$

- Thus, $\mathbf{r}^{(k+1)}$ is orthogonal to $\mathbf{r}^{(k)}$.
- In fact, it has been proven that $\mathbf{r}^{(k+1)}$ is orthogonal to the subspace spanned by $\{\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(k)}\}$.

The Conjugate Gradient Method, VIII

- Also, from Eq. (3.2.16)

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

  Then

$$\begin{aligned}\beta_k &= \frac{(\mathbf{p}^{(k)})^T\mathbf{A}\mathbf{r}^{(k+1)}}{(\mathbf{p}^{(k)})^T\mathbf{A}\mathbf{p}^{(k)}} = \frac{(\mathbf{r}^{(k+1)})^T\mathbf{A}\mathbf{p}^{(k)}}{(\mathbf{p}^{(k)})^T\mathbf{A}\mathbf{p}^{(k)}} \\ &= \frac{(\mathbf{r}^{(k+1)})^T(\mathbf{r}^{(k)} - \mathbf{r}^{(k+1)})}{(\mathbf{p}^{(k)})^T\mathbf{A}\mathbf{p}^{(k)}}\frac{1}{\alpha_k} \\ &= -\frac{(\mathbf{r}^{(k+1)})^T\mathbf{r}^{(k+1)}}{(\mathbf{p}^{(k)})^T\mathbf{A}\mathbf{p}^{(k)}}\frac{(\mathbf{p}^{(k)})^T\mathbf{A}\mathbf{p}^{(k)}}{(\mathbf{r}^{(k)})^T\mathbf{r}^{(k)}}\end{aligned}$$

  Thus

$$\beta_k = -\frac{(\mathbf{r}^{(k+1)})^T\mathbf{r}^{(k+1)}}{(\mathbf{r}^{(k)})^T\mathbf{r}^{(k)}}. \tag{3.2.31}$$

- These leads to the alternative form of the conjugate gradient algorithm.

The Conjugate Gradient Method, IX

Algorithm 3.2.6. Conjugate Gradient Method (2nd Form).

Given a linear system $\mathbf{Ax} = \mathbf{b}$ with a symmetric positive definite matrix $\mathbf{A}$ and an initial guess $\mathbf{x}^{(0)}$, let $\mathbf{p}^{(0)} = \mathbf{r}^{(0)} = \mathbf{b} - \mathbf{Ax}^{(0)}$, for $k = 0, 1, \dots$

$$\alpha_k = \frac{(\mathbf{r}^{(k)})^T \mathbf{r}^{(k)}}{(\mathbf{p}^{(k)})^T \mathbf{A} \mathbf{p}^{(k)}}, \tag{3.2.32}$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}, \tag{3.2.33}$$

$$\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha_k \mathbf{A} \mathbf{p}^{(k)}, \tag{3.2.34}$$

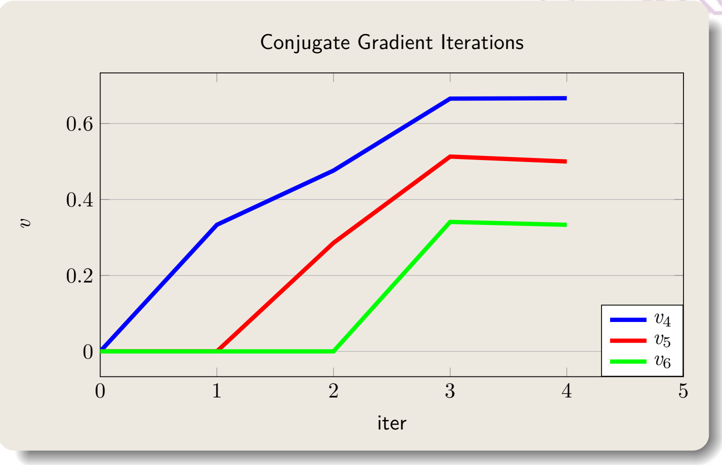
$$\beta_k = \frac{(\mathbf{r}^{(k+1)})^T \mathbf{r}^{(k+1)}}{(\mathbf{r}^{(k)})^T \mathbf{r}^{(k)}}, \tag{3.2.35}$$

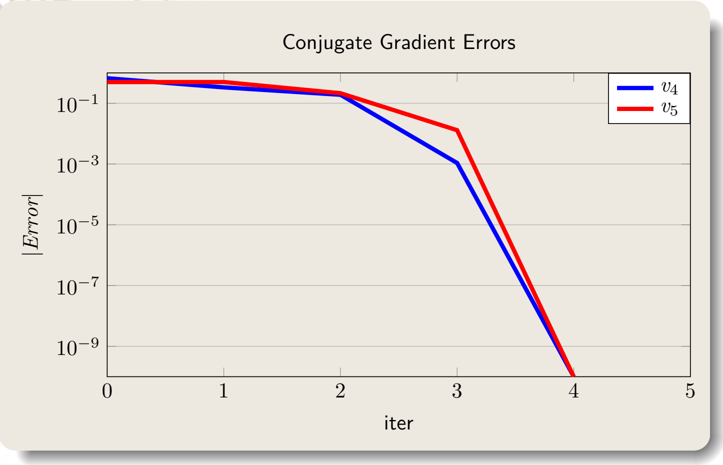
$$\mathbf{p}^{(k+1)} = \mathbf{r}^{(k+1)} + \beta_k \mathbf{p}^{(k)}. \tag{3.2.36}$$

- In this form, the scalar $(\mathbf{r}^T \mathbf{r})$ can be saved and reused from this iteration to the next iteration.

The Conjugate Gradient Method, X

- Using the resistor network as an example





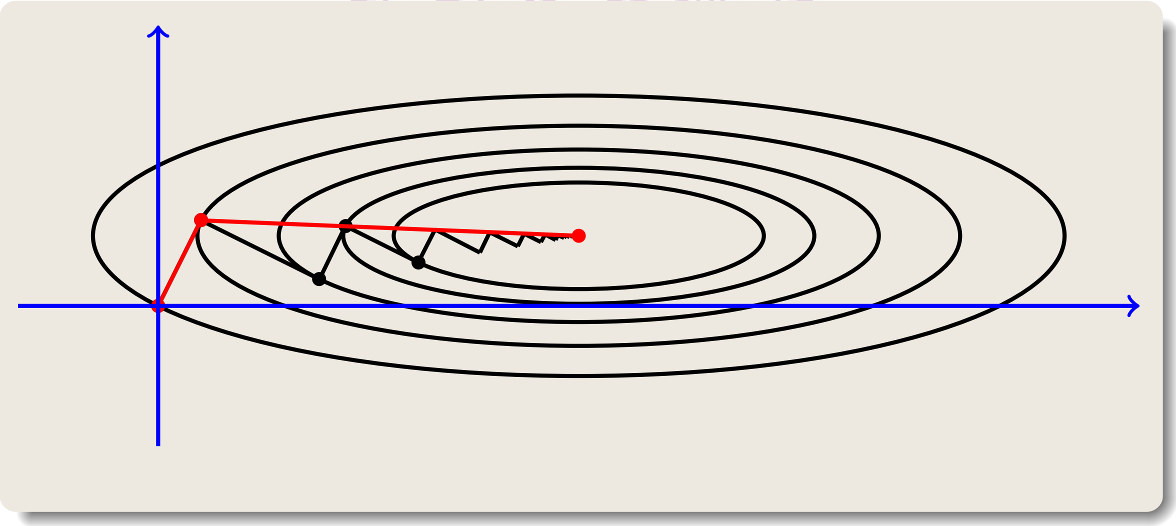
- The conjugate gradient method converges quickly.
  - Again, it is applied to the 7×7 symmetric and positive definite matrix.

The Conjugate Gradient Method, XI

- Example iteration process for

$$\begin{bmatrix} 1 & 0 \\ 0 & 12 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 12 \end{bmatrix}$$

with $\mathbf{x}_0^T = [0 \quad 0]$.



- Conjugate gradient method converges in 2 iterations.
  - With the same $\mathbf{x}^{(0)}$, the $\mathbf{x}^{(1)}$ is the same as the steepest descent method.
  - Then, the conjugate gradient method converges in the next iteration.
    - Independent to the initial condition and matrix condition number.

The Conjugate Gradient Method, XII

- Note that in the alternative form of conjugate gradient algorithm, only one matrix-vector multiplication is needed for each iteration.
  - Most operations involve vector-vector operations.
  - $\mathcal{O}(n^2)$ in the worst case
  - If the matrix is sparse, $\mathcal{O}(NZ)$, where NZ is the number of nonzero entries in the matrix.
  - Since the conjugate gradient method takes at most n iterations, the overall complexity is $\mathcal{O}(n^3)$.
  - In case of sparse matrix $\mathcal{O}(n \times NZ)$.
- Thus, the conjugate gradient method is very efficient.

Summary

- Bilinear form
- Minimum point of bilinear form and solution of linear system
- Steepest descent method
- Conjugate gradient method