The Conjugate Gradient Method

Jason E. Hicken

Aerospace Design Lab
Department of Aeronautics & Astronautics
Stanford University

14 July 2011

Lecture Objectives

- describe when CG can be used to solve $Ax = b$
- relate CG to the method of conjugate directions
- describe what CG does geometrically
- explain each line in the CG algorithm

We are interested in solving the linear system

$$Ax = b$$

where $x, b \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n \times n}$

Matrix is symmetric positive-definite (SPD)

$$\begin{aligned} A^T &= A && \text{(symmetric)} \\ x^T A x &> 0, \quad \forall\, x \neq 0 && \text{(positive-definite)} \end{aligned}$$

We are interested in solving the linear system

$$Ax = b$$

where $x,\ b \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n\times n}$

Matrix is symmetric positive-definite (SPD)

$$\begin{aligned} A^T &= A & &\text{(symmetric)} \\ x^T A x &> 0, \quad \forall\, x \neq 0 & &\text{(positive-definite)} \end{aligned}$$

- discretization of elliptic PDEs
- optimization of quadratic functionals
- nonlinear optimization problems

When A is SPD, solving the linear system is the same as minimizing the quadratic form

$$f(x) = \frac{1}{2}x^T Ax - b^T x.$$

Why?

When A is SPD, solving the linear system is the same as minimizing the quadratic form

$$f(x) = \frac{1}{2}x^T Ax - b^T x.$$

Why? If $x^\star$ is the minimizing point, then

$$\nabla f(x^\star) = Ax^\star - b = 0$$

and, for $x \neq x^\star$

$$f(x) - f(x^\star) > 0. \qquad \text{(homework)}$$

Definitions

Let x_i be the approximate solution to $Ax = b$ at iteration i.

$$\text{error:} \qquad e_i \equiv x_i - x$$

$$\text{residual:} \quad r_i \equiv b - Ax_i$$

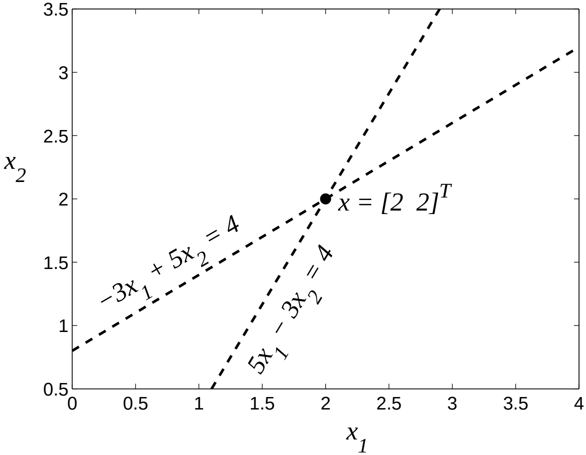
The following identities for the residual will be useful later.

$$r_i = -Ae_i$$

$$r_i = -\nabla f(x_i)$$

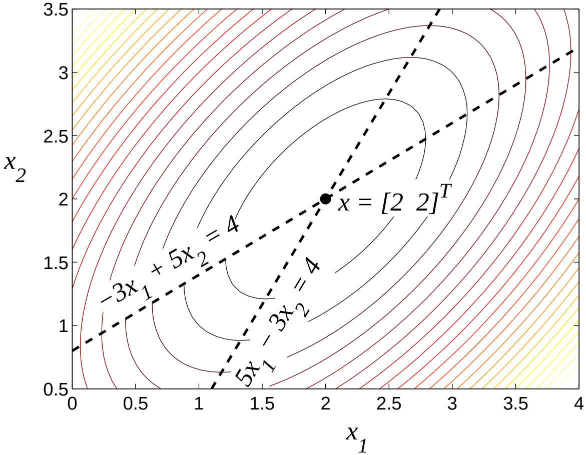
Model problem

$$\begin{bmatrix} 5 & -3 \\ -3 & 5 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \end{pmatrix}$$



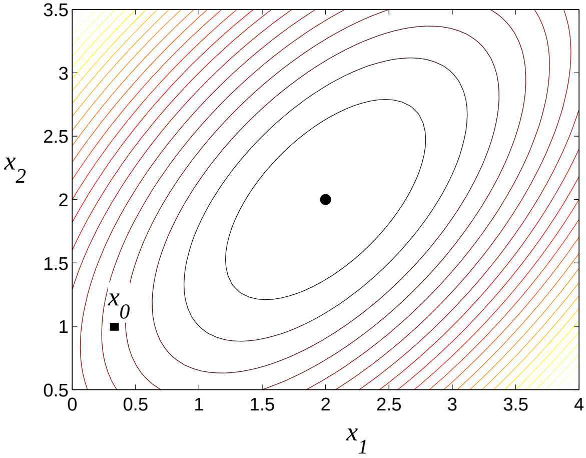
Model problem

$$\begin{bmatrix} 5 & -3 \\ -3 & 5 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \end{pmatrix}$$



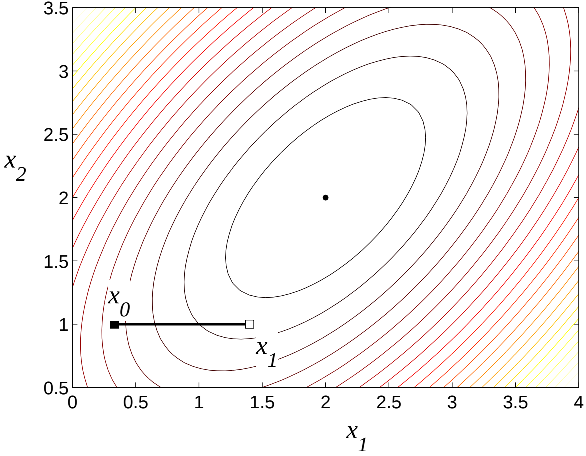
Review: Steepest Descent Method

Qualitatively, how will steepest descent proceed on our model problem, starting at $x_0 = (\frac{1}{3}, 1)^T$?



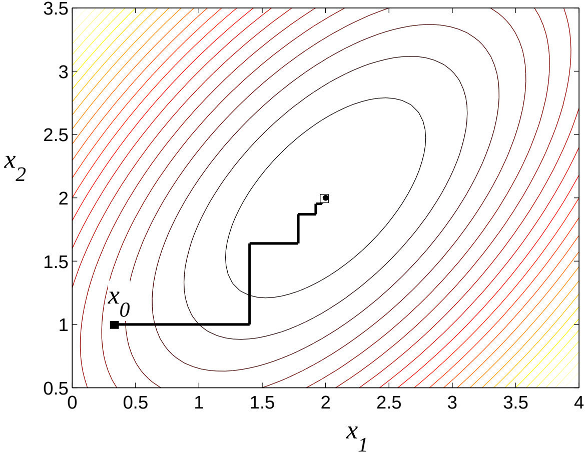
Review: Steepest Descent Method

Qualitatively, how will steepest descent proceed on our model problem, starting at $x_0 = (\frac{1}{3}, 1)^T$?



Review: Steepest Descent Method

Qualitatively, how will steepest descent proceed on our model problem, starting at $x_0 = (\frac{1}{3}, 1)^T$?



How can we eliminate this zig-zag behaviour?

To find the answer, we begin by considering the easier problem

$$\begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4\sqrt{2} \\ 0 \end{pmatrix},$$

$$f(x) = x_1^2 + 4x_2^2 - 4\sqrt{2}x_1.$$

How can we eliminate this zig-zag behaviour?

To find the answer, we begin by considering the easier problem

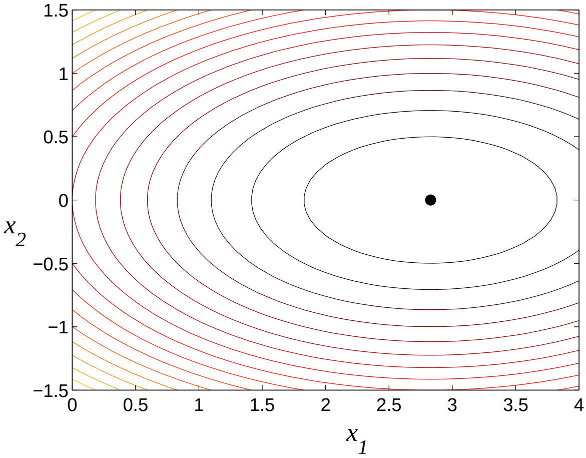
$$\begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4\sqrt{2} \\ 0 \end{pmatrix},$$

$$f(x) = x_1^2 + 4x_2^2 - 4\sqrt{2}x_1.$$

Here, the equations are decoupled, so we can minimize in each direction independently.
What do the contours of the corresponding quadratic form look like?

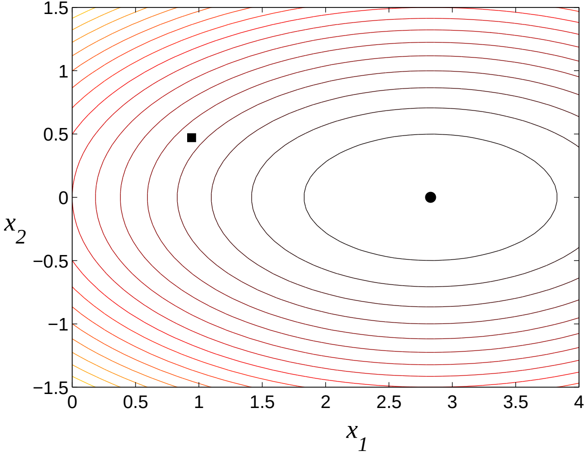
Simplified problem

$$\begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4\sqrt{2} \\ 0 \end{pmatrix}$$



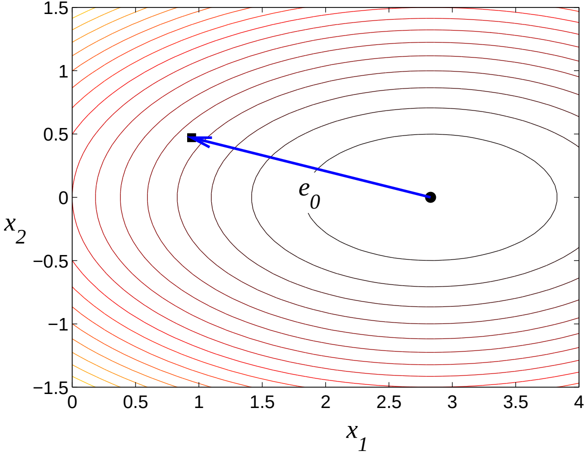
Simplified problem

$$\begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4\sqrt{2} \\ 0 \end{pmatrix}$$



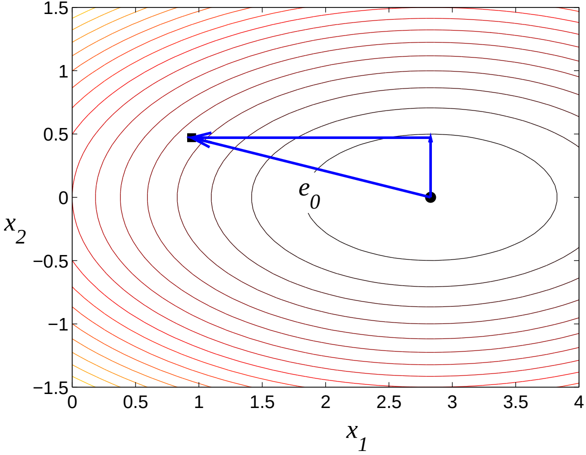
Simplified problem

$$\begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4\sqrt{2} \\ 0 \end{pmatrix}$$



Simplified problem

$$\begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4\sqrt{2} \\ 0 \end{pmatrix}$$



Method of Orthogonal Directions

Idea: Express error as a sum of n orthogonal search directions

$$e \equiv x_0 - x = \sum_{i=0}^{n-1} \alpha_i d_i.$$

At iteration $i+1$, eliminate component $\alpha_i d_i$.

Method of Orthogonal Directions

Idea: Express error as a sum of n orthogonal search directions

$$e \equiv x_0 - x = \sum_{i=0}^{n-1} \alpha_i d_i.$$

At iteration $i+1$, eliminate component $\alpha_i d_i$.

- never need to search along d_i again
- converge in n iterations!

Method of Orthogonal Directions

Idea: Express error as a sum of n orthogonal search directions

$$e \equiv x_0 - x = \sum_{i=0}^{n-1} \alpha_i d_i.$$

At iteration $i+1$, eliminate component $\alpha_i d_i$.

- never need to search along d_i again
- converge in n iterations!

How would we apply the method of orthogonal directions to a non-diagonal matrix?

Review of Inner Products

The search directions in the method of orthogonal directions are orthogonal with respect to the dot product.

The dot product is an example of an inner product.

Review of Inner Products

The search directions in the method of orthogonal directions are orthogonal with respect to the dot product.

The dot product is an example of an inner product.

Inner Product

For $x, y, z \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$, an inner product $(,) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ satisfies

- symmetry: $(x, y) = (y, x)$
- linearity: $(\alpha x + y, z) = \alpha(x, z) + (y, z)$
- positive-definiteness: $(x, x) > 0 \Leftrightarrow x \neq 0$

Fact: $(x, y)_A \equiv x^T A y$ is an inner product

A-orthogonality (conjugacy)

We say two vectors $x, y \in \mathbb{R}^n$ are A-orthogonal, or conjugate, if

$$(x, y)_A = x^T A y = 0.$$

What happens if we use A-orthogonality rather than standard orthogonality in the method of orthogonal directions?

Let $\{p_0, p_1, \ldots, p_{n-1}\}$ be a set of n linearly independent vectors that are A-orthogonal. If p_i is the i^{th} column of P, then

$$P^T A P = \Sigma$$

where Σ is a diagonal matrix.

Substitute $x = Py$ into the quadratic form:

Let $\{p_0, p_1, \ldots, p_{n-1}\}$ be a set of n linearly independent vectors that are A-orthogonal. If p_i is the i^{th} column of P, then

$$P^T A P = \Sigma$$

where Σ is a diagonal matrix.

Substitute $x = Py$ into the quadratic form:

$$f(Py) = y^T \Sigma y - (P^T b)^T y.$$

We can apply the method of orthogonal directions in y-space.

New Problem: how do we get the set $\{p_i\}$ of conjugate vectors?

New Problem: how do we get the set $\{p_i\}$ of conjugate vectors?

Gram-Schmidt Conjugation

Let $\{d_0, d_1, \ldots, d_{n-1}\}$ be a set of linearly independent vectors, e.g., coordinate axes.

- set $p_0 = d_0$
- for $i > 0$

$$p_i = d_i - \sum_{j=0}^{i-1} \beta_{ij} p_j$$

where $\beta_{ij} = (d_i, p_j)_A / (p_j, p_j)_A$.

The Method of Conjugate Directions

Force the error at iteration $i+1$ to be conjugate to the search direction p_i.

$$p_i^T A e_{i+1} = p_i^T A(e_i + \alpha_i p_i) = 0$$

The Method of Conjugate Directions

Force the error at iteration $i + 1$ to be conjugate to the search direction p_i.

$$p_i^T A e_{i+1} = p_i^T A(e_i + \alpha_i p_i) = 0$$

$$\Rightarrow \quad \alpha_i = -\frac{p_i^T A e_i}{p_i^T A p_i}$$

The Method of Conjugate Directions

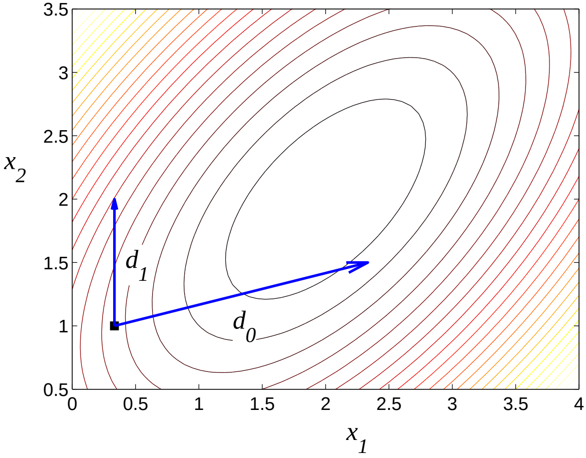
Force the error at iteration $i+1$ to be conjugate to the search direction p_i.

$$
\begin{aligned}
p_i^T A e_{i+1} &= p_i^T A(e_i + \alpha_i p_i) = 0 \\
\Rightarrow \quad \alpha_i &= -\frac{p_i^T A e_i}{p_i^T A p_i} \\
&= \frac{p_i^T r_i}{p_i^T A p_i}
\end{aligned}
$$

- never need to search along p_i again
- converge in n iterations!

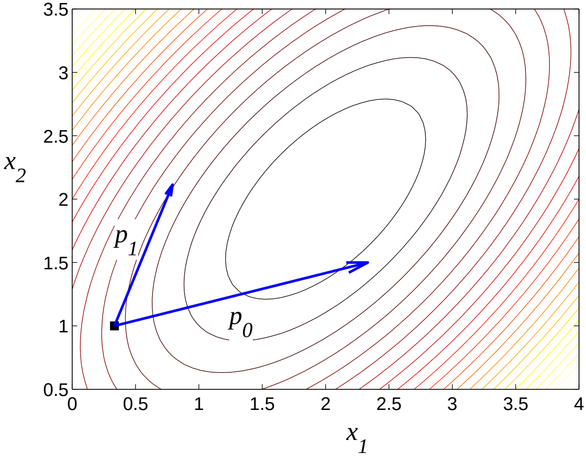
The Method of Conjugate Directions

$$\begin{bmatrix} 5 & -3 \\ -3 & 5 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \end{pmatrix}$$



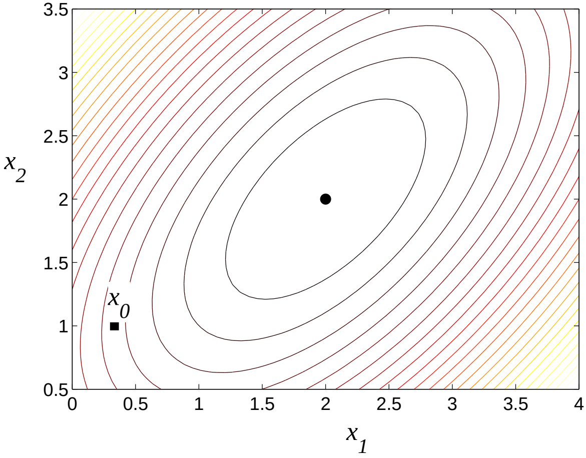
The Method of Conjugate Directions

$$\begin{bmatrix} 5 & -3 \\ -3 & 5 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \end{pmatrix}$$



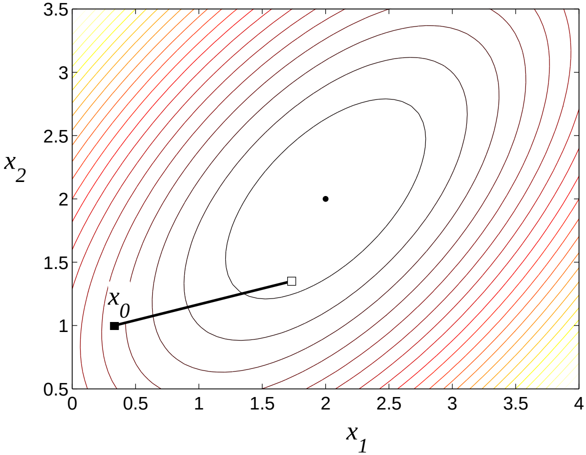
The Method of Conjugate Directions

$$\begin{bmatrix} 5 & -3 \\ -3 & 5 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \end{pmatrix}$$



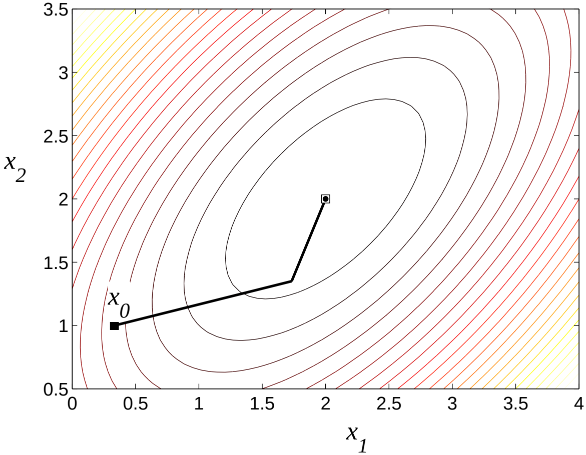
The Method of Conjugate Directions

$$\begin{bmatrix} 5 & -3 \\ -3 & 5 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \end{pmatrix}$$



The Method of Conjugate Directions

$$\begin{bmatrix} 5 & -3 \\ -3 & 5 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \end{pmatrix}$$



The Method of Conjugate Directions is well defined, and avoids the "zig-zagging" of Steepest Descent.

What about computational expense?

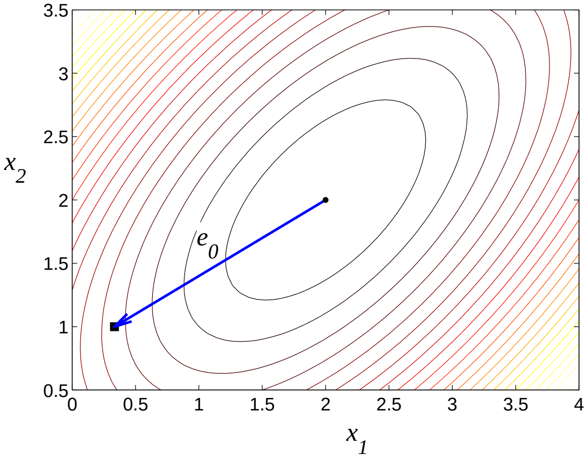
The Method of Conjugate Directions is well defined, and avoids the "zig-zagging" of Steepest Descent.

What about computational expense?

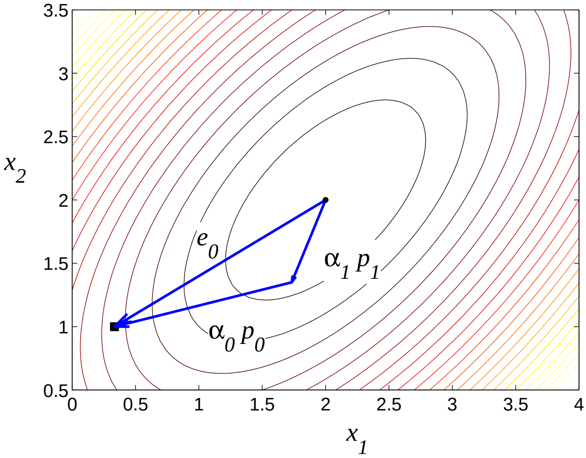
- If we choose the d_i in Gram-Schmidt conjugation to be the coordinate axes, the Method of Conjugate Directions is equivalent to Gaussian elimination.
- Keeping all the p_i is the same as storing a dense matrix!

Can we find a smarter choice for d_i?

Error Decomposition Using p_i



Error Decomposition Using p_i



The error at iteration i can be expressed as

$$e_i = \sum_{k=i}^{n-1} \alpha_k p_k,$$

The error at iteration i can be expressed as

$$e_i = \sum_{k=i}^{n-1} \alpha_k p_k,$$

so the error must be conjugate to p_j for $j < i$:

$$p_j^T A e_i = 0,$$

The error at iteration i can be expressed as

$$e_i = \sum_{k=i}^{n-1} \alpha_k p_k,$$

so the error must be conjugate to p_j for $j < i$:

$$p_j^T A e_i = 0, \quad \Rightarrow p_j^T r_i = 0,$$

The error at iteration i can be expressed as

$$e_i = \sum_{k=i}^{n-1} \alpha_k p_k,$$

so the error must be conjugate to p_j for $j < i$:

$$p_j^T A e_i = 0, \quad \Rightarrow p_j^T r_i = 0,$$

but from Gram-Schmidt conjugation we have

$$p_j = d_j - \sum_{k=0}^{j-1} \beta_{jk} p_k.$$

The error at iteration i can be expressed as

$$e_i = \sum_{k=i}^{n-1} \alpha_k p_k,$$

so the error must be conjugate to p_j for $j < i$:

$$p_j^T A e_i = 0, \quad \Rightarrow p_j^T r_i = 0,$$

but from Gram-Schmidt conjugation we have

$$\begin{aligned} p_j^T r_i &= d_j^T r_i - \sum_{k=0}^{j-1} \beta_{jk} p_k^T r_i \\ 0 &= d_j^T r_i, \quad j < i. \end{aligned}$$

Thus, the residual at iteration i is orthogonal to the vectors d_j used in the previous iterations:

$$d_j^T r_i = 0, \quad j < i$$

Idea: what happens if we choose $d_i = r_i$?

Thus, the residual at iteration i is orthogonal to the vectors d_j used in the previous iterations:

$$d_j^T r_i = 0, \quad j < i$$

Idea: what happens if we choose $d_i = r_i$?

- residuals become mutually orthogonal
- r_i is orthogonal to p_j, for $j < i$ *
- r_{i+1} becomes conjugate to p_j, for $j < i$

This last point is not immediately obvious, so we will prove it. This result has significant implications for Gram-Schmidt conjugation.

---

*we showed this is true for any choice of d_i

The solution is updated according to

$$x_{j+1} = x_j + \alpha_j p_j$$

The solution is updated according to

$$
\begin{aligned}
& x_{j+1} = x_j + \alpha_j p_j \\
\Rightarrow \quad & r_{j+1} = r_j - \alpha_j A p_j
\end{aligned}
$$

The solution is updated according to

$$
\begin{aligned}
& x_{j+1} = x_j + \alpha_j p_j \\
\Rightarrow \quad & r_{j+1} = r_j - \alpha_j A p_j \\
\Rightarrow \quad & A p_j = \frac{1}{\alpha_j}(r_j - r_{j+1}).
\end{aligned}
$$

The solution is updated according to

$$\begin{aligned} x_{j+1} &= x_j + \alpha_j p_j \\ \Rightarrow \quad r_{j+1} &= r_j - \alpha_j A p_j \\ \Rightarrow \quad A p_j &= \frac{1}{\alpha_j}(r_j - r_{j+1}). \end{aligned}$$

Next, take the dot product of both sides with an arbitrary residual r_i:

The solution is updated according to

$$
\begin{aligned}
& x_{j+1} = x_j + \alpha_j p_j \\
\Rightarrow \quad & r_{j+1} = r_j - \alpha_j A p_j \\
\Rightarrow \quad & A p_j = \frac{1}{\alpha_j}(r_j - r_{j+1}).
\end{aligned}
$$

Next, take the dot product of both sides with an arbitrary residual r_i:

$$
r_i^T A p_j = \begin{cases} \frac{r_i^T r_i}{\alpha_i}, & i = j \\ -\frac{r_i^T r_i}{\alpha_{i-1}}, & i = j+1 \\ 0, & \text{otherwise.} \end{cases}
$$

We can show that the first case ($i = j$) contains no new information (homework). Divide the remaining cases by $p_j^T A p_j$ and insert the definition of α_{i-1}:

$$\underbrace{\frac{r_i^T A p_j}{p_j^T A p_j}} = \begin{cases} -\frac{r_i^T r_i}{r_{i-1}^T r_{i-1}}, & i = j+1 \\ 0, & \text{otherwise.} \end{cases}$$

We can show that the first case ($i = j$) contains no new information (homework). Divide the remaining cases by $p_j^T A p_j$ and insert the definition of α_{i-1}:

$$\underbrace{\frac{r_i^T A p_j}{p_j^T A p_j}}_{\beta_{ij}} = \begin{cases} -\frac{r_i^T r_i}{r_{i-1}^T r_{i-1}}, & i = j + 1 \\ 0, & \text{otherwise.} \end{cases}$$

We recognize the L.H.S. as the coefficients in Gram-Schmidt conjugation

- only one coefficient is nonzero!

The Conjugate Gradient Method

Set $p_0 = r_0 = b - Ax_0$ and $i = 0$

The Conjugate Gradient Method

Set $p_0 = r_0 = b - Ax_0$ and $i = 0$

$$\alpha_i = (p_i^T r_i)/(p_i^T A p_i) \qquad \text{(step length)}$$

The Conjugate Gradient Method

Set $p_0 = r_0 = b - Ax_0$ and $i = 0$

$$\alpha_i = (p_i^T r_i)/(p_i^T A p_i) \qquad \text{(step length)}$$

$$x_{i+1} = x_i + \alpha_i p_i \qquad \text{(sol. update)}$$

The Conjugate Gradient Method

Set $p_0 = r_0 = b - Ax_0$ and $i = 0$

$$\alpha_i = (p_i^T r_i)/(p_i^T A p_i) \qquad \text{(step length)}$$

$$x_{i+1} = x_i + \alpha_i p_i \qquad \text{(sol. update)}$$

$$r_{i+1} = r_i - \alpha_i A p_i \qquad \text{(resid. update)}$$

The Conjugate Gradient Method

Set $p_0 = r_0 = b - Ax_0$ and $i = 0$

$$\alpha_i = (p_i^T r_i)/(p_i^T A p_i) \qquad \text{(step length)}$$

$$x_{i+1} = x_i + \alpha_i p_i \qquad \text{(sol. update)}$$

$$r_{i+1} = r_i - \alpha_i A p_i \qquad \text{(resid. update)}$$

$$\beta_{i+1,i} = -(r_{i+1}^T r_{i+1})/(r_i^T r_i) \qquad \text{(G.S. coeff.)}$$

The Conjugate Gradient Method

Set $p_0 = r_0 = b - Ax_0$ and $i = 0$

$$
\begin{aligned}
\alpha_i &= (p_i^T r_i)/(p_i^T A p_i) && \text{(step length)} \\
x_{i+1} &= x_i + \alpha_i p_i && \text{(sol. update)} \\
r_{i+1} &= r_i - \alpha_i A p_i && \text{(resid. update)} \\
\beta_{i+1,i} &= -(r_{i+1}^T r_{i+1})/(r_i^T r_i) && \text{(G.S. coeff.)} \\
p_{i+1} &= r_{i+1} - \beta_{i+1,i}\, p_i && \text{(Gram Schmidt)}
\end{aligned}
$$

The Conjugate Gradient Method

Set $p_0 = r_0 = b - Ax_0$ and $i = 0$

$$\alpha_i = (p_i^T r_i)/(p_i^T A p_i) \qquad \text{(step length)}$$

$$x_{i+1} = x_i + \alpha_i p_i \qquad \text{(sol. update)}$$

$$r_{i+1} = r_i - \alpha_i A p_i \qquad \text{(resid. update)}$$

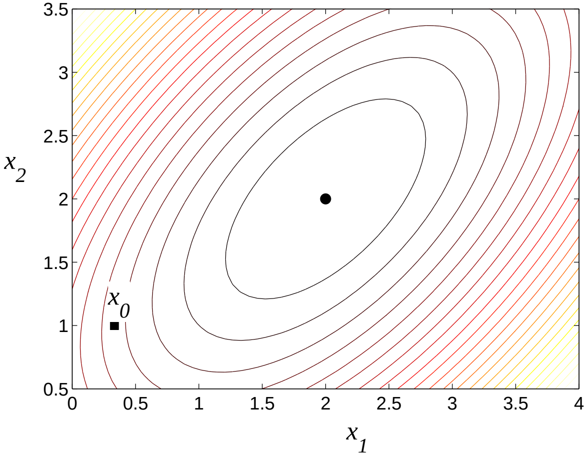
$$\beta_{i+1,i} = -(r_{i+1}^T r_{i+1})/(r_i^T r_i) \qquad \text{(G.S. coeff.)}$$

$$p_{i+1} = r_{i+1} - \beta_{i+1,i}\, p_i \qquad \text{(Gram Schmidt)}$$

$$i := i + 1$$

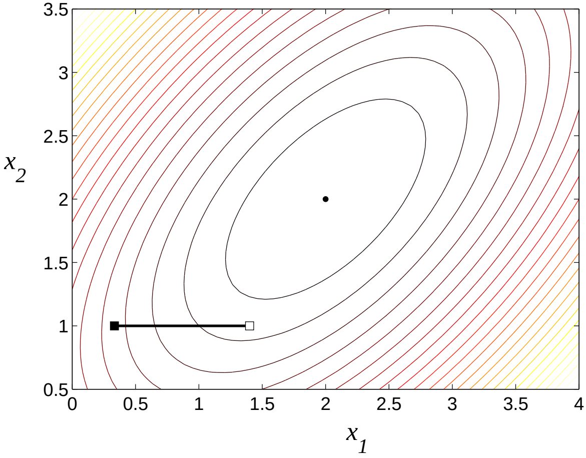
The Conjugate Gradient Method

$$\begin{bmatrix} 5 & -3 \\ -3 & 5 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \end{pmatrix}$$



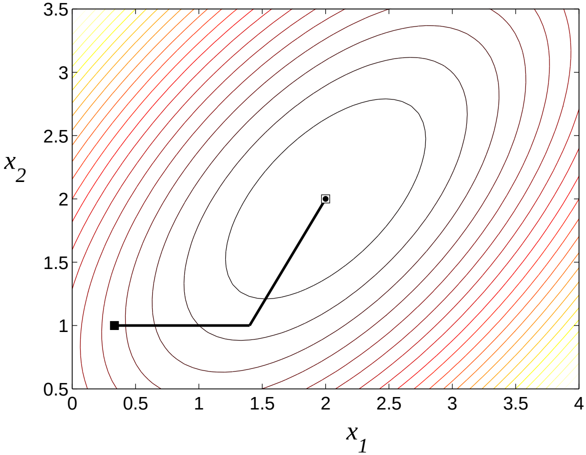
The Conjugate Gradient Method

$$\begin{bmatrix} 5 & -3 \\ -3 & 5 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \end{pmatrix}$$



The Conjugate Gradient Method

$$\begin{bmatrix} 5 & -3 \\ -3 & 5 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \end{pmatrix}$$



Lecture Objectives

- describe when CG can be used to solve $Ax = b$

Lecture Objectives

- describe when CG can be used to solve $Ax = b$
  A must be symmetric positive-definite

Lecture Objectives

- describe when CG can be used to solve $Ax = b$
  A must be symmetric positive-definite
- relate CG to the method of conjugate directions

Lecture Objectives

- describe when CG can be used to solve $Ax = b$
  A must be symmetric positive-definite
- relate CG to the method of conjugate directions
  CG is a method of conjugate directions with the choice $d_i = r_i$, which simplifies Gram-Schmidt conjugation

Lecture Objectives

- describe when CG can be used to solve $Ax = b$
  A must be symmetric positive-definite
- relate CG to the method of conjugate directions
  CG is a method of conjugate directions with the choice $d_i = r_i$, which simplifies Gram-Schmidt conjugation
- describe what CG does geometrically

Lecture Objectives

- describe when CG can be used to solve $Ax = b$
  A must be symmetric positive-definite
- relate CG to the method of conjugate directions
  CG is a method of conjugate directions with the choice $d_i = r_i$, which simplifies Gram-Schmidt conjugation
- describe what CG does geometrically
  Performs the method of orthogonal directions in a transformed space where the contours of the quadratic form are aligned with the coordinate axes

Lecture Objectives

- describe when CG can be used to solve $Ax = b$
  A must be symmetric positive-definite
- relate CG to the method of conjugate directions
  CG is a method of conjugate directions with the choice $d_i = r_i$, which simplifies Gram-Schmidt conjugation
- describe what CG does geometrically
  Performs the method of orthogonal directions in a transformed space where the contours of the quadratic form are aligned with the coordinate axes
- explain each line in the CG algorithm

References

- Saad, Y., "Iterative Methods for Sparse Linear Systems", second edition
- Shewchuk, J. R., "An introduction to the Conjugate Gradient method without the agonizing pain"