Conjugate Gradient

Section 10.8 of Numerical Recipes (3rd ed.)

Objective

Minimize a function f with the following properties:

- f can be evaluated at an N-dimensional point $\mathbf{P}$ (i.e., $f(\mathbf{P})$).
- The gradient of f can be found at an N-dimensional point $\mathbf{P}$ (i.e., $\nabla f(\mathbf{P})$).

First Intuitive Solution

Line Methods

(i.e. Powell's Method)

Characteristics

Where N is # of dimensions, requires $\boldsymbol{O}(\boldsymbol{N}^2)$ line minimizations.

First Intuitive Solution

Line Methods

(i.e. Powell's Method)

Characteristics

Where N is # of dimensions, requires $\boldsymbol{O}(\boldsymbol{N}^2)$ line minimizations.

Problem

Evaluating the gradient yields **same information** with $\boldsymbol{O}(\boldsymbol{N})$ line minimizations.

First Intuitive Solution

Line Methods

(i.e. Powell's Method)

Characteristics

Where N is # of dimensions, requires $\boldsymbol{O}(\boldsymbol{N}^2)$ line minimizations.

Problem*

Evaluating the gradient yields **same information** with $\boldsymbol{O}(\boldsymbol{N})$ line minimizations.

*Calculating *each component* of the gradient **may** take as long as evaluating the function itself, meaning the order of function evaluations **may** be the same. Even in that case, methods which use the gradient are **substantially** less computationally expensive.

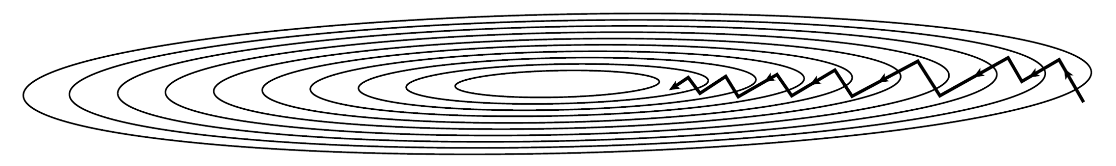
Second Intuitive Solution

Steepest Descent: Start at a point $\mathbf{P}_0$. As many times as needed, move from point $\mathbf{P}_i$ to the point $\mathbf{P}_{i+1}$ by minimizing along the line from $\mathbf{P}_i$ in the direction of the local downhill gradient $-\nabla f(\mathbf{P}_i)$.

Second Intuitive Solution

Steepest Descent: Start at a point $\mathbf{P}_0$. As many times as needed, move from point $\mathbf{P}_i$ to the point $\mathbf{P}_{i+1}$ by minimizing along the line from $\mathbf{P}_i$ in the direction of the local downhill gradient $-\nabla f(\mathbf{P}_i)$.

Problem



*"the new gradient at the minimum point of any line minimization is perpendicular to the direction just traversed. Therefore, with the steepest descent method, you **must** make a right angle turn, which does **not**, in general, take you to the minimum."*

The Big Idea

Conjugate Directions!

Defining Informally

“’**noninterfering**’ directions with the special property that minimization along one is not ‘spoiled’ by subsequent minimization along another”

Informal definition from section 10.7

Starting Note

"First, note that if we minimize a function along some direction $\mathbf{u}$, then the gradient of the function must be perpendicular to $\mathbf{u}$ at the line minimum; if not, then there would still be a nonzero directional derivative along $\mathbf{u}$."

Recall two vectors $\mathbf{v}$ *and* $\mathbf{w}$ *are perpendicular if:*

$$\mathbf{v} \cdot \mathbf{w} = 0$$

From section 10.7.1

Quadratic Forms

Take some particular point $\mathbf{P}$ as the origin of the coordinate system with coordinates $\mathbf{x}$. Then any function f can be approximated by its Taylor series (10.7.1):

$$f(\mathbf{x}) = f(\mathbf{P}) + \sum_i \frac{\partial f}{\partial x_i} x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j} x_i x_j + \cdots$$

$$\approx c - \mathbf{b} \cdot \mathbf{x} + \frac{1}{2} \mathbf{x} \cdot \mathbf{A} \cdot \mathbf{x}$$

Quadratic Forms

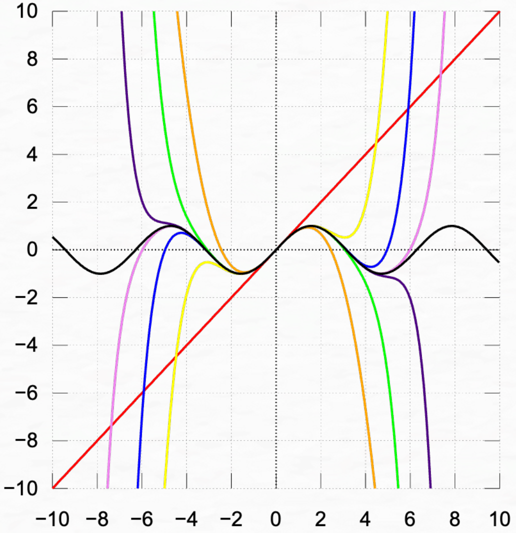
Take some particular point $\mathbf{P}$ as the origin of the coordinate system with coordinates $\mathbf{x}$. Then any function f can be approximated by its Taylor series (10.7.1):

$$f(\mathbf{x}) = f(\mathbf{P}) + \sum_i \frac{\partial f}{\partial x_i} x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j} x_i x_j + \cdots$$

$$\approx \boxed{c - \mathbf{b} \cdot \mathbf{x} + \frac{1}{2} \mathbf{x} \cdot \mathbf{A} \cdot \mathbf{x}}$$

Functions with the form above are **quadratic forms**.

Taylor Series Review



“The **Taylor series** of a real or complex-valued function $f(x)$, that is infinitely differentiable at a real or complex number a, is the power series”:

$$f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

“As the degree of the Taylor polynomial rises, **it approaches the correct function**. This image shows $\sin x$ and its Taylor approximations by polynomials of degree **1**, **3**, **5**, **7**, **9**, **11**, and **13** at $x = 0$”

From Wikipedia

Quadratic Forms

Any function can be expressed as or approximated by a quadratic form (10.7.1):

$$f(\mathbf{x}) \approx c - \mathbf{b} \cdot \mathbf{x} + \frac{1}{2}\mathbf{x} \cdot \mathbf{A} \cdot \mathbf{x}$$

Where (10.7.2):

- $c \equiv f(\mathbf{P})$
- $b \equiv -\nabla f|_{\mathbf{P}}$, the **negative gradient** at $\mathbf{P}$
- $[\mathbf{A}]_{ij} = \left.\frac{\partial^2 f}{\partial x_i \partial x_j}\right|_{\mathbf{P}}$, the **Hessian matrix** of the function at $\mathbf{P}$

Quadratic Forms

Any function can be expressed as or approximated by a quadratic form (10.7.1):

$$f(\mathbf{x}) \approx c - \mathbf{b} \cdot \mathbf{x} + \frac{1}{2}\mathbf{x} \cdot \mathbf{A} \cdot \mathbf{x}$$

Where (10.7.2):

- $c \equiv f(\mathbf{P})$
- $b \equiv -\nabla f|_{\mathbf{P}}$, the **negative gradient** at $\mathbf{P}$
- $[\mathbf{A}]_{ij} = \left.\frac{\partial^2 f}{\partial x_i \partial x_j}\right|_{\mathbf{P}}$, the **Hessian matrix** of the function at $\mathbf{P}$

The gradient of f in (10.7.1) can be easily calculated as (10.7.3):

$$\nabla f = \mathbf{A} \cdot \mathbf{x} - \mathbf{b}$$

Defining Formally

The gradient ∇f changes like such as we move along the direction $\mathbf{x}$ (10.7.4):

$$\delta(\nabla f) = \mathbf{A} \cdot (\delta \mathbf{x})$$

Defining Formally

The gradient ∇f changes like such as we move along the direction $\mathbf{x}$ (10.7.4):

$$\delta(\nabla f) = \mathbf{A} \cdot (\delta \mathbf{x})$$

If we minimize on a direction $\mathbf{u}$, and now want to minimize on a direction $\mathbf{v}$, the condition that motion along $\mathbf{v}$ not *spoil* or minimization along $\mathbf{u}$ is just that the gradient stay perpendicular to $\mathbf{u}$. By eq. (10.7.4), this is (10.7.5):

$$0 = \mathbf{u} \cdot \delta(\nabla f) = \mathbf{u} \cdot \mathbf{A} \cdot \mathbf{v}$$

Defining Formally

The gradient ∇f changes like such as we move along the direction $\mathbf{x}$ (10.7.4):

$$\delta(\nabla f) = \mathbf{A} \cdot (\delta \mathbf{x})$$

If we minimize on a direction $\mathbf{u}$, and now want to minimize on a direction $\mathbf{v}$, the condition that motion along $\mathbf{v}$ not *spoil* or minimization along $\mathbf{u}$ is just that the gradient stay perpendicular to $\mathbf{u}$. By eq. (10.7.4), this is (10.7.5):

$$0 = \mathbf{u} \cdot \delta(\nabla f) = \mathbf{u} \cdot \mathbf{A} \cdot \mathbf{v}$$

When (10.7.5) holds for two vectors $\mathbf{u}$ and $\mathbf{v}$, they are said to be **conjugate**. When the relation holds pairwise for all members of a set of vectors, they are said to be a **conjugate set**.

If you can derive a set of N linearly independent, mutually conjugate directions, then **one pass of N line minimizations** will put it **exactly at the minimum of a quadratic form**, with **no need to redo directions**!

From section 10.7.1

If you can derive a set of N linearly independent, mutually conjugate directions, then **one pass of N line minimizations** will put it **exactly at the minimum of a quadratic form**, with **no need to redo directions**!

For functions f that are not exactly quadratic forms, it won't be exactly at the minimum, but repeated cycles of N line minimizations will converge quadratically to the minimum.

From section 10.7.1

Simple Conjugate Gradient (CG)

Given a **quadratic form** f where $\mathbf{A}$ is symmetric and positive-definite, N is the number of dimensions, and $\mathbf{p}$ is a set of conjugate search directions, solve (2.7.29):

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$

From section 2.7.6

Simple Conjugate Gradient (CG)

Given a **quadratic form** f where $\mathbf{A}$ is symmetric and positive-definite, *N* is the number of dimensions, and $\mathbf{p}$ is a set of conjugate search directions, solve (2.7.29):

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$

Iterate with k until *N* iterations are complete:

1. Find α_k that minimizes $f(\mathbf{x}_k + \alpha_k \mathbf{p}_k)$ using 1D search (ex. Brent)
2. Set x_{k+1} to $\mathbf{x}_k + \alpha_k \mathbf{p}_k$

The minimum of f shall be $\mathbf{x}_N$.

From section 2.7.6

Problems with Simple CG

- **Severely** limited in applicability.
- Even if you can approximate a quadratic form and $\mathbf{A}$ fits requirements, it is **computationally expensive**.

Expanding CG

Simple CG

Quadratic forms with highly restricted $\mathbf{A}$s, can't find search directions alone

Linear CG

Quadratic forms with symmetric $\mathbf{A}$s

Nonlinear CG

(Fletcher-Reeves)
Any f where $f(x)$ and $f'(x)$ can be evaluated

Polak-Ribiere

Superior convergence to Fletcher-Reeves

Linear CG

01 Gradient

Calculates the gradient (**g**) at the current point (**P**).

02 Search Direction

Uses the gradient to find the next conjugate search direction (**h**)

03 Next Point

Minimizes on that search direction to get the next point (**P**).

04 Repeat

Performs as many iterations as the number of dimensions.

Linear CG

Theory (10.8)

Given a quadratic form f where $\mathbf{A}$ is symmetric, N is the number of dimensions, $\mathbf{P}_0$ is an arbitrary initial vector, and $\mathbf{h}_0 = \mathbf{g}_0 = \mathbf{b} - \mathbf{A}\mathbf{P}_0$.

The minimum of f is $\mathbf{P}_N$.

Example

$$\mathbf{A} = \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 2 \\ -8 \end{pmatrix}, \mathbf{P}_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\mathbf{h}_0 = \mathbf{g}_0 = \begin{pmatrix} 2 \\ -8 \end{pmatrix} - \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -3 \\ -16 \end{pmatrix}$$

Linear CG

Theory (10.8)

Given a quadratic form f where $\mathbf{A}$ is symmetric, N is the number of dimensions, $\mathbf{P}_0$ is an arbitrary initial vector, and $\mathbf{h}_0 = \mathbf{g}_0 = \mathbf{b} - \mathbf{A}\mathbf{P}_0$.

Iterate with i until N iterations are complete.

1. Calculate $\lambda_i = \frac{\mathbf{g}_i^T \cdot \mathbf{g}_i}{\mathbf{h}_i^T \cdot \mathbf{A} \cdot \mathbf{h}_i}$

The minimum of f is $\mathbf{P}_N$.

Example

$$\mathbf{A} = \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 2 \\ -8 \end{pmatrix}, \mathbf{P}_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\mathbf{h}_0 = \mathbf{g}_0 = \begin{pmatrix} 2 \\ -8 \end{pmatrix} - \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -3 \\ -16 \end{pmatrix}$$

$$\lambda_0 = \frac{\begin{pmatrix} -3 & -16 \end{pmatrix}\begin{pmatrix} -3 \\ -16 \end{pmatrix}}{\begin{pmatrix} -3 & -16 \end{pmatrix}\begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}\begin{pmatrix} -3 \\ -16 \end{pmatrix}} \approx 0.151$$

Linear CG

Theory (10.8)

Given a quadratic form f where $\mathbf{A}$ is symmetric, N is the number of dimensions, $\mathbf{P}_0$ is an arbitrary initial vector, and $\mathbf{h}_0 = \mathbf{g}_0 = \mathbf{b} - \mathbf{A}\mathbf{P}_0$.

Iterate with i until N iterations are complete.

1. Calculate $\lambda_i = \frac{\mathbf{g}_i^T \cdot \mathbf{g}_i}{\mathbf{h}_i^T \cdot \mathbf{A} \cdot \mathbf{h}_i}$
2. Calculate $\mathbf{P}_{i+1} = \mathbf{P}_i + \lambda_i \mathbf{h}_i$

The minimum of f is $\mathbf{P}_N$.

Example

$$\mathbf{A} = \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 2 \\ -8 \end{pmatrix}, \mathbf{P}_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\mathbf{h}_0 = \mathbf{g}_0 = \begin{pmatrix} 2 \\ -8 \end{pmatrix} - \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -3 \\ -16 \end{pmatrix}$$

$$\lambda_0 = \frac{\begin{pmatrix} -3 & -16 \end{pmatrix}\begin{pmatrix} -3 \\ -16 \end{pmatrix}}{\begin{pmatrix} -3 & -16 \end{pmatrix}\begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}\begin{pmatrix} -3 \\ -16 \end{pmatrix}} \approx 0.151$$

$$\mathbf{P}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 0.151\begin{pmatrix} -3 \\ -16 \end{pmatrix} = \begin{pmatrix} 0.547 \\ -1.416 \end{pmatrix}$$

Linear CG

Theory (10.8)

Given a quadratic form f where $\mathbf{A}$ is symmetric, N is the number of dimensions, $\mathbf{P}_0$ is an arbitrary initial vector, and $\mathbf{h}_0 = \mathbf{g}_0 = \mathbf{b} - \mathbf{A}\mathbf{P}_0$.

Iterate with i until N iterations are complete.

1. Calculate $\lambda_i = \frac{\mathbf{g}_i^T \cdot \mathbf{g}_i}{\mathbf{h}_i^T \cdot \mathbf{A} \cdot \mathbf{h}_i}$
2. Calculate $\mathbf{P}_{i+1} = \mathbf{P}_i + \lambda_i \mathbf{h}_i$
3. Calculate $\mathbf{g}_{i+1} = \mathbf{g}_i - \lambda_i \mathbf{A} \cdot \mathbf{h}_i$

The minimum of f is $\mathbf{P}_N$.

Example

$$\mathbf{A} = \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 2 \\ -8 \end{pmatrix}, \mathbf{P}_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\mathbf{h}_0 = \mathbf{g}_0 = \begin{pmatrix} 2 \\ -8 \end{pmatrix} - \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -3 \\ -16 \end{pmatrix}$$

$$\lambda_0 = \frac{\begin{pmatrix} -3 & -16 \end{pmatrix} \begin{pmatrix} -3 \\ -16 \end{pmatrix}}{\begin{pmatrix} -3 & -16 \end{pmatrix} \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix} \begin{pmatrix} -3 \\ -16 \end{pmatrix}} \approx 0.151$$

$$\mathbf{P}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 0.151 \begin{pmatrix} -3 \\ -16 \end{pmatrix} = \begin{pmatrix} 0.547 \\ -1.416 \end{pmatrix}$$

$$\mathbf{g}_1 = \begin{pmatrix} -3 \\ -16 \end{pmatrix} - 0.151 \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix} \begin{pmatrix} -3 \\ -16 \end{pmatrix} = \begin{pmatrix} 3.191 \\ -0.598 \end{pmatrix}$$

Linear CG

Theory (10.8)

Given a quadratic form f where $\mathbf{A}$ is symmetric, *N* is the number of dimensions, $\mathbf{P}_0$ is an arbitrary initial vector, and $\mathbf{h}_0 = \mathbf{g}_0 = \mathbf{b} - \mathbf{A}\mathbf{P}_0$.

Iterate with *i* until *N* iterations are complete.

1. Calculate $\lambda_i = \frac{\mathbf{g}_i^T \cdot \mathbf{g}_i}{\mathbf{h}_i^T \cdot \mathbf{A} \cdot \mathbf{h}_i}$
2. Calculate $\mathbf{P}_{i+1} = \mathbf{P}_i + \lambda_i \mathbf{h}_i$
3. Calculate $\mathbf{g}_{i+1} = \mathbf{g}_i - \lambda_i \mathbf{A} \cdot \mathbf{h}_i$
4. Calculate $\gamma_i = \frac{\mathbf{g}_{i+1}^T \cdot \mathbf{g}_{i+1}}{\mathbf{g}_i^T \cdot \mathbf{g}_i}$

The minimum of f is $\mathbf{P}_N$.

Example

$$\mathbf{A} = \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 2 \\ -8 \end{pmatrix}, \mathbf{P}_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\mathbf{h}_0 = \mathbf{g}_0 = \begin{pmatrix} 2 \\ -8 \end{pmatrix} - \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -3 \\ -16 \end{pmatrix}$$

$$\lambda_0 = \frac{\begin{pmatrix} -3 & -16 \end{pmatrix}\begin{pmatrix} -3 \\ -16 \end{pmatrix}}{\begin{pmatrix} -3 & -16 \end{pmatrix}\begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}\begin{pmatrix} -3 \\ -16 \end{pmatrix}} \approx 0.151$$

$$\mathbf{P}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 0.151\begin{pmatrix} -3 \\ -16 \end{pmatrix} = \begin{pmatrix} 0.547 \\ -1.416 \end{pmatrix}$$

$$\mathbf{g}_1 = \begin{pmatrix} -3 \\ -16 \end{pmatrix} - 0.151\begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}\begin{pmatrix} -3 \\ -16 \end{pmatrix} = \begin{pmatrix} 3.191 \\ -0.598 \end{pmatrix}$$

$$\gamma_0 = \frac{\begin{pmatrix} 3.191 & -0.598 \end{pmatrix}\begin{pmatrix} 3.191 \\ -0.598 \end{pmatrix}}{\begin{pmatrix} -3 & -16 \end{pmatrix}\begin{pmatrix} -3 \\ -16 \end{pmatrix}} \approx 0.04$$

Linear CG

Theory (10.8)

Given a quadratic form f where $\mathbf{A}$ is symmetric, N is the number of dimensions, $\mathbf{P}_0$ is an arbitrary initial vector, and $\mathbf{h}_0 = \mathbf{g}_0 = \mathbf{b} - \mathbf{A}\mathbf{P}_0$.

Iterate with i until N iterations are complete.

1. Calculate $\lambda_i = \frac{\mathbf{g}_i^T \cdot \mathbf{g}_i}{\mathbf{h}_i^T \cdot \mathbf{A} \cdot \mathbf{h}_i}$
2. Calculate $\mathbf{P}_{i+1} = \mathbf{P}_i + \lambda_i \mathbf{h}_i$
3. Calculate $\mathbf{g}_{i+1} = \mathbf{g}_i - \lambda_i \mathbf{A} \cdot \mathbf{h}_i$
4. Calculate $\gamma_i = \frac{\mathbf{g}_{i+1}^T \cdot \mathbf{g}_{i+1}}{\mathbf{g}_i^T \cdot \mathbf{g}_i}$
5. Calculate $\mathbf{h}_{i+1} = \mathbf{g}_{i+1} + \gamma_i \mathbf{h}_i$

The minimum of f is $\mathbf{P}_N$.

Example

$$\mathbf{A} = \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 2 \\ -8 \end{pmatrix}, \mathbf{P}_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\mathbf{h}_0 = \mathbf{g}_0 = \begin{pmatrix} 2 \\ -8 \end{pmatrix} - \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -3 \\ -16 \end{pmatrix}$$

$$\lambda_0 = \frac{\begin{pmatrix} -3 & -16 \end{pmatrix}\begin{pmatrix} -3 \\ -16 \end{pmatrix}}{\begin{pmatrix} -3 & -16 \end{pmatrix}\begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}\begin{pmatrix} -3 \\ -16 \end{pmatrix}} \approx 0.151$$

$$\mathbf{P}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 0.151\begin{pmatrix} -3 \\ -16 \end{pmatrix} = \begin{pmatrix} 0.547 \\ -1.416 \end{pmatrix}$$

$$\mathbf{g}_1 = \begin{pmatrix} -3 \\ -16 \end{pmatrix} - 0.151\begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}\begin{pmatrix} -3 \\ -16 \end{pmatrix} = \begin{pmatrix} 3.191 \\ -0.598 \end{pmatrix}$$

$$\gamma_0 = \frac{\begin{pmatrix} 3.191 & -0.598 \end{pmatrix}\begin{pmatrix} 3.191 \\ -0.598 \end{pmatrix}}{\begin{pmatrix} -3 & -16 \end{pmatrix}\begin{pmatrix} -3 \\ -16 \end{pmatrix}} \approx 0.04$$

$$\boldsymbol{h}_1 = \begin{pmatrix} 3.191 \\ -0.598 \end{pmatrix} + 0.04\begin{pmatrix} -3 \\ -16 \end{pmatrix} = \begin{pmatrix} 3.071 \\ -1.238 \end{pmatrix}$$

Beyond Linearity

- Linear CG is **pretty good**! It finds conjugate search directions and performs line minimization **'in house'.**
- However, it requires the full **Hessian matrix** ($\mathbf{A}$) to be known and stored in memory.
- This means linear CG:
  - Only works on **quadratic forms**, which violates our objective to be able to minimize **any differentiable function**.
  - Has high space complexity.

How so Linearity?

- Linear CG methods are dubbed so because they are applied to quadratic forms, where the gradient is a **linear function of $\boldsymbol{x}$**:

$$\nabla f = \mathbf{A} \cdot \mathbf{x} - \mathbf{b}$$

- Now, we turn to **nonlinear** CG methods, which except any function where the gradient can be obtained.
- But how can we use CG without knowing $\mathbf{A}$?

What if I told you...

We only need 4 changes to drop **A** entirely.

Fletcher-Reeves Nonlinear CG

01 Initialization

Old: $\mathbf{h}_0 = \mathbf{g}_0 = \mathbf{b} - \mathbf{A}\mathbf{P}_0$
New: $\mathbf{h}_0 = \mathbf{g}_0 = -\nabla f(\mathbf{P}_0)$

02 Line Search

Use line search (like Dbrent) to find λ_i that minimizes $f(\mathbf{P}_i + \lambda_i \mathbf{h}_i)$

03 Gradient Calculation

Old: $\mathbf{g}_{i+1} = \mathbf{g}_i - \lambda_i \mathbf{A} \cdot \mathbf{h}_i$
New: $\mathbf{g}_{i+1} = -\nabla f(\mathbf{P}_{i+1})$

04 Lifecycle

Minimization stops at convergence instead of N iterations.

Compare & Contrast

Linear

Given a quadratic form f where $\mathbf{A}$ is symmetric, *N* is the number of dimensions, $\mathbf{P}_0$ is an arbitrary initial vector, and $\mathbf{h}_0 = \mathbf{g}_0 = \mathbf{b} - \mathbf{A}\mathbf{P}_0$.

Iterate with *i* until *N* iterations are complete.

1. Calculate $\lambda_i = \frac{\mathbf{g}_i^T \cdot \mathbf{g}_i}{\mathbf{h}_i^T \cdot \mathbf{A} \cdot \mathbf{h}_i}$
2. Calculate $\mathbf{P}_{i+1} = \mathbf{P}_i + \lambda_i \mathbf{h}_i$
3. Calculate $\mathbf{g}_{i+1} = \mathbf{g}_i - \lambda_i \mathbf{A} \cdot \mathbf{h}_i$
4. Calculate $\gamma_i = \frac{\mathbf{g}_{i+1}^T \cdot \mathbf{g}_{i+1}}{\mathbf{g}_i^T \cdot \mathbf{g}_i}$
5. Calculate $\mathbf{h}_{i+1} = \mathbf{g}_{i+1} + \gamma_i \mathbf{h}_i$

The minimum of f is $\mathbf{P}_N$.

Nonlinear

Given a function f where $f(x)$ and $f'(x)$ can be evaluated, *N* is the number of dimensions, $\mathbf{P}_0$ is an arbitrary initial vector, and

$$\mathbf{h}_0 = \mathbf{g}_0 = -\nabla f(\mathbf{P}_0)$$

Iterate with *i* until convergence.

1. Find λ_i that minimizes $f(\mathbf{P}_i + \lambda_i \mathbf{h}_i)$
2. Calculate $\mathbf{P}_{i+1} = \mathbf{P}_i + \lambda_i \mathbf{h}_i$
3. Calculate $\mathbf{g}_{i+1} = -\nabla f(\mathbf{P}_{i+1})$
4. Calculate $\gamma_i = \frac{\mathbf{g}_{i+1}^T \cdot \mathbf{g}_{i+1}}{\mathbf{g}_i^T \cdot \mathbf{g}_i}$
5. Calculate $\mathbf{h}_{i+1} = \mathbf{g}_{i+1} + \gamma_i \mathbf{h}_i$

The minimum of f is the final value of $\mathbf{P}$.

Benefits of Fletcher-Reeves

- **Singnificantly** improves applicability.
- Knowledge of the **Hessian matrix** (**A**) and the storage necessary to store it are **not required**.

But what if I told you...

It can converge **even faster.**

Polak-Ribiere

A one-line code change (literally) can make Fletcher-Reeves converge **even faster** on non-quadratic form functions.

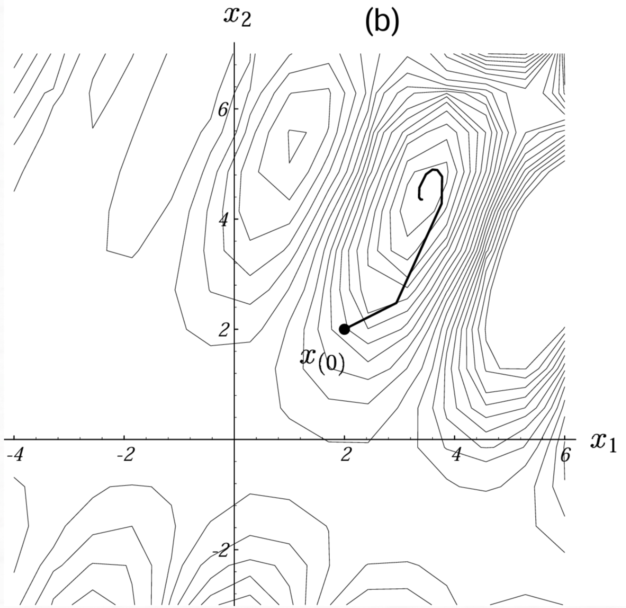
$$\gamma_i = \frac{\mathbf{g}_{i+1} \cdot \mathbf{g}_{i+1}}{\mathbf{g}_i \cdot \mathbf{g}_i}$$

Polak-Ribiere

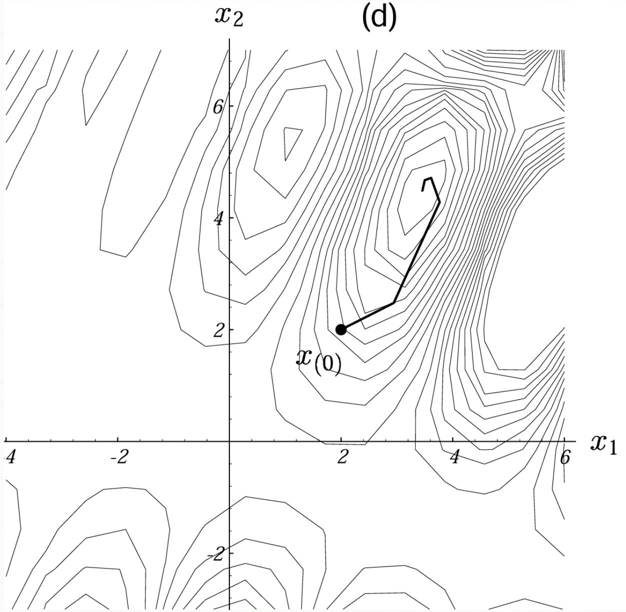
A one-line code change (literally) can make Fletcher-Reeves converge **even faster** on non-quadratic form functions.

$$\gamma_i = \frac{\mathbf{g}_{i+1} \cdot \mathbf{g}_{i+1}}{\mathbf{g}_i \cdot \mathbf{g}_i} \longrightarrow \gamma_i = \frac{(\mathbf{g}_{i+1} - \mathbf{g}_i) \cdot \mathbf{g}_{i+1}}{\mathbf{g}_i \cdot \mathbf{g}_i}$$

Convergence, Visualized



Fletcher-Reeves



Polak-Ribiere

But what if I told you there's a last frontier of CG expansion...

Quasi-Newton Methods

Until then...

Thanks!

Joshua Sheldon
jsheldon2022@my.fit.edu

Additional Resources

- Nonlinear conjugate gradient method (Wikipedia)
- An Introduction to the Conjugate Gradient Method Without the Agonizing Pain (Edition 1¼)
- Introduction to Mathematical Optimization with Python (Chapter 5: Conjugate Gradient Methods)
- Taylor series (Wikipedia)
- Taylor series | Chapter 11, Essence of calculus (3Blue1Brown)