Sparse Matrices — Conjugate Gradient Method
Matrix Computations — CPSC 5006 E

Julien Dompierre

Department of Mathematics and Computer Science
Laurentian University

Sudbury, December 6, 2010

Sparse Matrices — Conjugate Gradient Method

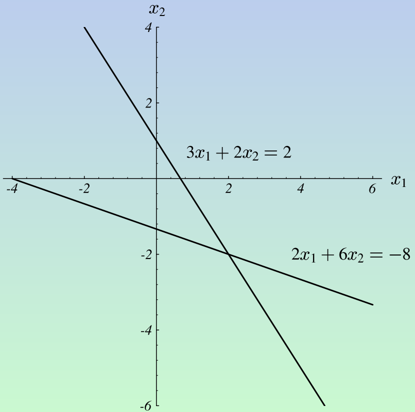
- Quadratic Form.
- Steepest Decent.
- Conjugate Gradients.
- Preconditioning.
- **Shewchuk. An Introduction to the Conjugate Gradient Method Without the Agonizing Pain**.
- Golub-Van Loan, sections 10.2 and 10.3.
- Saad. Iterative Methods for Sparse Linear System, §6.7
- Barrett et al. Templates for the Solution of LS, §2.3.1
- M. R. Hestenes and E. Stiefel. "Methods of Conjugate Gradients for Solving Linear Systems", *J. Res. Nat. Bur. Stand.*, 49, 409–436, 1952.

"When I decided to learn the Conjugate Gradient Method (henceforth, CG), I read four different descriptions, which I shall politely not identify. I understood none of them. Most of them simply wrote down the method, then proved its properties without any intuitive explanation or hint of how anybody might have invented CG in the first place. This article was born of my frustration, with the wish that future students of CG will learn a rich and elegant algorithm, rather than a confusing mass of equations."

Jonathan Richard Shewchuk

All examples and pictures are related to this simple 2D example

$$A = \begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ -8 \end{bmatrix}, \quad c = 0.$$

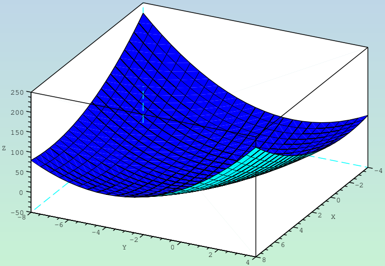


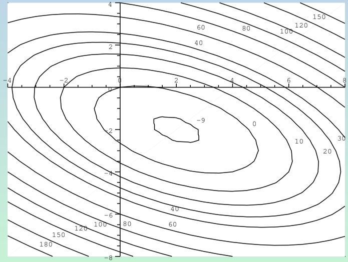
Quadratic Form

A **quadratic form** is simply a scalar, quadratic function of a vector with the form

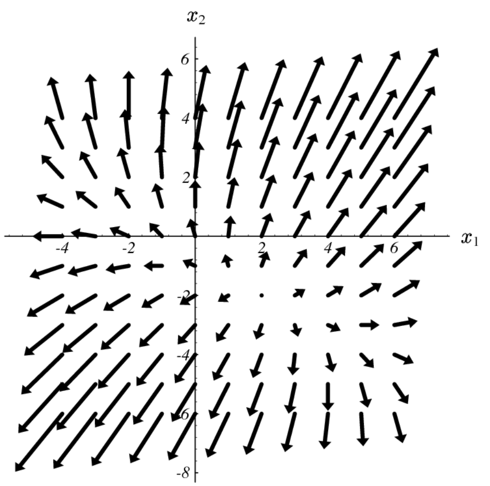
$$f(x) = \frac{1}{2}x^T Ax - b^T x + c$$

where A is a matrix, x and b are vectors, and c is a scalar constant.





For every point x, the gradient points in the direction of steepest increase of $f(x)$, and is orthogonal to the contour lines.



Gradient of Quadratic Form and Solution of Linear System

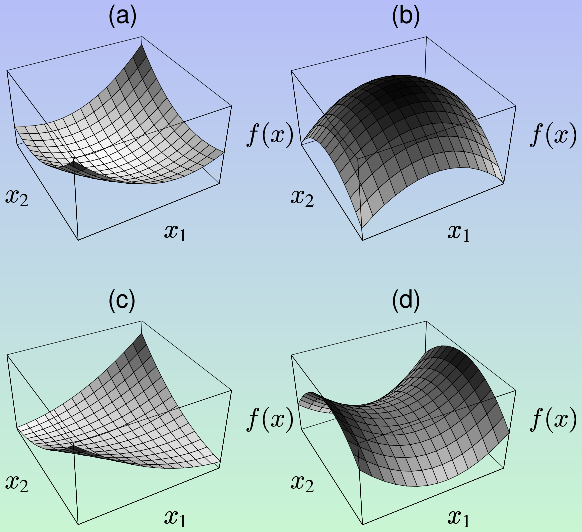
The gradient of the quadratic form is given by

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

if A is symmetric. Therefore, if x is solution of the linear system $Ax = b$, then $\nabla f(x)$ is equal to 0 and x is a critical point of $f(x)$.

If A is positive definite, as well as symmetric, then this critical point is a minimum of $f(x)$. So $Ax = b$ can be solved by finding an x that minimizes $f(x)$.

Quadratic Forms



Some Definitions

In the method of **Steepest Descent**, we start at an arbitrary point $x_{(0)}$ and slide down to the bottom of the paraboloid. We take a series of steps $x_{(1)}, x_{(2)}, \ldots$, until we are satisfied that we are close enough to the solution x.

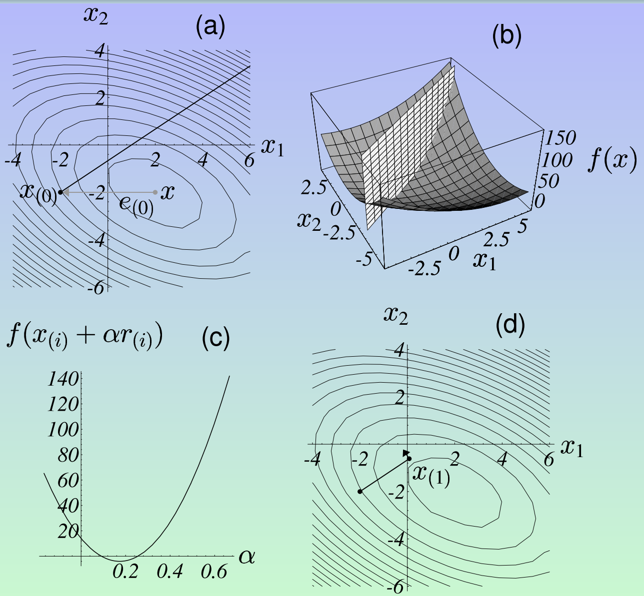
When we take a step, we choose the direction in which f decreases most quickly, which is the direction opposite to $\nabla f(x_{(i)})$, which is $-\nabla f(x_{(i)}) = -(Ax_{(i)} - b) = b - Ax_{(i)}$.

The **error** $e_{(i)} = x_{(i)} - x$ is a vector that indicates how far we are from the solution.

The **residual** $r_{(i)} = b - Ax_{(i)}$ indicates how far we are from the correct value of b. It is easy to see that $r_{(i)} = -Ae_{(i)}$, and you should think of the residual as being the error transformed by A into the same space as b.

More importantly, $r_{(i)} = -\nabla f(x_{(i)})$, and you should also think of the "**residual**" as the "**direction of steepest descent**."

Steepest Descent and Line Search

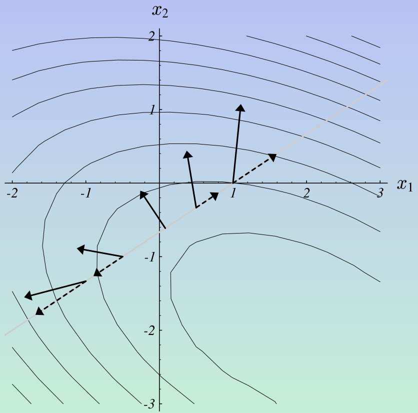


Orthogonality of the Residuals

Starting at $x_{(0)}$ in the direction of steepest descent $r_{(0)}$, choose the point $x_{(1)}$ given by

$$x_{(1)} = x_{(0)} + \alpha r_{(0)}$$

for some coefficient α. The value of α is set such that the next direction of steepest descent $r_{(1)}$ is perpendicular to the current direction of steepest descent $r_{(0)}$.



Computation of α

$$
\begin{aligned}
r_{(1)}^T r_{(0)} &= 0 \\
(b - Ax_{(1)})^T r_{(0)} &= 0 \\
(b - A(x_{(0)} + \alpha r_{(0)}))^T r_{(0)} &= 0 \\
(b - Ax_{(0)})^T r_{(0)} - \alpha (Ar_{(0)})^T r_{(0)} &= 0 \\
(b - Ax_{(0)})^T r_{(0)} &= \alpha (Ar_{(0)})^T r_{(0)} \\
r_{(0)}^T r_{(0)} &= \alpha r_{(0)}^T (Ar_{(0)}) \\
\alpha &= \frac{r_{(0)}^T r_{(0)}}{r_{(0)}^T A r_{(0)}}
\end{aligned}
$$

A Naive Steepest Descent Algorithm (p. 521)

Algorithm 1 Steepest Descent. Let $A \in \mathbb{R}^{n \times n}$ be a symmetric positive definite matrix, $b \in \mathbb{R}^n$, the right-hand side, and $x^{(0)}$, an initial guess. This algorithm finds the solution x of the system $Ax = b$ with the iterative steepest descent algorithm.

$r_{(0)} = b - Ax_{(0)}$
$k = 0$
while $r_k \neq 0$
    $k = k + 1$
    $\alpha_{(k)} = (r_{(k-1)}^T r_{(k-1)}) \,/\, (r_{(k-1)}^T A r_{(k-1)})$
    $x_{(k)} = x_{(k-1)} + \alpha_{(k)} r_{(k-1)}$
    $r_{(k)} = b - Ax_{(k)}$
end

This algorithm requires two matrix-vector multiplications per iteration, namely $Ar_{(k-1)}$ and $Ax_{(k)}$.

We can remove the matrix-vector multiplication $Ax_{(k)}$ of line 7 in the previous algorithm as follows:

$$
\begin{aligned}
x_{(k)} &= x_{(k-1)} + \alpha_{(k)} r_{(k-1)} \quad \text{(line 6)} \\
-Ax_{(k)} + b &= -A(x_{(k-1)} + \alpha_{(k)} r_{(k-1)}) + b \\
r_{(k)} &= -Ax_{(k-1)} + b - \alpha_{(k)} Ar_{(k-1)} \\
r_{(k)} &= r_{(k-1)} - \alpha_{(k)} Ar_{(k-1)}
\end{aligned}
$$

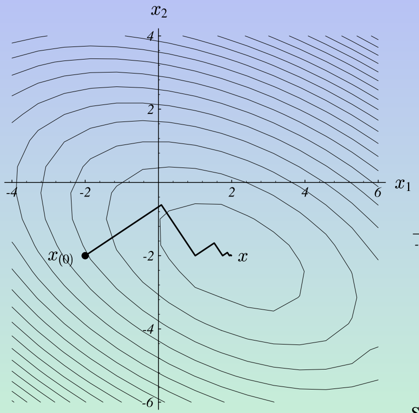
We can store the result of the matrix multiplication $Ar_{(k-1)}$ from line 5 and reuse this result to evaluate $r_{(k)}$ at line 7. The disadvantage of using this reccurence relation to avoid $Ax_{(k)}$ is that the sequence is generated without any feedback from the values $x_{(k)}$, so the floating point roundoff error may accumulate. This can be avoid by periodically using the initial equation $r_{(k)} = b - Ax_{(k)}$.

Steepest Descent Algorithm

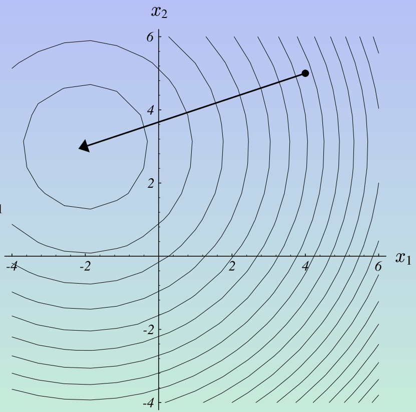
Algorithm 2 Steepest Descent. Let $A \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix, $b \in \mathbb{R}^n$, the right-hand side, $x^{(0)}$, an initial guess, ε a stopping criterion, *MaxIter* the maximum number of iterations, and *update* $= \lceil \sqrt{n} \rceil$ the interval between residual updates.

$k = 0$
$r_{(0)} = b - Ax_{(0)}, \quad \delta_{(0)} = r_{(0)}^T r_{(0)}$
while $k < MaxIter$ and $\delta_{(k)} > \varepsilon^2 \delta_{(0)}$
  $q = Ar_{(k)}, \quad \alpha_{(k)} = \delta_{(k)} \,/\, r_{(k)}^T q$
  $x_{(k+1)} = x_{(k)} + \alpha_{(k)} r_{(k)}$
  if k is divisible by *update* **then**
    $r_{(k+1)} = b - Ax_{(k+1)}$
  else
    $r_{(k+1)} = r_{(k)} - \alpha q$
  end
  $k = k + 1, \quad \delta_k = r_{(k)}^T r_{(k)}$
end

Steepest Descent



The method of Steepest Descent.



Steepest Descent converges to the exact solution on the first iteration if the eigenvalues are all equal.

Convergence of the Steepest Descent (p 521)

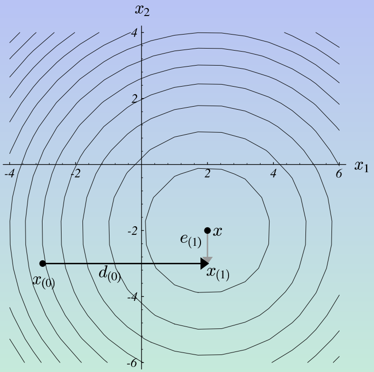
With the 2-norm, remind that the **condition number** of a matrix A is given by $\kappa_2(A) = \lambda_{\max}/\lambda_{\min}$.

We define also the **energy norm** $\|e\|_A = (e^T A e)^{1/2}$.

Then the convergence result for the steepest descent is

$$\|e_{(k)}\|_A \leq \left(\frac{\kappa_2 - 1}{\kappa_2 + 1} \right)^k \|e_{(0)}\|_A.$$

Steepest descent often finds itself taking steps in the same direction as earlier steps. Wouldn't it be better if, every time we took a step, we got it right the first time? Here's an idea: let's pick a set of orthogonal **search directions** $d_{(0)}, d_{(1)}, \ldots, d_{(n-1)}$. In each search direction, we'll take exactly one step, and that step will be just the right length to line up evenly with x. After n steps, we'll be done.

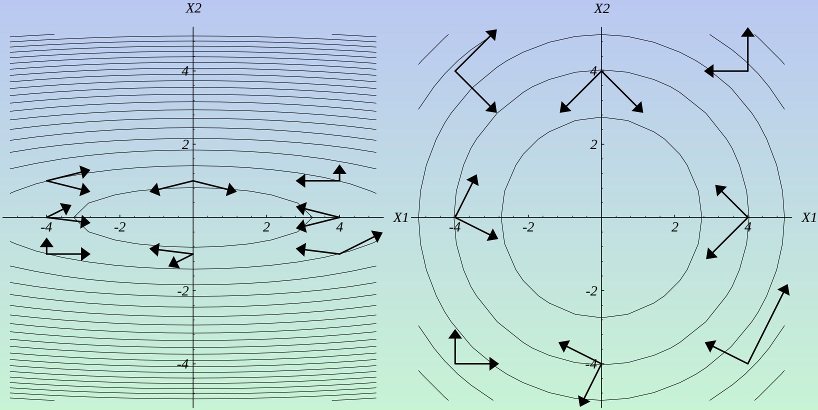


The Method of Orthogonal Directions.

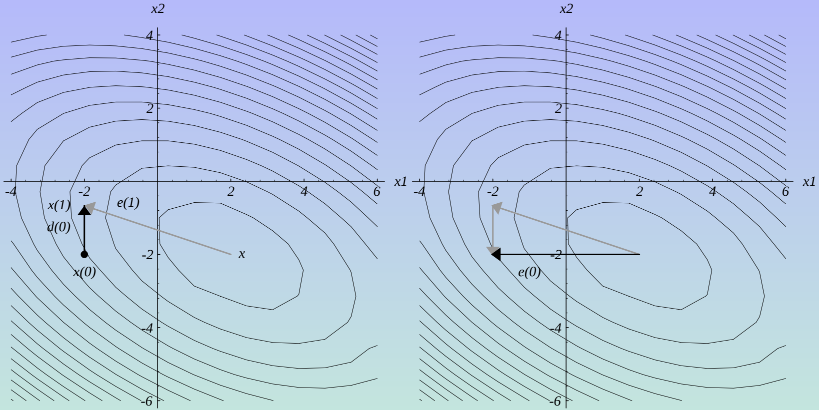
Conjugate Directions

Two vectors $d_{(i)}$ and $d_{(j)}$ are A-**orthogonal**, or **conjugate**, if

$$(d_{(i)}, d_{(j)})_A = (d_{(i)}, Ad_{(j)}) = d_{(i)}^T A d_{(j)} = 0.$$



These pairs of vectors are A-orthogonal … … because these pairs of vectors are orthogonal.



The method of conjugate directions converges in n steps. The first step is taken along some direction $d_{(0)}$. The minimum point $x_{(1)}$ is chosen by the constraint that $e_{(1)}$ must be A-orthogonal to $d_{(0)}$. The initial error $e_{(0)}$ can be expressed as a sum of A-orthogonal components (gray arrows). Each step of conjugate directions eliminates one of these components.

Conjugate Gradient Algorithm (p. 527)

Algorithm 3 Conjugate Gradients. Let $A \in \mathbb{R}^{n \times n}$ be a symmetric positive definite matrix, $b \in \mathbb{R}^n$, the right-hand side, $x^{(0)}$, an initial guess, ε a stopping criterion, *MaxIter* the maximum number of iterations, and $update = \lceil \sqrt{n} \rceil$ the interval between residual updates.

$k = 0, \quad r_{(0)} = b - Ax_{(0)}$
$d_{(0)} = r_{(0)}, \quad \delta_{(0)} = r_{(0)}^T r_{(0)}$
while $k < MaxIter$ and $\delta_{(k)} > \varepsilon^2 \delta_{(0)}$
  $q = Ad_{(k)}, \quad \alpha = \delta_{(k)} \,/\, d_{(k)}^T q, \quad x_{(k+1)} = x_{(k)} + \alpha d_{(k)}$
  if k is divisible by *update* **then**
    $r_{(k+1)} = b - Ax_{(k+1)}$
  else
    $r_{(k+1)} = r_{(k)} - \alpha q$
  end
  $\delta_{(k+1)} = r_{(k+1)}^T r_{(k+1)}, \quad \beta = \delta_{(k+1)} / \delta_{(k)}$
  $d_{(k+1)} = r_{(k+1)} + \beta d_{(k)}, \quad k = k + 1$
end

Convergence

Suppose we want $\|e_{(k)}\| \le \varepsilon \|e_{(0)}\|$, then the number of iterations k for the steepest descent is

$$k \le \left\lceil \frac{1}{2}\kappa(A) \ln\left(\frac{1}{\varepsilon}\right) \right\rceil,$$

and for the conjugate gradient is

$$k \le \left\lceil \frac{1}{2}\sqrt{\kappa(A)} \ln\left(\frac{2}{\varepsilon}\right) \right\rceil.$$