Conjugate Gradient

Hamad El Kahza

October 23, 2020

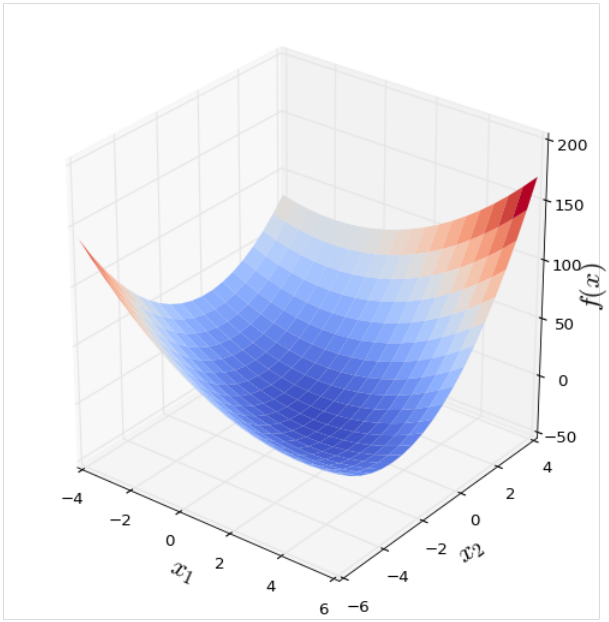
1 Definitions:

CG is a popular method in solving systems of the form Ax=B. To find x, we would like to have a function of x, whose global minimum is the solution. Such function can, for instance, be the following quadratic form:

$$f(x) = \frac{1}{2}x^T Ax - b^T x + c \tag{3}$$

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

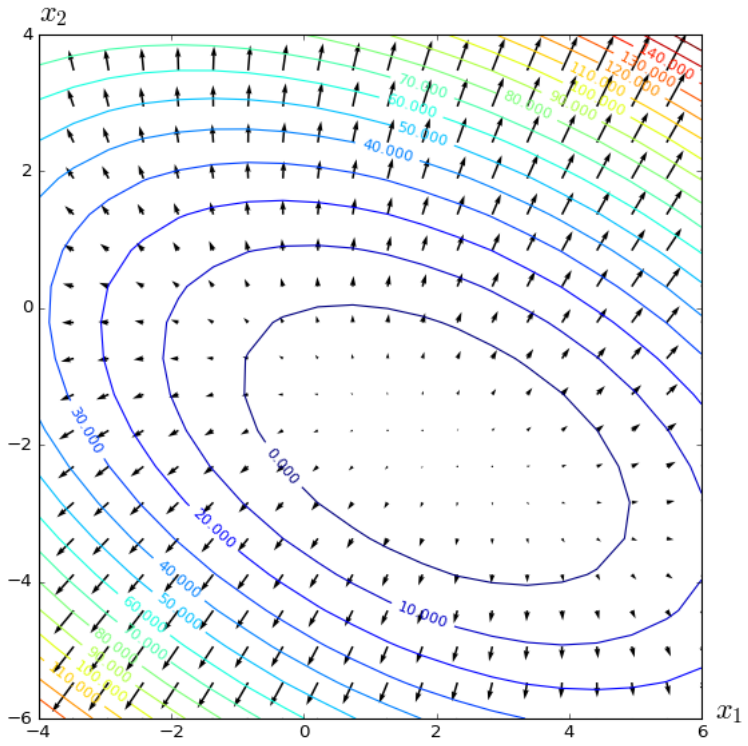
- if A is symmetric $A^T = A$ and positive-definite ($x^T Ax > 0$), $f(x)$ is minimized by the solution to $Ax = b$.
- Because A is positive-definite, the surface defined by $f(x)$ is shaped like a paraboloid bowl.



The gradient of the f is express as follows:

$$f'(x) = \begin{bmatrix} \frac{\partial}{\partial x_1} f(x) \\ \frac{\partial}{\partial x_2} f(x) \\ \vdots \\ \frac{\partial}{\partial x_n} f(x) \end{bmatrix}. \tag{5}$$

The gradient is a vector field that, for a given point x, points in the direction of greatest increase of $f(x)$. We can minimize $f(x)$ by setting $f'(x)$ equal to zero.

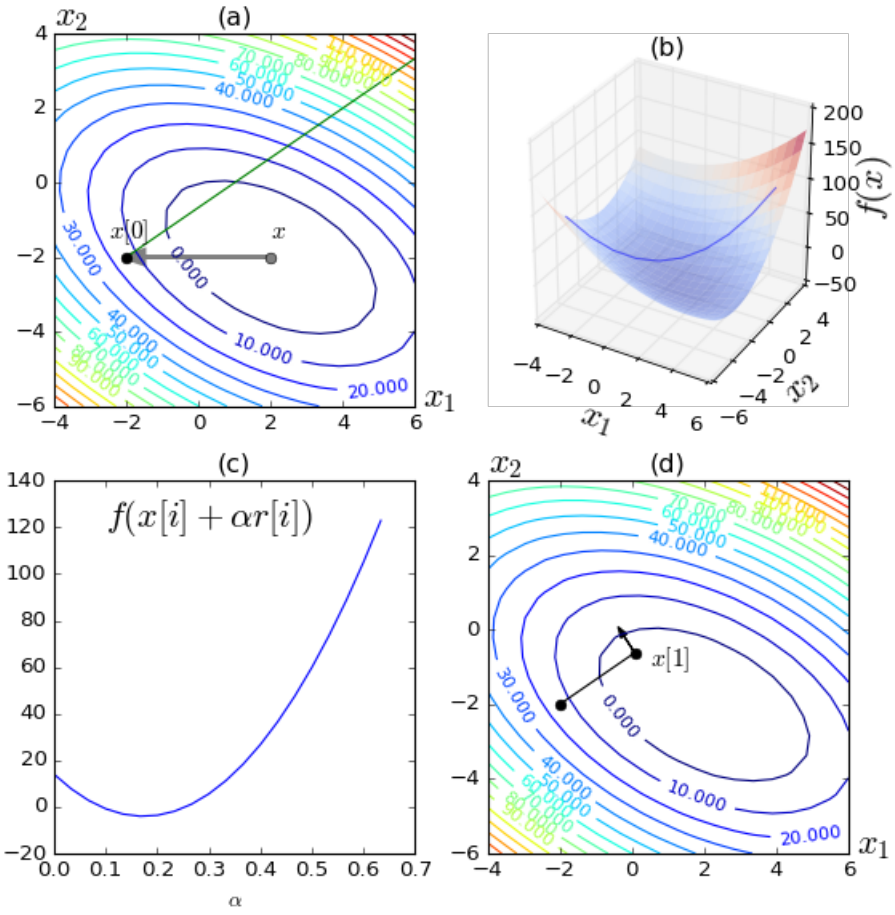


2 Method of steepest descent

- In the method of Steepest Descent, we start at an arbitrary point $x_{[0]}$ and slide down to the bottom of the paraboloid.
- When we take a step, we choose the direction in which f decreases most quickly, which is the direction opposite to $f'(x_{[i]})$. This direction is $-f'(x_{[i]}) = b - Ax_{[i]}$.
- Definitions:
  - The error $e_{[i]} = x_{[i]} - x$
  - The residual $r_{[i]} = b - Ax_{[i]}$
  - We obtain $r_{[i]} = -Ae_{[i]}$
  - $r_{[i]} = -f'(x_{[i]})$ residual is the direction of steepest descent.

How big should be the step taken:

A line search is a procedure that chooses α to minimize f along a line.
α minimizes f when the directional derivative $\frac{\partial}{\partial \alpha} f(x_{[1]})$ is equal to zero.
By the chain rule, $\frac{\partial}{\partial \alpha} f(x_{[1]}) = f'(x_{[1]})^T \frac{\partial}{\partial \alpha} x_{[1]} = f'(x_{[1]})^T r_{[0]}$
Setting this expression to zero, we find that α should be chosen so that $r_{[0]}$ and $f'(x_{[1]})$ are orthogonal.



The slope of the parabola at any point is equal to the magnitude of the projection of the gradient onto the line. These projections represent the rate of increase we traverses the search line. f is minimized where the projection is zero — where the gradient is orthogonal to the search line.
We determine alpha using the following derivation, given $f'(x_{[1]}) = -r_{[1]}$:

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

The method of Steepest Descent is then summarized as follows:

$$r_{[i]} = b - Ax_{[i]}, \tag{10}$$

$$\alpha_{[i]} = \frac{r_{[i]}^T r_{[i]}}{r_{[i]}^T A r_{[i]}}, \tag{11}$$

$$x_{[i+1]} = x_{[i]} + \alpha_{[i]} r_{[i]}. \tag{12}$$

The example is run until it converges in the figure below. Each gradient is orthogonal to the previous gradient.

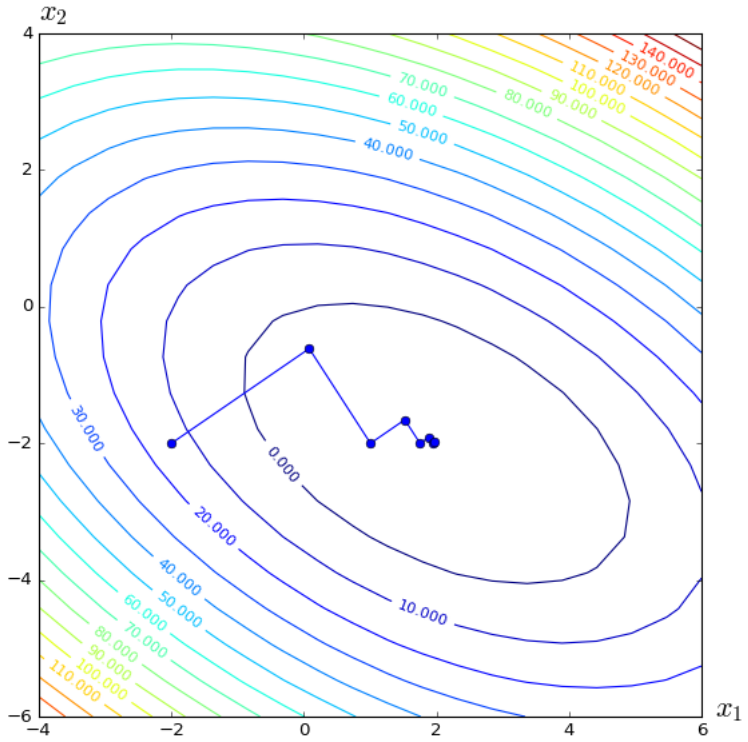


Figure 1: Method of steepest descent. The iterations starts at $[-2, -2]^T$ and converges at $[2, -2]^T$

3 Importance of Eigen-vectors:

- Each eigenvector has a corresponding eigenvalue, denoted $\lambda_1, \lambda_2, \ldots, \lambda_n$. These are uniquely defined for a given matrix.
- a vector x is illustrated as a sum of two eigenvectors v_1 and v_2. Applying B to x is equivalent to applying B to the eigenvectors, and summing the result. On repeated application, we have $B^i x = B^i v_1 + B^i v_2 = \lambda_1^i v_1 + \lambda_2^i v_2$
- For any constant α, the vector αv is also an eigenvector with eigenvalue λ, because $B(\alpha v) = \alpha B v = \lambda \alpha v$. In other words, if you scale an eigenvector, it's still an eigenvector.
- If $|\lambda| < 1$, then $B^i v = \lambda^i v$ will vanish as i approaches infinity . If $|\lambda| > 1$, then will grow to infinity.

4 The Jacobi Method:

In order to solve for $Ax = b$, The matrix is split into two parts:

- D, whose diagonal elements are identical to those of A, and whose off-diagonal elements are zero
- D, whose diagonal elements are identical to those of A, and whose off-diagonal elements are zero; and E, whose diagonal elements are zero, and whose off-diagonal elements are identical to those of A
- $A = D + E$

We derive the Jacobi Method

$$\begin{aligned} Ax &= b \\ Dx &= -Ex + b \\ x &= -D^{-1}Ex + D^{-1}b \\ x = Bx + z, \qquad &\text{where } B = -D^{-1}E, \qquad z = D^{-1}b. \end{aligned} \tag{14}$$

Because D is diagonal, it is easy to invert. This identity can be converted into an iterative method by forming the recurrence

$$x_{[i+1]} = Bx_{[i]} + z. \tag{1}$$

We then express each iterate $x_{[i]}$ as the sum of the exact solution x and the error term $e_{[i]}$:

$$\begin{aligned} x_{[i+1]} &= Bx_{[i]} + z \\ &= B(x + e_{[i]}) + z \\ &= Bx + z + Be_{[i]} \\ &= x + Be_{[i]} \\ \therefore\ e_{[i+1]} &= Be_{[i]}. \end{aligned}$$

- If $\rho(B) < 1$, then the error term $e_{[i]}$ will converge to zero as i approaches infinity.
- The choice of $x_{[0]}$ does affect the number of iterations required to converge to x within a given tolerance.
- the spectral radius $\rho(B)$ determines the speed of convergence.

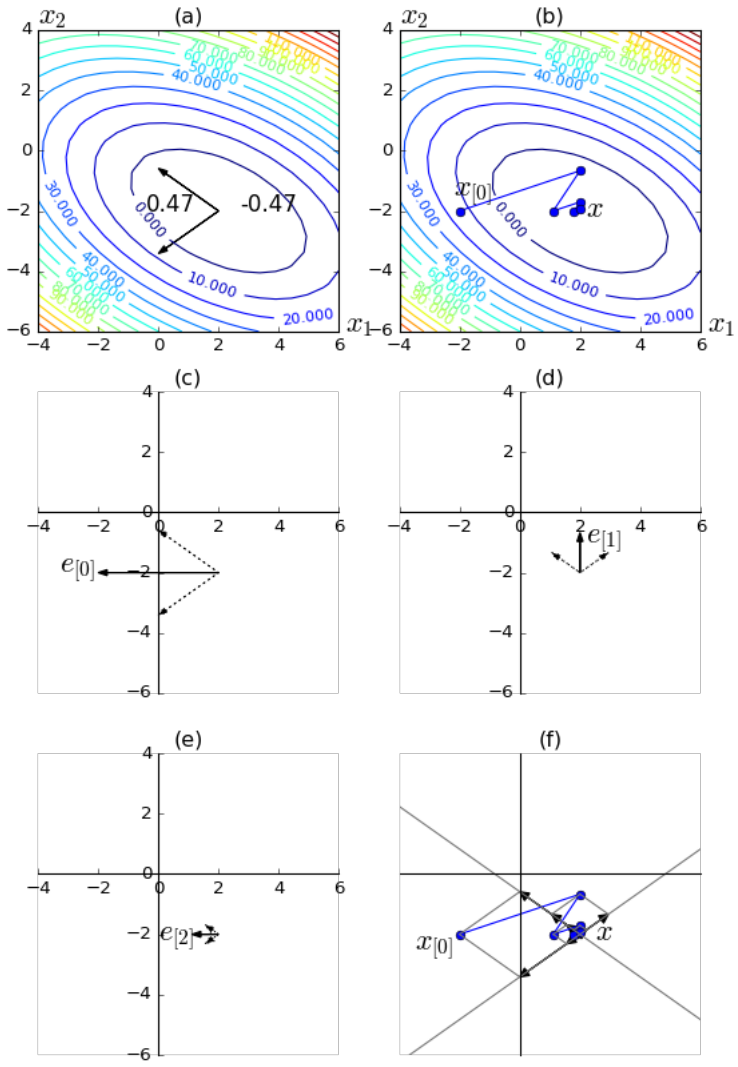


Figure 2: Jacobi Method. At every step, the eigen vector of the error term are computer determining the directions of the next step.

5 Method of Conjugate directions:

- The method of Steepest descent often takes steps in the same direction as previous steps. **It would be better, every time we took a step, we got it right the first time**
- Assume 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.

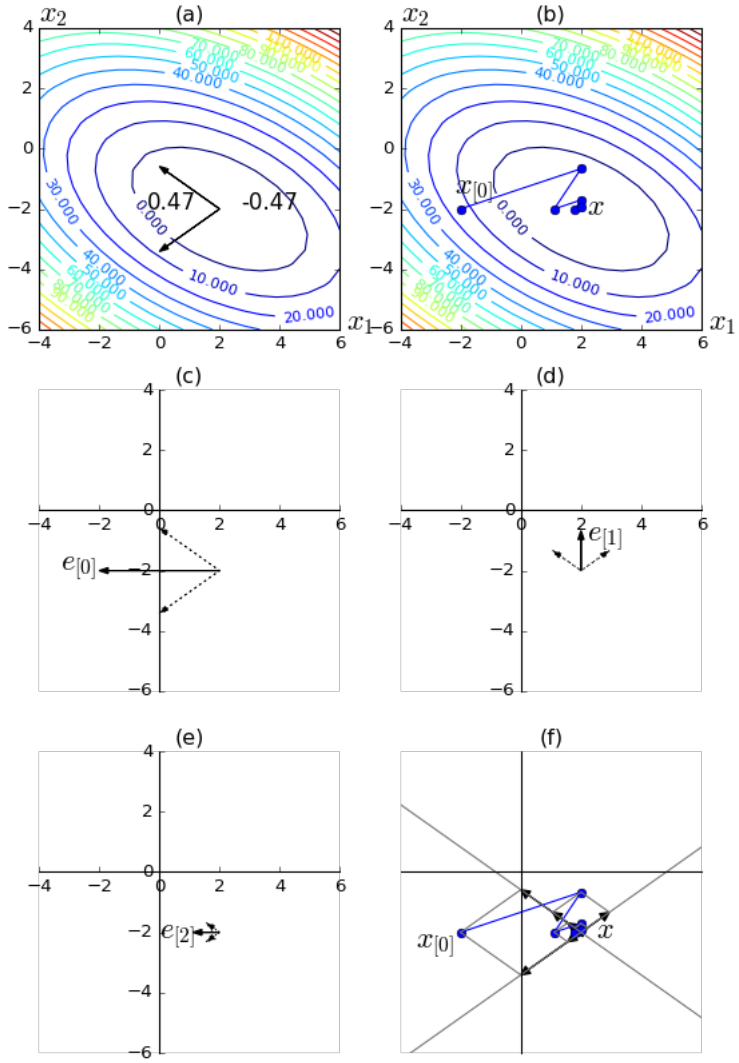


Figure 3: The Method of Orthogonal Directions. Unfortunately, this method only works if you already know the exact answer of x

$e_{[1]}$ is orthogonal to $d_{[0]}$. Therefore, for each step we choose a point

$$x_{[i+1]} = x_{[i]} + \alpha_{[i]} d_{[i]}. \tag{29}$$

In order to find the value of $\alpha_{[i]}$, we choose $e_{[i+1]}$ to be orthogonal to $d_{[i]}$, so that we need never step in the direction of $d_{[i]}$ again. Therefore:

$$\begin{aligned} d_{[i]}^T e_{[i+1]} &= 0 \\ d_{[i]}^T (e_{[i]} + \alpha_{[i]} d_{[i]}) &= 0 \\ \alpha_{[i]} &= -\frac{d_{[i]}^T e_{[i]}}{d_{[i]}^T d_{[i]}}. \end{aligned} \tag{30}$$

- The limitation is we still don't know the error term $e_{[i]}$ to compute $\alpha_{[i]}$.
- The solution is to make the search directions A-orthogonal instead of orthogonal. Two vectors $d_{[i]}$ and $d_{[j]}$ are A-orthogonal, or conjugate, if

$$d_{[i]}^T A d_{[j]} = 0.$$

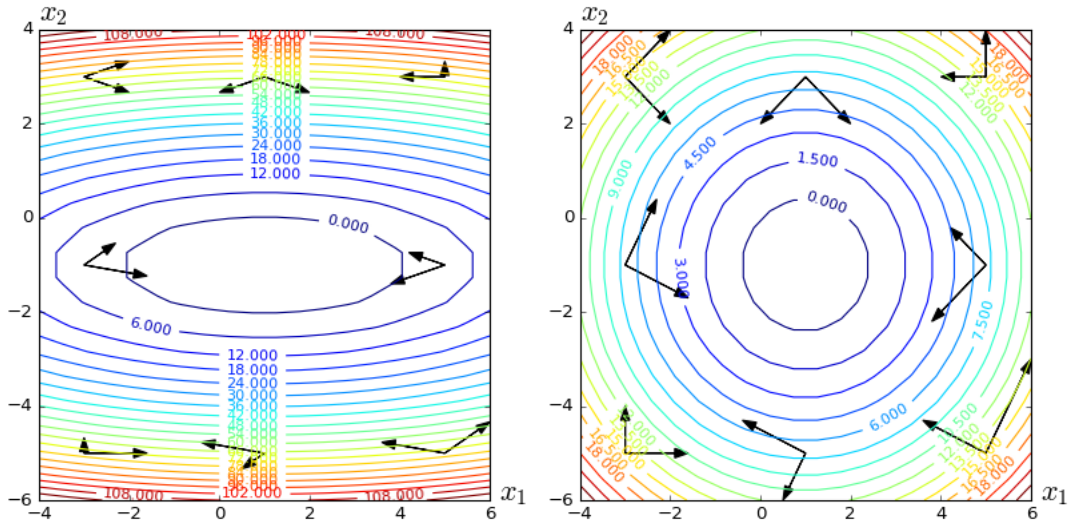


Figure 4: The pairs of vectors in first graph are A-orthogonal, because the pairs of vectors in second graph are orthogonal.

The method of Conjugate Directions converges in n steps. (a) 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 $d_{[0]}$. (b) The initial error $e_{[0]}$ can be expressed as a sum of A-orthogonal components. Each step of Conjugate Directions eliminates one of these components.

- 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. Each step of Conjugate Directions eliminates one of these components.
- $$\alpha_{[i]} = -\frac{d_{[i]}^T A e_{[i]}}{d_{[i]}^T A d_{[i]}} \tag{31}$$
$$= -\frac{d_{[i]}^T r_{[i]}}{d_{[i]}^T A d_{[i]}}. \tag{32}$$

  Note: If the search vector were the residual, this formula would be identical to the formula used by Steepest Descent.

6 Gram-Schmidt Conjugation

In order to not step in the same previous direction in the search process, we need a set of A-orthogonal search directions $d_{[i]}$. A simple way to generate them is conjugate Gram-Schmidt process.
Suppose we have a set of n linearly independent vectors $\mu_0, \mu_1, \ldots, \mu_{n-1}$:
set $d_{[0]} = \mu_0$ and for $i > 0$, set

$$d_{[i]} = \mu_i + \sum_{k=0}^{i-1} \beta_{ik} d_{[k]}, \tag{2}$$

where the β_{ik} are defined for $i > k$:

$$\begin{aligned} d_{[i]}^T A d_{[j]} &= \mu_i^T A d_{[j]} + \sum_{k=0}^{i-1} \beta_{ik} d_{[k]}^T A d_{[j]} \\ 0 &= \mu_i^T A d_{[j]} + \beta_{ij} d_{[j]}^T A d_{[j]}, \ i > j \quad \text{(by } A\text{-orthogonality of } d \text{ vectors)} \\ \beta_{ij} &= -\frac{\mu_i^T A d_{[j]}}{d_{[j]}^T A d_{[j]}} \end{aligned}$$

The limitation of using Gram-Schmidt conjugation in the method of Conjugate Directions is that:

- all the old search vectors must be kept in memory to construct each new one
- $\mathcal{O}(n^3)$ operations are required to generate the full set

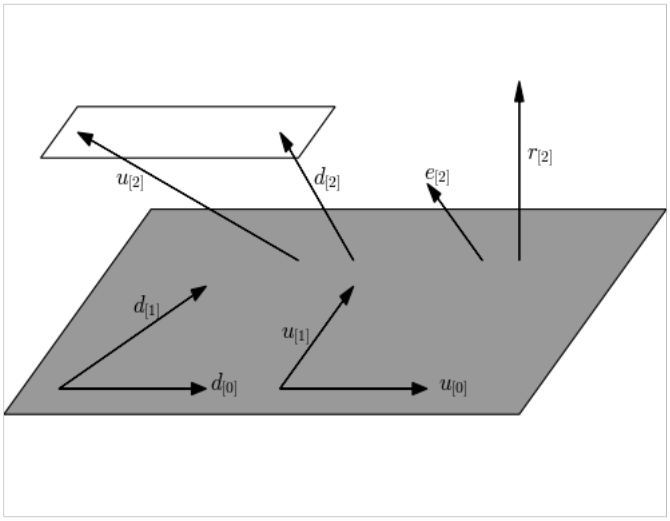


Figure 5: Because the search directions $d_{[0]}, d_{[1]}$ are constructed from the vectors μ_0, μ_1, they span the same subspace $\mathcal{D}_2$ (the gray-colored plane). The error term $e_{[2]}$ is A-orthogonal to $\mathcal{D}_2$, the residual $r_{[2]}$ is orthogonal to $\mathcal{D}_2$, and a new search direction $d_{[2]}$ is constructed (from μ_2) to be A-orthogonal to $\mathcal{D}_2$. The endpoints of μ_2 and $d_{[2]}$ lie on a plane parallel to $\mathcal{D}_2$, because $d_{[2]}$ is constructed from μ_2 by Gram-Schmidt conjugation.

7 Conjugate Gradient:

Conjugate gradient is similar to Conjugate directions, except the search directions are constructed by conjugation of the residual for the following reasons:

- First, the residuals worked for Steepest Descent
- residual is orthogonal to the previous search directions, so it's guaranteed always to produce a new, linearly independent search direction unless the residual is zero.

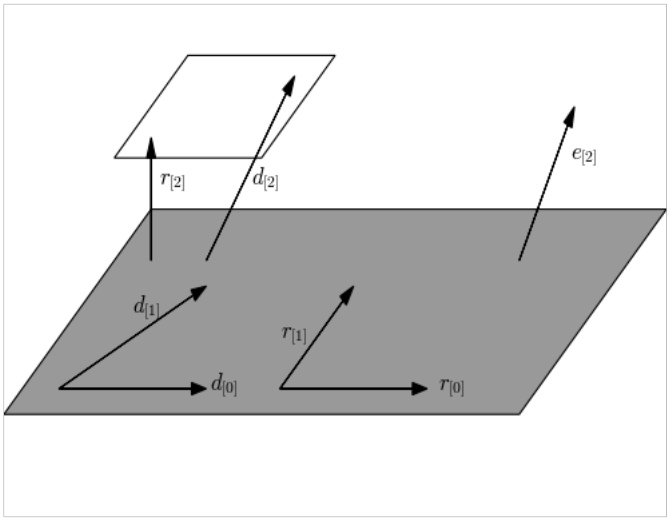


Figure 6: In the method of Conjugate Gradients, each new residual is orthogonal to all the previous residuals and search directions; and each new search direction is constructed (from the residual) to be A-orthogonal to all the previous residuals and search directions. The endpoints of $r_{[2]}$ and $d_{[2]}$ lie on a plane parellel to $\mathcal{D}_2$ (the shaded subspace). In this method, $d_{[2]}$ is a linear combination of $r_{[2]}$ and $d_{[1]}$¡/center¿

Implications of this choice:

- the search vectors are built from the residuals, the subspace span$r_{[0]}, r_{[1]}, \ldots, r_{[i-1]}$ is equal to $\mathcal{D}_i$. As each residual is orthogonal to the previous search directions, it is also orthogonal to the previous residuals
- $$r_{[i]}^T r_{[j]} = 0, \quad i \neq j \tag{3}$$
- each new subspace $\mathcal{D}_{i+1}$ is formed from the union of the previous subspace $\mathcal{D}_i$ and the subspace $A\mathcal{D}_i$. Hence,

$$\begin{aligned} \mathcal{D}_i &= \text{span}\{d_{[0]}, Ad_{[0]}, A^2 d_{[0]}, \ldots, A^{i-1} d_{[0]}\} \\ &= \text{span}\{r_{[0]}, Ar_{[0]}, A^2 r_{[0]}, \ldots, A^{i-1} r_{[0]}\}. \end{aligned}$$

This subspace is called Krylov subspace, a subspace created by repeatedly applying a matrix to a vector.
Gram-Schmidt conjugation becomes easy, because $r_{[i+1]}$ is already A-orthogonal to all of the previous search directions except $d_{[i]}$
Recalling the Gram-Schmidt constants : $\beta_{ij} = -r_{[i]}^T A d_{[j]} / d_{[j]}^T A d_{[j]}$; let us simplify this expression. Taking the inner product of $r_{[i]}$ and Equation 43,

$$
\begin{aligned}
r_{[i]}^T r_{[j+1]} &= r_{[i]}^T r_{[j]} - \alpha_{[j]} r_{[i]}^T A d_{[j]} \\
\alpha_{[j]} r_{[i]}^T A d_{[j]} &= r_{[i]}^T r_{[j]} - r_{[i]}^T r_{[j+1]} \\
r_{[i]}^T A d_{[j]} &= \begin{cases} \frac{1}{\alpha_{[i]}} r_{[i]}^T r_{[i]}, & i = j, \\ -\frac{1}{\alpha_{[i-1]}} r_{[i]}^T r_{[i]}, & i = j+1, \\ \text{(by Equation 44)} & \\ 0, & \text{otherwise.} \end{cases} \\
\therefore \beta_{ij} &= \begin{cases} \frac{1}{\alpha_{[i-1]}} \frac{r_{[i]}^T r_{[i]}}{d_{[i-1]}^T A d_{[i-1]}}, & i = j+1 \\ 0, & i > j+1. \end{cases} \qquad \text{(by Equation 37)}
\end{aligned}
$$

We notice that it is no longer necessary to store old search vectors to ensure the A-orthogonality of new search vectors. Therefore the benefits of CG:

- both the space complexity and time complexity per iteration are reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(m)$; m is the number of non zero entries in the Matrix A.

Therefore: $\beta_{[i]} = \beta_{i,i-1}$. Simplifying further:

$$
\begin{aligned}
\beta_{[i]} &= \frac{r_{[i]}^T r_{[i]}}{d_{[i-1]}^T r_{[i-1]}} \qquad \text{(by Equation 32)} \\
&= \frac{r_{[i]}^T r_{[i]}}{r_{[i-1]}^T r_{[i-1]}}
\end{aligned}
$$

According to all these expressions derived, we obtain a summary of the Conjugate gradient mehthod as follows:

$$
\begin{aligned}
d_{[0]} &= r_{[0]} = b - A x_{[0]} && (45) \\
\alpha_{[i]} &= \frac{r_{[i]}^T r_{[i]}}{d_{[i]}^T A d_{[i]}} \qquad \text{(by Equations 32 and 42),} && (46) \\
x_{[i+1]} &= x_{[i]} + \alpha_{[i]} d_{[i]}, \\
r_{[i+1]} &= r_{[i]} - \alpha_{[i]} A d_{[i]}, && (46) \\
\beta_{[i+1]} &= \frac{r_{[i+1]}^T r_{[i+1]}}{r_{[i]}^T r_{[i]}}, && (48) \\
d_{[i+1]} &= r_{[i+1]} + \beta_{[i+1]} d_{[i]}. && (49)
\end{aligned}
$$

- The maximum number of iterations required to achieve this bound using Steepest Descent is

$$
i \leq \left\lceil \frac{1}{2} \kappa \ln \left(\frac{1}{\epsilon} \right) \right\rceil,
$$

- The maximum number of iterations CG requires is

$$
i \leq \left\lceil \frac{1}{2} \sqrt{\kappa} \ln \left(\frac{2}{\epsilon} \right) \right\rceil.
$$

- Steepest Descent has a time complexity of $\mathcal{O}(m\kappa)$

- CG has a time complexity of $\mathcal{O}(m\sqrt{\kappa})$
- Both algorithms have a space complexity of $\mathcal{O}(m)$.

Starting:

Starting point is arbitrary. When an approximation is lacking, we genrally start from the origin.

Stoping:

When Steepest Descent or CG reaches the minimum point, the residual becomes zero