KRYLOV SUBSPACE ITERATION

Presented by: Nab Raj Roshyara
Master and Ph.D. Student
Supervisors:
Prof. Dr. Peter Benner,
Department of Mathematics, TU Chemnitz
and
Dipl.-Geophys. Thomas Günther

1. Februar 2005

1 Introductory part

1.1 What is an iterative method and why is it useful?

- All the numerical methods can be divided into two broad categories.

  1. **Direct methods:**
     A direct method is one that produces the result in a prescribed, finite number of steps.
     e.g. All versions of Gaussian elimination for solving

     $$Ax = b \tag{1}$$

     where A is $n \times n$ and non-singular, are direct methods.
  2. **Iterative methods(Indirect methods):**
     Iterative methods are in contrast to direct methods. An *iterative method* is an attempt to solve a problem, for example an equation or system of equations, by finding successive approximations to the solution starting from an initial guess.

- The answer of the question *"why is iterative method useful?"* is contained in the following questions.

1. How do you solve a system of linear equations $Ax = b$ when your coefficient matrix A is large and sparse (i.e., contains many zero entries)?
2. What if the order n of the matrix is so large that you cannot afford to spend about n^3 operations to solve the system by Gaussian elimination?
3. What if you do not have direct access to the matrix?

- Iterative methods are useful for problems involving large number of variables (could be of the order of millions) and to find the solution of the large and sparse linear systems where direct methods would be very expensive even with the best available computing power except where A has a special structure. So, researchers have long tried to approximate the solution of x iteratively.

■

1.2 Example of Large and sparse linear systems

- Linear systems have a central position in scientific computing. They arise frequently in scientific and engineering calculations. Simulation of continuous events and the discretization of a PDE by means of finite element method and finite difference method usually lead to large and sparse systems of equations. Some examples are given below.
  - Chemical process simulations
  - Device and circuit simulations
  - magnetic- field computation,
  - Earth and environmental sciences
  - Fusion energy.
  - Structural analysis
  - Structural biology

- There are lots of other examples which lead to very large linear systems of equations $Ax = b$ such as
  - weather prediction
  - chemical processes

– semiconductor-device simulation
– nuclear-reactor safety problems
– mechanical- structure stress
– and so on.

■

1.3 Basic idea of iterative method

- To find the solution x of the linear system (1), all iterative methods require an initial guess $x_0 \in R^n$ that approximate the true solution. So, we start with a good guess x_0 which can be taken by solving a much easier and similar problem.

- Once we have x_0, we use it to generate a new guess x_1 which is used to guess x_2 and so on. Moreover, we attempt to improve this guess by reducing the error with a convenient and cheap approximation.

- Let K be $n \times n$ be a simpler matrix which is similar to matrix A of the equation $Ax = b$. Then
in step $i + 1$:
we get a relation as follows:

$$Kx_{i+1} = Kx_i + b - Ax_i \tag{2}$$

From above equation (2), we solve the new approximation x_{i+1} for the solution x of $Ax = b$.

- **Remarks**

1. The arbitrary initial start x_0 converges if the modulus of largest eigen value of the matrix $(I - K^{-1})A$ is less than 1.
2. It is difficult to explain exactly about the choice of the similar matrix K of (2) but normally one can take

$$K = diag(A)$$

For example, to solve the discretized Poisson equation, the choice $K = diag(A)$ leads to a convergence rate $1 - 0(h^2)$ where h is the distance between grid points.

■

1.4 Classification of Iterative Methods

- There are two main types of iterative methods.
  1. Stationary methods
  2. Non-stationary methods
- Stationary methods are older, simpler to understand but usually not effective. On the other hand, non stationary methods are relatively recent development, their analysis is harder to understand and to implement but they can be highly effective.
- Many iterative methods have been developed. The most important iterative methods are listed below.

Stationary methods

- Jacobi method
- Gauss-Seidel method
- Successive over relaxion method(This is called SOR method.)
- Symmetric SOR method (This is called SSOR method.)

Non stationary methods

- Congugate Gradient method(CG method)
- General minimal Residual method(GMRES method)
- Minimal Residual (MINRES method)
- Symmetric LQ method(SYMMLQ method)
- Biconjugate Gradient method (BiCG method)
- Biconjugate Gradient Stabilized (Bi-CGSTAB method)
- Conjugate Gradient Squared (CGS method)
- Quasi-Minimal Residual (QMR method)
- Conjugate Gradients on the Normal Equations (CGNE and CGNR methods)
- Chebyshev Iteration

- **Remark:**
  All of the above listed non-stationary methods except Chebyshev Iteration are Krylov Subspace methods.

■

1.5 Preconditioners

- **Recall:**

  1. If $Ax = b$ is a linear system, then the condition number of the matrix A is denoted by $\kappa(A)$ and defined as

  $$\kappa(A) = \| A \| \| A^{-1} \|$$

  2. **Lemma:** Let A be non singular and x and $\hat{x} = x + \delta x$ be the solution of $Ax = b$ and $A\hat{x} = b + \delta b$, then

  $$\frac{\| \delta x \|}{\| x \|} \leq \kappa(A) \frac{\| \delta b \|}{\| b \|} \tag{3}$$

  3. From the above equation, we can say that if $\kappa(A)$ is not too large then, A is well conditioned. On the other hand, if $\kappa(A)$ is too large, a small value of $\frac{\|\delta b\|}{\|b\|}$ does not guarantee that $\frac{\|\delta x\|}{\|x\|}$ will be small. So if $\kappa(A)$ is large, we say that A is ill conditioned.

- The convergence rate of iterative methods depends on condition number of the coefficient matrix. Hence, we can try to attempt to transform the linear system into that one which is simpler, and in better condition but equivalent to the original system in the sense that it has the same solution.
  Let the system $Ax = b$ transformed into $\tilde{A}\tilde{x} = \tilde{b}$ for which $\tilde{A}$ is better conditioned than A. This procedure is called *"preconditioning"*. There are many ways to make such transformations.
  For example;
  Let M be any simple matrix that approximates A.
  Multiply $Ax = b$ by M^{-1} on the left to get

  $$\tilde{A}\tilde{x} = \tilde{b}$$

  Where
  $M^{-1}Ax = M^{-1}b$

$\tilde{A} = M^{-1}A$

$\tilde{b} = M^{-1}b$

$\tilde{x} = x$

■

2 Historical Background

- Since the early 1800s, researchers have been attracted towards iteration methods to approximate the solutions of large linear systems.
- The first iterative method for system of linear equations is due to *Karl Friedrich Gauss*. His least square method led him to a system of equations which is called " *Block wise Gauss-seidel method"*. The others who played an important role in the development of iterative methods, are:
  - Carl Gustav Jacobi
  - David Young
  - Richard Varga
  - Cornelius Lanczos
  - Walter Arnoldi
  - and so on
- From the beginning, scientists and researchers used to have both positive and negative feelings about the iterative methods because they are better in comparison of direct methods but the so-called *stationary iterative methods* like Jacobi, Gauss Seidel and SOR methods in general converge too slow to the solution of practical use. In the mid 1950's, the observation in *Ewald Bodewig's textbook* was that iteration methods were not useful, except when A approaches a diagonal matrix.
- Despite the negative feelings, researchers continued to design the faster iterative methods.
- Around the early 1950's the idea of *Krylov subspace iteration* was established by *Cornelius Lanczos* and *Walter Arnoldi*. Lanczos' method was

based on two mutually orthogonal vector sequences and his motivation came from eigenvalue problems. In that context, the most prominent feature of the method is that it reduces the original matrix to tridiagonal form. Lanczos later applied his method to solve linear systems, in particular the symmetric ones.

- In 1952, *Magnus Hestenes* and *Eduard Stiefel* presented a paper [1] which is about the classical description of the conjugate gradient method for solving linear systems. Although error-reduction properties are proved and experiments showing premature convergence are reported in this paper, the conjugate gradient method is presented here as a direct method, rather than an iterative method.

- This *Hestenes/Stiefel method* is closely related to a reduction of the Lanczos method to symmetric matrices, reducing the two mutually orthogonal sequences to one orthogonal sequence, but there is an important algorithmic difference. Whereas Lanczos used three-term recurrences, the method by Hestenes and Stiefel uses coupled two-term recurrences. By combining the two two-term recurrences (eliminating the "search directions") the Lanczos method is obtained.

- The Conjugate gradient method did not receive much recognition in its first 20 years because of the lack of exactness. Later researchers realized that this method is more fruitful to consider as truly iterative method. *John Reid (in 1972)* was the first one who pointed out in this direction. Moreover, it is now an important and cheaper iterative method for the symmetric case.

- Similarly, for the nonsymmetric case, *GMRES method* was proposed in 1986 by *Youcef Saad* and *Martin Schultz*. This method is based on *Arnoldi algorithm* and it is comparatively expensive. In this method, one has to store a full orthogonal basis for the Krylov subspace for nonsymmetric matrix A, which means the more iterations, the more basis vectors one must store. For many practical problems, GMRES can take hundreds of iterations, which makes a full GMRES unfeasable. This led to a search for cheaper near-optimal methods.

- *Vance Faber* and *Thomas Manteuffel's* famous result showed that generally there is no possibility of constructing optimal solutions in the Krylov subspace for nonsymmetric matrix A by short recurrences, as in the conjugate gradients method,

- The generalization of conjugate gradients for nonsymmetric systems, i.e. Bi-CG, often displays an irregular convergence behavior, including a possible breakdown. *Roland Freund* and *Noel Nachtigal* gave an elegant remedy for both phenomena in their QMR method. BiCG and QMR have the disadvantage that they require an operation with A^T per iteration step. This additional operation does not lead to a further residual reduction.

- In the mid 1980s, *Peter Sonneveld* recognized that one can use the A^T operation for a further residual reduction through a minor modification to the Bi-CG scheme, almost without additional computational costs which is called CGS method. This method was often faster but significantly more irregular.

- In 1992, *Henk A. van der Vorst*, showed that Bi-CG could be made faster and smoother, at almost no additional cost, with minimal residual steps.i.e. Bi-CGSTAB algorithm.

- Another class of acceleration methods that has been developed since around 1980, are the *multigrid* or *multilevel methods*. These methods apply to grid-oriented problems, and the idea is to work with coarse and fine grids.

■

2.1 Current Research Areas

- Although, multigrid methods are faster and efficient, there is no clear separation between the multigrid methods and Krylov methods because one can use the multigrid as a preconditioner for Krylov methods for less regular problems and the Krylov techniques are as smoothers for multigrid. So this is one of the fruitful direction for further exploration.

- In the past Researchers have been trying to find a new and better Krylov methods which converge in a less number of iterations. The convergence of a Krylov method depends on whether you are able to define a nearby matrix K that will serve as a preconditioner or not. Recent research is more oriented in that direction(i.e. to find a better preconditioner) than in trying to improve the Krylov subspace methods.

■

3 Krylov Subspace Iteration

- *Krylov Subspace Iterations* or *Krylov Subspace Methods* (or one can say only *Krylov Methods*) are a subset of non-stationary iterative methods. They are used as

  1. linear system solvers
  2. iterative solvers of eigenvalue problems.

- **Recall:**

  1. **Definition:** For the given two vectors $x = [x_1, x_2, x_3, ..., x_n]^T$ and $y = [y_1, y_2, y_3, ..., y_n]$ in R^n, the *inner product* of the two vectors x and y is denoted by $\langle x, y \rangle$ and defined as

$$\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i$$

  2. Although the inner product is a real number, it can be expressed as a matrix product.

$$\langle x, y \rangle = x^T y = y^T x$$

  3. **Definition:** Any matrix A is *symmetric* if

$$A = A^T$$

     and *positive definite* if

$$x^T A x > 0 \text{ for all } x \neq 0$$

  4. Any matrix is called *symmetric positive definite (SPD)* if it is symmetric and positive definite.

- Most important Krylov methods for solving linear system and for solving eigenvalue problems are listed below.

Type of Matrix	Krylov methods for $Ax = b$	Krylov methods for $Ax = \lambda x$
$A = SPD$	CG CGNE CGNR	Lanczos
$A \neq SPD$	GMRES BiCG Bi-CGSTAB MINRES SYMMLQ CGS QMR	Bi-Lanczos Arnoldi

- Here we will concentrate on those methods which are used as linear system solvers but first let us define a *Krylov subspace* and recall some of its elementary properties.

■

3.1 Definition and Elementary properties

- ***Definition:*** Let $A \in R^{n \times n}$ and $v \in R^n$ with $v \neq 0$ then,

  - The sequence of the form

    $$v, Av, A^2v, A^3v, ...,$$

    is called Krylov sequence.
  - The Matrix of the form

    $$K_k(A, v) = [v, Av, A^2v, A^3v, ..., A^{k-1}v]$$

    is called the k^{th} *Krylov Matrix* associated with A and v.
  - The corresponding subspace

    $$\mathscr{K}_\kappa(A, v) = span\{v, Av, A^2v, A^3v, ..., A^{k-1}v\}$$

    is called the k^{th} *Krylov subspace* or *Krylov Space* associated with A and v.

- ***Lemma*** Let $A \in R^{n\times n}$ and $v \in R^n$ with $v \neq 0$ then,

  1. $\mathscr{K}_\kappa(A, v) \subseteq \mathscr{K}_{\kappa+1}(A, v)$ and $A\mathscr{K}_\kappa(A, v) \subseteq \mathscr{K}_{\kappa+1}(A, v)$

  2. If $\sigma \neq 0$, $\mathscr{K}_\kappa(A, v) = \mathscr{K}_\kappa(\sigma A, v) = \mathscr{K}_\kappa(A, \sigma v)$

     i.e. Krylov Space remains unchanged when either A or v is scaled.

  3. For any c $\mathscr{K}_\kappa(A, v) = \mathscr{K}(A - cI, v)$

     i.e. Krylov Space is invariant under Shifting.

  4. $\mathscr{K}_\kappa(W^{-1}AW, W^{-1}v) = W^{-1}\mathscr{K}_\kappa(A, v)$

     i.e. Krylov Space behaves in a predictable way under similarity transformation.

  ■

- ***Theorem*** The space $\mathscr{K}_\kappa(A, v)$ can be written in the form

  $$\mathscr{K}_\kappa(A, v) = \{p(A)(v) : p \text{ is a polynomial of degree at most } k - 1\}$$

  ■

- ***Theorem*** The Krylov sequence $v, Av, A^2v, A^3v, ...$ terminates at l if l is the smallest integer s.t.

  $$\mathscr{K}_{l+i}(A, v) = \mathscr{K}_l(A, v)$$

  and

  $$dim[\mathscr{K}_{l+i}(A, v)] = dim[\mathscr{K}_l(A, v)]$$

  ■

3.2 Basic Idea of Krylov solvers

- In Krylov subspace methods, we keep all computed approximants $\{x_i\}$ and combine them for a better solution. Define the *Krylov Subspace* as

  $$\mathscr{K}_n(A, r_0) = span\{r_0, Ar_0, A^2r_0, a^3r_0, ..., A^{n-1}r_0\}$$

  where

  $$r_0 = b - Ax_0.$$

- If the vectors $r_0, Ar_0, ..., A^{n-1}r_0$, spanning the Krylov subspace $\mathscr{K}_n(A, r_0)$, are linearly independent, we note that

$$dim\mathscr{K}_n(A, r_0) = n$$

As soon as $A^n r_0 \in \mathscr{K}_n(A, r_0)$, we obtain that

$$\mathscr{K}_{n+i}(A, r_0) = \mathscr{K}_n(A, r_0) \text{ for all } i \in N.$$

and we get a linear combination

$$0 = c_0 r_0 + c_1 A r_0 + ... + c_{n-1} A^{n-1} r_0 + c_n A^n r_0$$

where at least $c_n \neq 0$ and $c_0 \neq 0$, otherwise the multiplication by A^{-1} would contradict the assumption that the vectors $r_0, Ar_0, ..., A^{n-1}r_0$ are linearly independent.
Indeed as $c_0 \neq 0$ we get;

$$A^{-1}r_0 = \frac{-1}{c_0} \sum_{i=1}^{n} c_i A^{i-1} r_0 \in \mathscr{K}_n(A, r_0) \tag{4}$$

The smallest index n with

$$n = dim[\mathscr{K}_n(A, r_0)] = dim[\mathscr{K}_{n+1}(A, r_0)$$

is called *the grade of A with respect to* r_0 and denoted by $\bar{r}(A, r_0)$.

■

- **Remarks:**

  1. The above argument also shows that there is no smaller Krylov subspace which contains $A^{-1}r_0$; therefore

$$\bar{r}(A, r_0) = min\{n : A^{-1}r_0 \in \mathscr{K}_n(A, r_0)\}$$

  2. The degree of the minimum polynomial of A is an upper bound for $\bar{r}(A, r_0)$.

- The basic idea of a *Krylov solver* is to construct a sequence of approximations getting closer to the exact solution x^*, such that

$$x_n \in x_0 + \mathscr{K}_n(A, r_0) \tag{5}$$

where x_0 is the initial approximation,

$$r_i = b - Ax_i \tag{6}$$

is the i^{th} residual and

$$e_i = x_i - x^* \tag{7}$$

denotes the i^{th} error. After $\bar{r}(A, r_0)$ iterations the exact solution is contained in the current affine Krylov subspace, i.e.

$$x^* \in x_0 + \mathscr{K}_{\bar{r}}(A, r_0) \tag{8}$$

as $x_0 + A^{-1}r_0 = x_0 + A^{-1}(Ax^* - Ax_0) = x^*$

■

3.3 Conjugate Gradient Method

- Conjugate gradient method is kind of Krylov Subspace method. As we explained in above table, this method is especially for symmetric and positive definite matrix A. The problem of solving $Ax = b$ by conjugate gradient method can also be known as a minimization problem.

3.3.1 Quadratic Form

- **Definition:** Define a function $\mathbf{J} : \mathbf{R^n} \to \mathbf{R}$ by

$$J(x) = \frac{1}{2}x^T Ax - x^T b + c \tag{9}$$

Where A is a matrix x and b are vectors and c is a scalar constant. Then the function $J(x)$ is said to be *quadratic form.*

- In addition, the *gradient* of a quadratic form $F(x)$ is defined as

$$\bigtriangledown F = F'(x) = [\frac{\partial}{\partial x_1}F, \frac{\partial}{\partial x_2}F, ...\frac{\partial}{\partial x_n}F]^T$$

$$\Longrightarrow \bigtriangledown F = F'(x) = \begin{bmatrix} \frac{\partial}{\partial x_1}F \\ \frac{\partial}{\partial x_2}F \\ . \\ . \\ . \\ \frac{\partial}{\partial x_n}F \end{bmatrix} \tag{10}$$

Using calculus from equations (9) and (10), one can get

$$\bigtriangledown F = \frac{1}{2}A^T x + \frac{1}{2}A^T x - b \tag{11}$$

If A is Symmetric, this equation can be reduced to

$$\bigtriangledown F = Ax - b \tag{12}$$

- Here is a very simple example of linear system to show the geometric meaning of quadratic form and its gradient which will help us to understand conjugate method well and easily.
  The system of linear quations
  $3x_1 + 2x_2 = 2$
  $2x_1 + 6x_2 = -8$
  is given by $Ax = b$ where

$$\mathbf{A} = \begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \text{ and } \quad b = \begin{bmatrix} 2 \\ -8 \end{bmatrix}$$

- The system $Ax = b$ is illustrated in fig.1. In general x^* lies at the intersection point of n hyperplanes each having dimension $n - 1$. For the above mentioned problem, the solution is

$$\mathbf{x}^* = \begin{bmatrix} 2 & -2 \end{bmatrix}^T$$

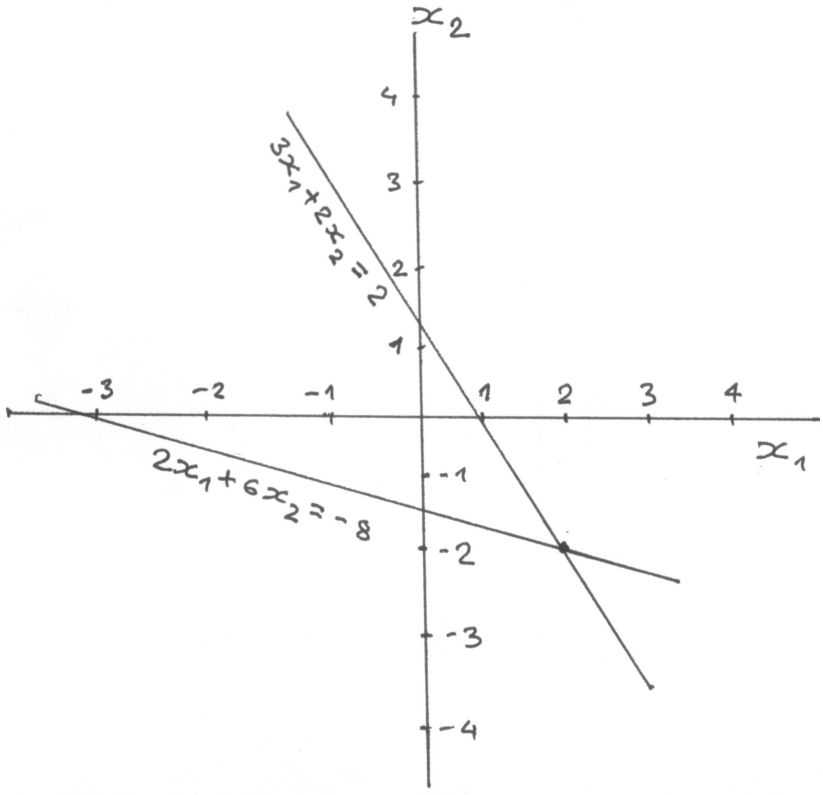

Abbildung 1: Sample of two dimensional linear system

- The visualization of the quadratic form of above system of linear equation where the constant c of the quadratic form is zero, is in the fig.2.

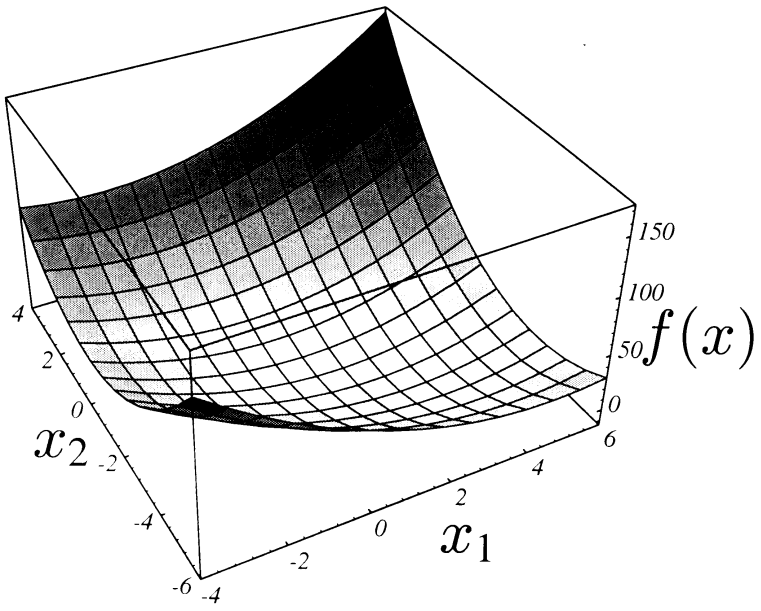

Abbildung 2: Graph of a quadratic form $F(x)$. The minimum point of this surface is the solution to $Ax = b$.

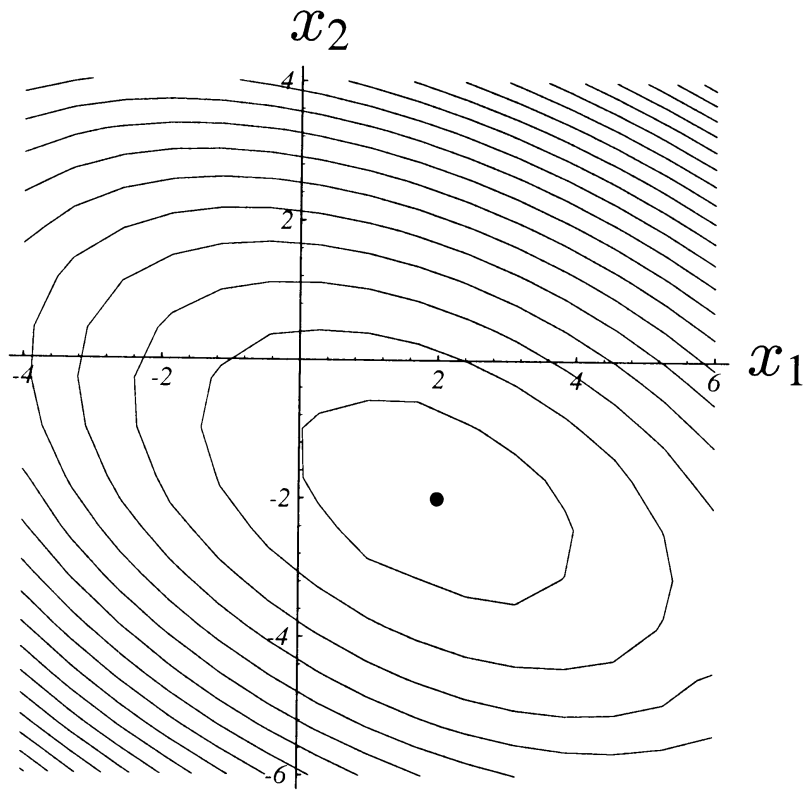

Abbildung 3: Contours of the quadratic form.

- The gradient $\bigtriangledown F$ for a given point x is a vector field which shows the direction of greatest increase of $F(x)$. Here is the figure of the gradient of the quadratic form.

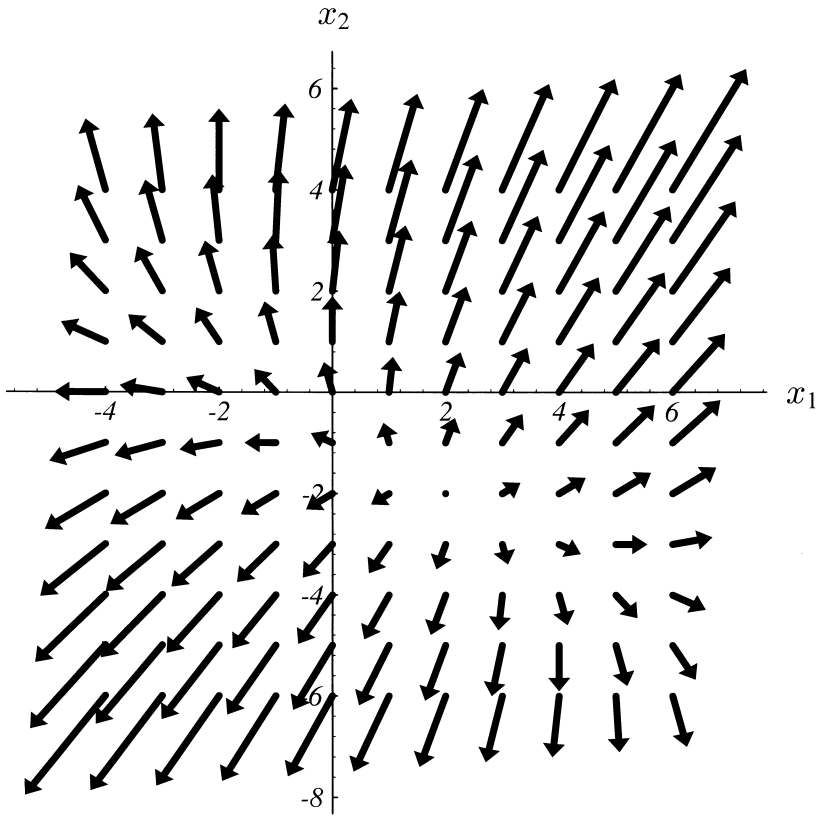

Abbildung 4: Gradient $\bigtriangledown F$ of the quadratic form. For every x, the gradient points in the direction of steepest increase of $F(x)$, and is orthogonal to the contour lines.

■

3.3.2 Minimization of Quadratic Form

- The gradient of $F(x)$ at the bottom of the paraboloid bowl(fig (2)) is zero. So one can minimize $F(x)$ by setting

$$\bigtriangledown F = F'(x) = 0$$

Setting $\bigtriangledown F = 0$, on equation (11) we get

$$Ax - b = 0 \Rightarrow Ax = b$$

which is our linear system to be solved. Thus, if A is positive definite and symmetric, then the solution of $Ax = b$ is the minimization of $F(x)$.

- **Claim:** Let $A \in R^{n \times n}$ be positive definite, $b \in R^n$ and a function F is defined as in (3), then there exists exactly one $x^* \in R^n$ for which
$F(x^*) = min\{F(x)\}$
and this x^* is the solution of $Ax = b$ also.

Proof:
The function is quadratic in $x_1, x_2, x_3, ...x_n,$. Let x^* denote the solution of $Ax = b$. Then
$F(x) = \frac{1}{2}x^T Ax - x^T Ax^*$

$= \frac{1}{2}x^T Ax - x^T Ax^* + \frac{1}{2}x^{*T} Ax^* - \frac{1}{2}x^{*T} Ax^*$

$= \underbrace{\frac{1}{2}(x - x^*)^T A(x - x^*)}_{\text{on which } F(x) \text{ depends.}} - \underbrace{\frac{1}{2}x^{*T} Ax^*}_{\text{independent of } x}$

So we can say that $F(x)$ is minimized only when $\frac{1}{2}(x - x^*)^T A(x - x^*)$ is minimized.
In this expression, A is positive definite. So we can say that the term $\frac{1}{2}x^{*T} Ax^*$ is positive unless $x - x^* = 0$.
$\Longrightarrow F(x)$ is minimum if and only if $y = x$.

■

- To minimize $F(x)$, we take a series of steps in the direction in which $F(x)$ decreases most quickly.
i.e. which is the direction opposite to $\bigtriangledown F$.

i.e. in the direction $- \bigtriangledown F = -(Ax - b) = b - Ax$

i.e. in the direction of the residual vector r where $r = b - Ax$.

- In our given example, Let's say, we start to guess from a point $x_{(0)} = [-2, -2]^T$.
- Here is the figure showing the location of our solution point and guess point.

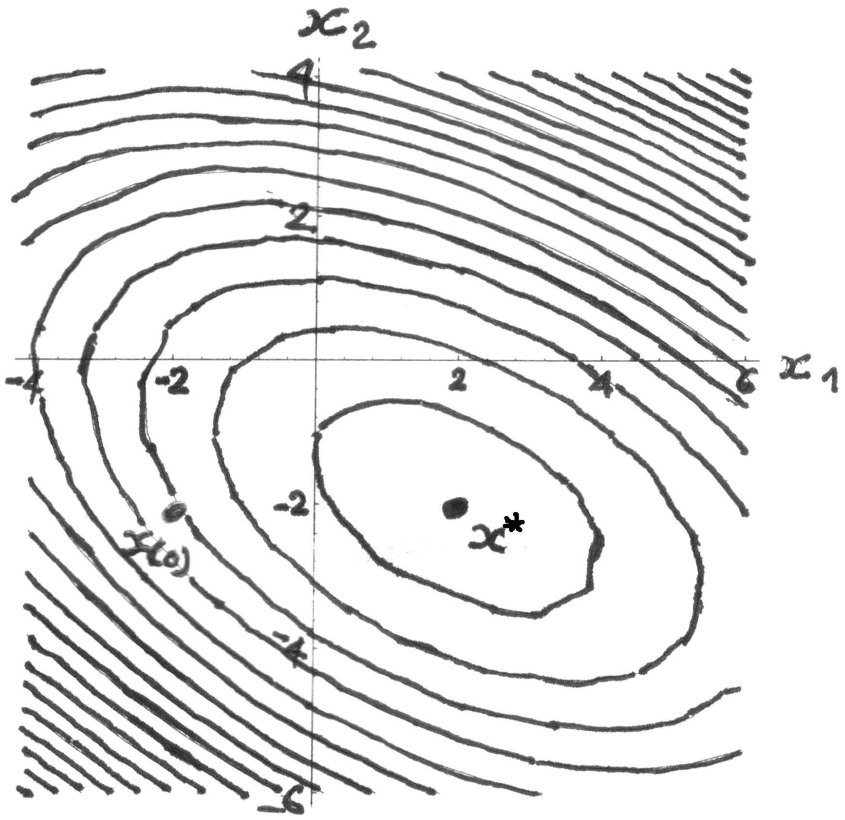

Abbildung 5: Contours of the quadratic form.

- **Remarks**

  1. The error $e_i = x_{(i)} - x$ is a vector that indicates how far we are from the solution.
  2. The residual $r_{(i)=b-Ax_{(i)}}$ indicates how far we are from the correct value of b.
  3. The relation between $r_{(i)}$ and $e_{(i)}$ is given by $r_{(i)} = -Ae_{(i)} = -\bigtriangledown F$.

■

3.3.3 Conjugate Gradient Algorithm

- **Definition:** Two vectors v_i and v_j are said to be *A-orthogonal or conjugate* to each other with respect to symmetric positive matrix A if

$$v_i^T A v_j = 0 \text{ for all } i \neq j$$
$$\Rightarrow (v_i, v_j) = 0 \forall i \neq j$$

- Let us consider a set of $A-$*congugate* search directions(vectors) $\{d_{(0)}, d_{(1)}, d_{(2)}, ..., d_{(n)}\}$ and the residual

$$r_0 = Ax_0 \tag{13}$$

and

$$d_0 = r_0 \tag{14}$$

Now, we start from a search line

$$x_{(1)} = x_{(0)} + \alpha d_{(0)} \tag{15}$$

- Here is a figure to show the Conjugate gradient method.

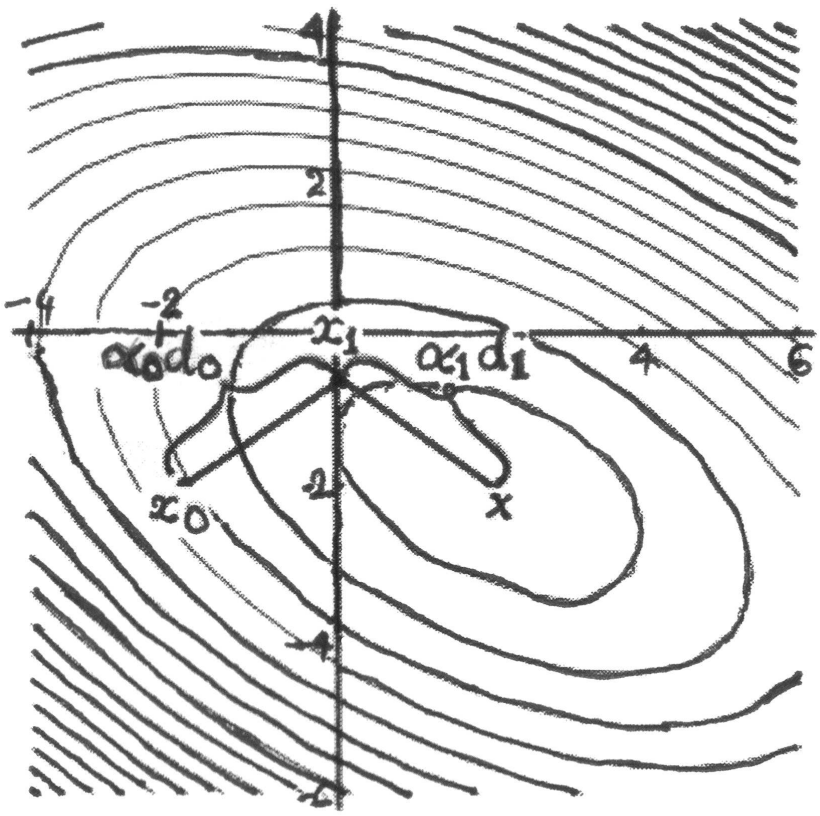

Abbildung 6: Geometrical View of Conjugate Gradient method

- In general, we take a point $x_{(i+1)}$ such that

$$x_{(i+1)} = x_{(i)} + \alpha_i d_{(i)} \tag{16}$$

By pre-multiplying both sides of (16) by $-A$ and adding both sides b we get

$$\underbrace{b - Ax_{(i+1)}}_{r_{(i+1)}} = \underbrace{b - Ax_i}_{r_{(i)}} - \alpha_{(i)} A d_{(i)}$$

$$\Rightarrow r_{(i+1)} = r_{(i)} - \alpha_{(i)} A d_{(i)} \tag{17}$$

But $r_{(i+1)}$ and $r_{(i)}$ are orthogonal to each other. So, we have

$$\begin{aligned}
&r_{(i+1)}^T, r_{(i)} = 0 \qquad [\because r_{(i)}^T r_{(j)} = o \text{ if } i \neq j] \\
&\Rightarrow (b - Ax_{(i+1)})^T, r_{(i)} = 0 \\
&\Rightarrow (b - A(x_{(i)} + \alpha_{(i)} d_{(i)}))^T, r_{(i)} = 0 [\because \text{ from equation (16)}] \\
&\Rightarrow (b - Ax_{(i)} - \alpha_{(i)} A d_{(i)})^T, r_{(i)} = 0 \\
&\Rightarrow (b - Ax_{(i)})^T r_{(i)} - \alpha_{(i)} (Ad_{(i)})^T r_{(i)} = 0 \\
&\Rightarrow \alpha_{(i)} = \frac{(b - Ax_{(i)})^T r_{(i)}}{(Ad_{(i)})^T r_{(i)}} \\
&\Rightarrow \alpha_{(i)} = \frac{r_{(i)}^T r_{(i)}}{(Ad_{(i)})^T r_{(i)}} \\
&\quad \Rightarrow \alpha_{(i)} = \frac{r_{(i)}^T r_{(i)}}{r_{(i)}^T (Ad_{(i)})}
\end{aligned} \tag{18}$$

Also, it is known that the next search direction d_{i+1} is a linear combination of r_{i+1} and d_i. So

$$d_{(i+1)} = r_{(i+1)} + \beta_{(i)} d_{(i)} \tag{19}$$

and

$$d_{(i)} = r_{(i)} + \beta_{(i-1)} d_{(i-1)} \tag{20}$$

Now,

$(Ad_{(i)})^T r_{(i)}$

$= (Ad_{(i)})^T (d_{(i)} - \beta_{(i-1)} d_{(i-1)})$

$$= (Ad_{(i)})^T d_{(i)} - \underbrace{\beta_{(i-1)}(Ad_{(i)})^T d_{(i-1)}}_{=0}$$

$$= (Ad_{(i)})^T d_{(i)}$$

$$\therefore (Ad_{(i)})^T r_{(i)} = (Ad_{(i)})^T d_{(i)} \tag{21}$$

From the equation (18) and (21), we get

$$\alpha_{(i)} = \frac{(r_{(i)})^T r_{(i)}}{(Ad_{(i)})^T, d_{(i)}} \tag{22}$$

In addition, $d_{(i+1)}$ is orthogonal to $Ad_{(i)}$. So, we get

$(d_{(i+1)})^T (Ad_{(i)}) = 0$

$\Rightarrow (r_{(i+1)} + \beta_{(i)} d(i))^T (Ad_{(i)}) = 0 \quad [\because$ From equation (20)]

$\Rightarrow (r_{(i+1)})^T (Ad_{(i)}) + \beta_{(i)} (d(i))^T (Ad_{(i)}) = 0$

$\Rightarrow (r_{(i+1)})^T (Ad_{(i)}) = -\beta_{(i)} (d(i))^T (Ad_{(i)})$

$$\Rightarrow \beta_{(i)} = \frac{-(r_{(i+1)})^T (Ad_{(i)})}{(d(i))^T (Ad_{(i)})}$$

$$\therefore \beta_{(i)} = \frac{-(r_{(i+1)})^T (Ad_{(i)})}{(d_{(i)})^T (Ad_{(i)})} \tag{23}$$

Note That from recurrence equation (17),

$$Ad_{(i)} = \frac{-1}{\alpha_{(i)}} (r_{(i+1)} - r_{(i)}) \tag{24}$$

So, from equation (23) and (24), we get

$$\beta_{(i)} = \frac{1}{\alpha} \frac{(r_{(i+1)})^T (r_{(i+1)} - r_{(i)})}{(d(i))^T (Ad_{(i)})}$$

Put the value of α on this relation,

$$\beta_{(i)} = \frac{1}{\frac{(r_{(i)})^T (r_{(i)})}{(Ad_{(i)})^T, d_{(i)}}} \frac{(r_{(i+1)})^T (r_{(i+1)}) - (r_{(i+1)})^T (r_{(i)})}{(d(i))^T (Ad_{(i)})}$$

$$\Rightarrow \beta_{(i)} = \frac{(r_{(i+1)})^T (r_{(i+1)})}{(r_{(i)})^T (r_{(i)})} \tag{25}$$

Putting all these above relations together, we get the following Algorithm.

- **Algorithm: Conjugate Gradient**

  1. compute $r_0 := b - Ax_0$, $d_0 := r_0$
  2. For $i = 01, 2, ...,$ until convergence, Do;
  3. $\alpha_{(i)} := \frac{(r_{(i)})^T r_{(i)}}{(Ad_{(i)})^T, d_{(i)}}$
  4. $x_{i+1} := x_i + \alpha_i d_i$
  5. $r_{(i+1)} = r_{(i)} - \alpha_{(i)} A d_{(i)}$
  6. $\beta_{(i)} = \frac{(r_{(i+1)})^T (r_{(i+1)})}{(r_{(i)})^T (r_{(i)})}$
  7. $d_{(i+1)} = r_{(i+1)} + \beta_{(i)} d_{(i)}$
  8. EndDo

■

3.4 Generalized Minimum Residual Method(GMRES)

3.4.1 Arnoldi Method

- Before the explanation of GMRES Method, Let me explain little bit about the Arnoldi method.

- The first step of Arnoldi method consists of the orthonormalization process. we start from a vector v as follows

$$v_1 = \frac{v}{\| v \|_2} \tag{26}$$

- For the k^{th} step,

$$\tilde{v}_{k+1} = Av_k - \sum_{j=1}^{k} v_k h_{jk} \tag{27}$$

where

$$h_{jk} = Av_k{}^T, v_j =< Av_k, v_j > \tag{28}$$

and

$$v_{k+1} = \frac{\tilde{v}_{k+1}}{h_{k+1,k}} \tag{29}$$

where

$$h_{k+1,k} =\| \tilde{v}_{k+1} \|_2 \tag{30}$$

- From the above two equations (27)(29), we get

$$Av_k = \sum_{j=1}^{k} v_j h_{jk} + v_{k+1} h_{k+1,k}$$

$$\Rightarrow Av_k = \sum_{j=1}^{k+1} v_j h_{jk} \tag{31}$$

- This process runs for $K = 1, 2, 3, ..., m$ to get

$$AV_m = V_{m+1} H_{m+1,m} \tag{32}$$

where

$$V_m = [v_1, v_2, ..., v_m] \in C_{n \times m} \tag{33}$$

and $H_{m+1,m}$ is a Hessenberg matrix given as follows.

$$= \begin{bmatrix} h_{11} & h_{11} & . & . & . & h_{1,m-1} & h_{1,m} \\ \\ h_{21} & h_{22} & . & . & . & h_{2,m-1} & h_{2,m} \\ 0 & h_{32} & . & . & . & h_{3,m-1} & h_{3,m} \\ \vdots & \ddots & & & & & \vdots \\ \vdots & & & & & & \vdots \\ 0 & & & & & h_{m,m-1} & h_{m,m} \\ 0 & 0 & & & & h_{m+1,m} & \end{bmatrix}$$

■

3.4.2 GMRES Method

- GMRES method is designed to solve nonsymmetric linear systems.This method is based on modified Gram-Schmidt procedure and was proposed in 1986 by *Yousef Saad and M.Schultz.*

- In the CG method,the residual form an orthogonal basis for the space

$$span\{r_{(0)}, Ar_{(0)}, A^2 r_{(0)}, ...\}$$

But in GMRES, the basis is formed explicitly by using arnoldi method and the iterates $x_{(m)}$ is minimized to

$$\| b - Ax_m \|_2$$

over the set

$$S_m = x_{(0)} + span\{r_{(0)}, Ar_{(0)}, A^2 r_{(0)}, ... A^{m-1} r_{(0)}\} = x_0 + \mathscr{K}_m(A, r_0) \tag{34}$$

- The key idea is to express $x_{(m)}$ in terms of Lanzos vectors $v_1, v_2, v_3, ...$ which span $\mathscr{K}_m(A, r_0)$ by assuming the first vector

$$v_1 := \frac{r_{(0)}}{\beta} \text{ where } \beta := \parallel r_{(0)} \parallel_2.$$

- The m^{th} step of Arnoldi method is given by

$$AV_m = V_{m+1}H_{m+1,m} \tag{35}$$

where V_m is a $n \times m$ matrix with column vectors $v_1, v_2, v_3, ...v_m$ and $H_{m+1,m}$ is a $m+1 \times m$ upper Hessenberg matrix.

- Similarly, in the m^{th} step of GMRES,

  1. The iterate $x_{(m)}$ can be written as

$$x_{(m)} = x_{(o)} + V_m y_{(m)} \text{where } y_{(m)} \in R^m \tag{36}$$

  and

  2. $\parallel b - Ax_{(m)} \parallel_2$ is minimized.

$\parallel b - Ax_{(m)} \parallel_2$

$= \parallel b - A(x_{(o)} + V_m y_{(m)}) \parallel_2$

$= \parallel b - Ax_{(o)} - AV_m y_{(m)} \parallel_2$

$= \parallel r_{(0)} - V_{m+1}H_{m+1,m}y_{(m)} \parallel_2$

$= \parallel \beta v_1 - V_{m+1}H_{m+1,m}y_{(m)} \parallel_2$

$= \parallel V_{m+1}(\beta e_1 - V_{m+1}H_{m+1,m}y_{(m)}) \parallel_2$

$= \parallel (\beta e_1 - V_{m+1}H_{m+1,m}y_{(m)}) \parallel_2$

[$\because$ the column-vectors of V_{m+1} are orthonormal]

- **Algorithm: GMRES**

1. compute $r_0 = b - Ax_0$, $\beta = \parallel r_0 \parallel_2$ and $V_1 := \frac{r_{(0)}}{\beta}$
2. Define the $(m+1) \times m$ matrix $H_{m+1,m} = \{h_{ij}\}_{1 \leq i \leq m+1, 1 \leq j \leq m}\}$. Set $H_{m+1,m} = 0$.
3. For $j = 01, 2, ..., m$ Do:
4. Compute $w_j := Av_j$
5. For $i = 1, 2, ..., m$ Do;
6. $h_{ij} := w_j^T v_i$
7. $w_j := w_j - h_{ij} v_i$
8. EndDO
9. $h_{j+1,j} := \parallel w_j \parallel_2$.IF $h_{j+1,j} = 0$, Set $m := j$ and go to 12.
10. $v_{j+1} = \frac{w_j}{h_{j+1,j}}$
11. EndDO
12. Compute $\parallel (\beta e_1 - V_{m+1} H_{m+1,m} y_{(m)}) \parallel_2$ which is the minimizer and $x_m = x_0 + V_m y_m$

- **References**

1. M. HESTENES AND E. STIEFEL, Methods of conjugate gradients for solving linear systems, J. Res. Nat. Bur. Stand., 49 (1952), pp. 409-436.
2. W. ARNOLDI, The principle of minimized iterations in the solution of the matrix eigenvalue problem, Quart. Appl. Math., 9 (1951), pp. 17-29
3. Jonathan Richard Shewchuk, An Introduction to the Conjugate Gradient Method Without the Agonizing Pain,
   under the web
   http://www.cs.ucdavis.edu/ bai/ECS231/schewchukcgpaper.pdf
4. Yousef Saad, Iterative methods for Sparse Linearsystems
5. Henk A. van der Vorst, The article on "KRYLOV SUBSPACE ITERATION" of the journal Computing in Science and Engineering; 2000.