

Dynamic Data Assimilation
Prof. S Lakshmivarahan
School of Computer Science
Indian Institute of Technology, Madras

Lecture - 18
Minimization algorithms

In this module, we are going to be talking about direct minimization algorithm. You may recall, there are essentially 2 ways of solving numerically the least square invert least square problems that arise within the context of linear and non-linear inverse problems; one is to be able to use matrix based algorithms. We saw at least 3 different families of matrix based algorithm; one is Cholesky; based a Cholesky factorization based algorithm derived from l u decomposition the second one is q r decomposition that comes out of Gramm Schmidt Orthogonalization process.


The third one is singular value decomposition that that is derived from the Eigen decomposition of the Grammians of the matrix h these matrix based techniques are called Dirac methods, we also alluded to iterate 2 methods for solving linear system, even though we did not cover them in this lecture, we are going to be talking about using minimization algorithms to be able to find the best least square estimate.

You may recall we have already reviewed the basic principles of minimization constraint unconstraint multivariate optimization problem in one of the early modules on mathematical preliminaries; we are going to derive a lot of the ideas in this module from those modules that are dealing with that that deals with constrained and unconstrained minimization algorithms.

(Refer Slide Time: 02:11)

MINIMIZATION PROBLEM – 1D

- $f: \mathbb{R} \rightarrow \mathbb{R}$, be a Convex function
- Example: $f(x) = ax^2 + bx + c$ with $a > 0$
- Rewrite: $f(x) = a\left[x + \frac{b}{2a}\right]^2 - \left(\frac{b^2 - 4ac}{4a}\right)$
- Minimizer $x^* = -\frac{b}{2a}$
- $f(x^*) = -\left(\frac{b^2 - 4ac}{4a}\right)$
- $f(x)$ is a parabola intersects the x -axis at
$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \text{ only if } \underline{b^2 > 4ac}$$
- Otherwise, $f(x)$ is above the x -axis



With that is the background D, I would like to be able to state the minimization problem in one dimension to that end let f be a function let f be a convex function in one dimension \mathbb{R} to \mathbb{R} ; that means, it is a scalar valid function of a scalar an example of such an f is $Ax^2 + bx + c$ with $a > 0$. We can rewrite this f of x as by perfecting the square as follows $a \left(x + \frac{b}{2a}\right)^2 - \frac{b^2 - 4ac}{4a}$.

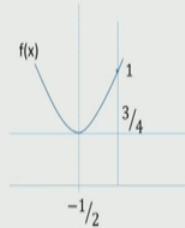
The second term does not depend on x the first term depends on x . So, by choosing x is equal to minus b by $2a$, we can; the first term, therefore, x^* is equal to minus b by $2a$ is the minimizer; the minimum value of the function is $f(x^*) = -\frac{b^2 - 4ac}{4a}$ geometrically $f(x)$ with a is equal to greater than 0 represents a parabola that intersects the x axis. So, you can think of this like this; that is the x axis, we are intersects like this if b^2 is greater than $4ac$ it intersects at 2 points; this is the minimum sorry the figure is not perfect, but you get the idea.

If $b^2 - 4ac > 0$; if b^2 is greater than $4ac$ the minimum is below the x axis otherwise the minimum is above the x axis and with $a > 0$ this quadratic function is a convex function its it is it is simply a parabola it has a unique minimum and the unique minimum or a minimizer is given by $x^* = -\frac{b}{2a}$.

(Refer Slide Time: 04:19)

MINIMIZATION PROBLEM – 1D

- $f(x) = x^2 + x + 1 = \left(x + \frac{1}{2}\right)^2 + \frac{3}{4}$
- $x^* = -\frac{1}{2}$ and $f(x^*) = \frac{3}{4}$
- Since $b^2 < 4ac$, $x_{1,2}$ are complex and $f(x)$ lies above the x -axis



I would like to be able to continue this example minimization in 1 D consider in f of x with x square plus x plus 1. This can be written as x plus 1 over half whole square plus 3 over 4; in this case x star is equal to minus half and f of x star is equal to 3 by 4.

Since b^2 is less than $4ac$ since b^2 is less than $4ac$ $x_{1,2}$ are complex and f of x lies above the x axis; that means, it has no real intersection with the x axis; it is a continuation of the discussion of the problem in the previous slide.

(Refer Slide Time: 05:08)

GENERALIZATION – n – DIMENSION

- $f: \mathbb{R}^n \rightarrow \mathbb{R}$ be Convex in \mathbb{R}^n
- Example: $f(x) = \frac{1}{2}x^T A x - b^T x + c$, A – SPD
- $\nabla f(x) = Ax - b = 0 \Rightarrow x^* = A^{-1}b$, minimizer of $f(x)$
- $f(x^*) = \frac{2c - b^T A^{-1}b}{2}$, minimum value of $f(x)$
- Instead of solving $Ax = b$, we seek to minimize $f(x)$ iteratively

Now I would like to be able to generalize it to n dimension, I would like to come; I would like to consider functions which are scalar valued function of a vector; they are convex in \mathbb{R}^n . A typical function that is convex in \mathbb{R}^n is this quadratic function one half of $x^T A x$ minus $b^T x$ plus c where A is a symmetric positive definite matrix.

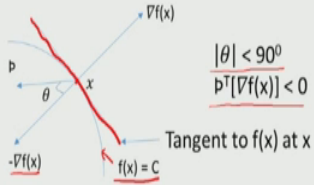
Using the principles of multivariate calculus, we can say that the gradient of this f ; f function is given by $Ax - b$. So, at the point where the gradient vanishes x^* is a minimizer of f of x the minimum value of f of x is given by this expression we can readily I would like all of you to be able to work this out in detail and understand the specific values how they are defined. So, one while one could solve the mini solve for the minimum by solving $Ax = b$ using any one of the matrix techniques A is SPD. So, I could apply Cholesky or qr or SPD kind of algorithm for solving $Ax = b$.

Instead we seek to minimize f of x iteratively by doing a gradient search and that is the theme of this module.

(Refer Slide Time: 06:54)

A DESCENT DIRECTION

- At any point $x \in \mathbb{R}^n$, $\nabla f(x)$ denotes the direction of maximum rate of increase



$|\theta| < 90^\circ$
 $p^T[\nabla f(x)] < 0$

- Since $p^T[\nabla f(x)] < 0$, p is called the descent direction
- $f(x)$ must decrease as we move a small distance along p away from x

So, what is the basis for the gradient technique; the gradient technique rests on the concept of what is called a descent direction descent direction; means the direction in which the function increases decreases increasing mean ascent decreasing mean descent.

So, let us consider a function f of x ; let us consider a contour where f of x is a constant; it takes a constant value that that contour is given by this curve.

So, let x be a point on the contour where f of x is equal to c that is this point that is that point; if you consider the gradient of f at that particular point the gradient refers to the direction of maximum rate of change. So, the negative of the gradient is the direction where the function decreases at the maximum rate. So, any direction p that makes an acute angle θ with the negative of the gradient is called a descent direction which.

So, the descent direction is described by this inequality $p^T \text{gradient of } f \text{ of } x$ must be less than 0. So, what does it mean the inner product of the of any direction p with the gradient if it is less than 0; what does that mean if you consider the perpendicular line like this which is the tangent f of x at the point x on the right side is the direction of increase the left side is the direction of decrease and p points to a direction of this tangent and the angle that p makes with the negative gradient is less than or equal to 90° .

Therefore θ ; the magnitude of this is less than or equal to 90° ; the inner product of p transpose with the gradient is less than less than 0 any p that satisfies; this is called the descent direction. So, what does what is the implication of this descent direction if you move along this descent direction p the function f is guaranteed to decrease f ; f of x must decrease as we move a small distance along p away from x .

So, this is the property of this is the property of descent direction why are we interested in descent direction; we are interested in minimization. Minimization means trying to find a point where the value of the function takes the least value. So, it makes sense to search for good descent directions and move along the descent directions to be able to find the minimum of a function and the picture essentially describes the role of the descent direction.

If you are trying to maximize you would you would consider action p in the right half that will be called ascent direction ascent direction in descent directions are duals of each other. So, without loss of generality in our discussion we will only consider minimization and descent direction.

(Refer Slide Time: 10:48)

STEEPEST DESCENT DIRECTION

- Let $\alpha > 0$ be a small real number
- Expand $f(x + \alpha p)$ in first order Taylor series
$$f(x + \alpha p) \approx f(x) + \alpha p^T [\nabla f(x)] < f(x) \text{ since } p \text{ is a descent direction}$$
- Setting $p = -\nabla f(x)$, the steepest descent direction:
$$f(x - \alpha \nabla f(x)) \approx f(x) - \alpha \|\nabla f(x)\|^2 < f(x)$$

and we get the maximum rate of decrease in $f(x)$ at x

Let α be a small parameter which are real number consider let p be a descent direction let p be a descent direction let α be a positive, but a small real number. So, x is a current point on the contour where f of x is equal to a constant value c as we saw in the previous picture.

So, x plus αp ; what does it tell you from x you move along p by a small distance that distance is controlled by α . So, f of x plus αp is a neighbor of x . So, f of x plus αp can be expanded in a Taylor series expand in a first order Taylor series. So, f of x plus αp is f of x plus α times p transpose gradient of f ; f of x if p is a descent direction from the previous slide, we know this quantity p tend to have x of x that is less than 0 α is positive. Therefore, I am subtracting something from the value f of x . Therefore, the whole value is less than f of x since p is the descent direction. So, what does this tell you?

Starting at x by identifying a descent direction if you move your small distance along the descent direction the function f of x ought to decrease that is the conclusion; that is the power of Taylor series expansion and p being a descent direction now our aim is to be able to maximize the decrease in other words we are we are always greedy in problem solving.

So, if I move away from α if I movie away from x by small distance controlled by α I am always looking for a direction p where the decrease in f will be the maximum

the direction p where the decrease will be maximum a little reflection will reveal is indeed p is equal to minus the gradient in other words if you move along the direction which is negative of the gradient the negative of the gradient the direction where f of x decreases of the maximum rate.

Such a direction p is called the deep steepest descent direction it is called the steepest descent direction because the rate of decrease along this direction is maximum. So, if I substitute p is equal to minus the gradient in the Taylor series expansion that we have about we get an expression which is like this where the quantity that being subtracted is alpha times the square of the norm of the gradient at the point x that essentially follows from that in that expression and which is definitely less than f of x therefore, by moving along a descent direction which is negative.

The gradient this direction is also called the steepest descent direction it guarantees the maximum rate of decrease of f of x at x . So, if I am interested in minimizing f of x . If I am moving away from a current operating point the best direction for one to move is always negative of the gradient at that particular point this is the basis for almost all of the known minimization algorithms, steepest descent that descent direction steepest descent direction these are the 2 key things.

(Refer Slide Time: 15:02)

ROLE OF RESIDUAL VECTOR

- x_k be the current operating point
- Residual $r_k = r(x_k) = -\nabla f(x)$
 $\quad \quad \quad = b - Ax_k$

$\xrightarrow{\quad}$
- r_k is the steepest descent direction of $f(x)$ at x_k
- Since $r_k = 0$ when $x_k = x^* = A^{-1}b$, $\|r_k\|$ is a measure of how far x_k is away from the minimum x^*
- $\|r_k\|$ could be used to test convergence of the iterative minimization

$z = Hx$
 $n(x) = z - Hx$

Now I am going to put it all together. So, let x_k be the current operating point what is the current operating point I am in my journey I started point x naught I would like to be

able to go and settle down on the minimum I am somewhere. So, at the k th iteration I am at a current operating point which is X_k . So, let X_k be the current operating point I am going to define a residual r_k is equal to r_k is a shortened form for r of X_k r of X_k is negative of the gradient at the point X_k .

Now, if you recall the gradient of f is $A X_k$ minus b . So, the negative the gradient is b minus $A X_k$ because a is a quadratic function given about this residual is not unrelated to the residual that we have talked about in the least square problems in the least square problems what is that we have we have z is equal to h of x we call r of x is equal to z minus h of x .

So, that is what we call the residual in this case b plays the role of z a plays the role of h . So, you can see the relation between b minus $A x$ and z minus h of x we called r of x as the residual within the context of least squares here we are calling r of X_k as a residual. This is the negative of the gradient of f at the point x .

This r_k must be a smaller r_k this r_k is the steepest descent direction of f of x at x_k . So, from the previous discussion we know negative of the gradient is the steepest descent direction. So, steepest descent direction; now have 2 interpretation one is the residual or it is a negative the gradient both are same by definition now how do I go how do I know that I have reached the minimum at the minimum the necessary condition is the gradient must vanish therefore, at the minimum r of k must be equal to 0. When X_k is equal to x^* $star$ is equal to a inverse b please remember a inverse b , I already know the value of a inverse b except that I is an expression for the minimum I do not know its actual numerical value, but I do I do know what is the mathematical expression that defines the inverse for this quadratic problem.

So, how do I test whether I have reached the minimum r naught one way to test if I have reached the minimum r naught is to compute a norm of the residual if the norm of the residual is very close to being 0; that means, I am very close to the minimum if the norm of the ah . So, the norm of r_k is a measure of how far my current operating point X_k is away from the minimum x^* . So, I have now a meter to speak. So, to speak to that measures how far I am away from the minimum please recall the I do not know where the minimum is if I had not; if I had known where the minimum is there is no problem there is no need to do any of these things.

So, without knowing the minimum I need to develop a test to understand how far I am away from the minimum one good measure one good way to measure the distance of the current point from the minimum is the norm of the residual which is also the norm of the negative the gradient. So, that is a very beautiful way to measure how far I am away from the minimum therefore, the norm of r_k could essentially be used as a convergence for the iterative minimization it is a test that measures the distance from where I am to where I want to be.

(Refer Slide Time: 19:28)

STEEPEST DESCENT FRAMEWORK

- Define the new operating points as

$$x_{k+1} = x_k + \alpha r_k$$
- Then: $f(x_{k+1}) < f(x_k)$ but
 $|f(x_{k+1}) - f(x_k)|$ depends on α
- α is called the step length parameter
- At x_k , the direction of search r_k is fixed
- Given x_k and r_k , how to choose α such that we get the maximum decrease in $f(x)$ as we move from x_k to $x_k + \alpha r_k$!
- New 1-D minimization problem: minimize $g: \mathbb{R} \rightarrow \mathbb{R}$ where

$$g(\alpha) = f(x_k + \alpha r_k) \quad \gamma: \mathbb{R} \rightarrow \mathbb{R}$$

So, with this I am now going to define the framework for this. So, called this steepest descent algorithm it the steepest descent framework if you wish. So, let x_k be the current operating point. Let x_k be the current operating point let r_k be the directions of steepest descent at x_k let α be the step length. So, $x_k + \alpha r_k$ it is going to give me my new operating point what is the proper to the new operating point f of x_k is less than f of x_{k+1} is less than f of x_k even though f of x_{k+1} is less than f of x_k the absolute value of the difference between f of x_{k+1} and f of x_k depends on the choice of α .

This is not α ; this is α sorry α is called the step length parameter α defines how far I go from x_k in the direction r_k at x_k . So, what is that we have now seen at time k x_k is fixed at time k the direction of search r_k is also fixed therefore, given x_k and r_k what is the question now how to choose α such that we get a maximum

decrease in f of x as we move away from X_k to $X_k + \alpha r_k$ I hope this question becomes very clear α is an arbitrary parameter.

We have not explained how to pick an α . So, now, the question here is that having decided to move along the direction negative the gradient the next question remains how far do I go in the chosen direction that question how to decide on α is formulated by this question is is the embedded within this statement given X_k and r_k how to choose α such that we get the maximum decrease in f of x as we move from X_k to $X_k + \alpha r_k$.

So, that is the important question that we would like to answer. So, look at this now we have we originally started minimizing f of x , we are at the current operating point X_k , we decided to move along the direction r_k ; r_k is a one dimensional vector. So, I would like to be able to minimize f of x as a function of α in that direction r_k . So, we get a new one dimensional minimization problem this minimization problem can be stated as follows define g of α which is equal to $X_k^T f$ of $X_k + \alpha r_k$ please understand f is a quadratic function is known X_k is known r_k is known. So, if you substitute $X_k + \alpha r_k$ is the quadratic form.

Since everything is known, but α I get a function of α ; α is the real parameter. So, g is a function from real to real therefore, we have reduced the problem to a 1 D minimization problem that the important recognition one needs to develop at this point.

(Refer Slide Time: 23:34)

A DIVIDE AND CONQUER PRINCIPLE

- Given n-dimensional minimization of $f(x)$ is reduced to a sequence of 1-dimensional minimization of $g(\alpha)$ at x_k along the steepest descent direction $r_k = -\nabla f(x_k)$, for $k = 0, 1, 2, \dots$
- This is the basis for the resulting iterative framework for the minimization of $f(x)$

HILL CLIMBING

$x_0, x_1, x_2, \dots, x_k, x_{k+1}, \dots$

So, I am now describing the fundamental principle of successive minimization. This is called divide and conquer principle given an n dimensional minimization of given the problem of n dimensional minimization of f of x , we reduce it to a sequence of one dimensional minimization of g of α at x of α at x of k along the steepest descent direction r_k which is the negative of the gradient.

So, what is the idea here I am here at the point I am here at the point x_k , let this be the direction of r_k , I would like to be able to go a distance α from here. So, let this point refer to α times r_k , I should not say like that; I am sorry, I will correct it. Now I would like to be able to go a distance of α times r_k that distance is α times r_k . So, this point now becomes x_{k+1} then from x_{k+1} I am going to consider r_{k+1} , I am going to again move α times r_{k+1} ; I am going to define x_{k+2} and so on. So, I move from x_k to x_{k+1} to x_{k+2} .

So, what is the basic idea given x_k given the direction r_k ; I would like to be able to minimize f of x along the direction r_k and find let x_{k+1} be the minimum point along this direction f of x becomes a function of α that is called g of α , then after having found x_{k+1} again compute the negative gradient you want to be able to go α times r_{k+1} that defines the point x_{k+2} . Therefore, I have start with x_0 I come to x_1 I come to x_2 I go to x_k I go to x_{k+1} the search continues.

So, in going from x_0 to x_1 , I essentially solve your one dimensional minimization problem the one dimensional minimization problem always happens in the direction of the negative gradient at each of these points. So, here in lies the basic principle of the divide and conquer a given n dimensional minimization problem is reduced to a sequence of one dimensional minimization problem g of α at X_k along the steepest different direction r_k for k is equal to $0, 1, 2, 3$ and that generates the sequence of points $x_0, x_1, x_2, \dots, x_k, x_{k+1}$ what is the idea here x_1 to the minimum then x_0 was x_{k+1} is closer to the minimum then x_k was.

So, there are 2 ideas in here there is a greedy principle involved in here what is the greedy principle. If I move, I would like to get closer to the minimum that is the greedy principle there is also a divide and conquer principle at work; here what is the divide and conquer principle I solve, we are given n dimensional minimization problem as a sequence of one dimensional minimization problem. So, I divide a larger problem into a smaller problem and solve a sequence of simpler problems to solve a complex problem s , that is the basic principle of divide and conquer. So, it is an amalgam of these 2 principle the greedy and the divide and conquer that is the basis for the iterative framework for the minimization of f of x .

I hope the basic ideas are very clear. I also would like to would like to remind the reader of the relation to hill climbing idea hill climbing sorry; hill climbing a mountaineer wants to go to the top of the Everest; what do they do? They move from base camp to base camp to base camp, they work for 8 hours, they are at a given base camp they know where the peak is they cannot go in a straight line from where they are to the peak if that were to be the case there will not be much interest in climbing Everest.

So, the path to the peak from where you are depends on the local properties of the mountain a travel mountaineer always looks at the local terrain and choose a direction such that if I worked for five hours I would like to be able to make sure my elevation increases. So, every time they move from one base camp to another base camp the level the height of the base camp is becomes higher and higher I get closer and closer to the hill humans now been utilizing this idea of hill climbing is based on the rate of change of the terrain at a given point for a long time. So, this steepest descent algorithm is patterned after what humans do when they climb hills to be able to go to the peak.

The only difference being instead of hill climbing I am trying to descent to the valley, but the analogy between hill climbing and descending to the valley must be extremely clear from the basic discussions from the basic discussions.

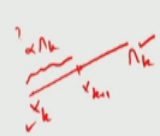
(Refer Slide Time: 30:00)

OPTIMAL STEP LENGTH – QUADRATIC PROBLEM

- Let $f(x) = \frac{1}{2}x^T A x - b^T x + c$, A – SPD
- Set $x_{k+1} = x_k + \alpha r_k$
- $g(\alpha) = f(x_{k+1}) = f(x_k + \alpha r_k)$
 $= f(x_k) + \frac{1}{2}(r_k^T A r_k)\alpha^2 + (r_k^T A x_k - r_k^T b)\alpha$
- $g(\alpha)$ is quadratic in α
- Setting: $\frac{dg}{d\alpha} = (r_k^T A r_k)\alpha + r_k^T (A x_k - b) = 0$
- Minimizer of $g(\alpha)$ is

$$\alpha_k = -\frac{r_k^T (A x_k - b)}{r_k^T A r_k} = \frac{r_k^T r_k}{r_k^T A r_k} > 0$$

Unless $r_k = 0$



So, what are the key things in any minimization the current operating point x_k this descent direction r_k and then the step length parameter α times r_k to decide x_{k+1} I know x_k I know r_k , I want to decide what is the best α I would like to be able to illustrate this idea of deciding the best α using an example of a quadratic function quadratic minimization.

Quadratic minimization problem is a model problem. So, I would like to illustrate the basic principles using the model problem $f(x)$ is equal to one half of $x^T A x$ minus $b^T x$ plus c A being SPD this is the problem we started with I am going to set x_{k+1} is equal to $x_k + \alpha r_k$, I still do not know; what is the best value of r_k . So, α is a free parameter I am going to define $g(\alpha)$ to be equal to $f(x_{k+1})$ that is equal to $f(x_k + \alpha r_k)$; if I substitute x is equal to $x_k + \alpha r_k$ in my $f(x)$ my $f(x)$ takes this particular form I would like all the readers to be able to do the substitution and simplification and do the simplification $f(x_k)$ is a number $r_k^T A x_k$ is a number $r_k^T b$ is also a number. So, given all these things $g(\alpha)$ is a simple quadratic function in α .

That means and that is to be expected because f of X_k is the slides of f of x along the direction r_k . So, g of α is a quadratic and α . So, to minimize g with respect to α please understand minimizing f of x along the direction r_k is the same as minimizing g with respect to α setting the derivative of g with respect to α equal to 0 we get this equation. So, the minimizer of g of α which is α_k is given by negative of $r_k^T A X_k$ minus b by $r_k^T A r_k$ please remember b minus $A X_k$ is r_k therefore, the whole expression becomes this. So, the optimal value of α that maximizes the decrease in the valley the function along the direction r_k is α_k and this α_k is always greater than 0 unless r_k is equal to 0 then α when r_k is equal to 0 we have already reached the minimum.

(Refer Slide Time: 33:11)

STEEPEST DESCENT/GRADIENT ALGORITHM

• $f(x) = \frac{1}{2}x^T A x - b^T x + c, x_0 \in \mathbb{R}^n$ given

$r_0 = r(x_0) = \cancel{A}x_0 - b$

For $k = 0, 1, 2, \dots$

Step 1 $\alpha_k = \frac{r_k^T r_k}{r_k^T A r_k}$ - optimal step length

Step 2 $x_{k+1} = x_k + \alpha_k r_k$ - iterates

Step 3 Test for convergence. If yes, exit

Step 4 $r_{k+1} = r_k - \alpha_k A r_k$ - residual update

$x_0, x_1, x_2, \dots, x_k, \dots$

11

So, so long as I am not in the minimum the step length is going to be positive. So, I have now all the information that I need to be able to build my algorithm. So, the steepest descent algorithm another name for it is also gradient algorithm. So, I am going to summarize my algorithm here let f of x be the function given by this let x naught be a starting point. At the starting point, I am going to compute my residual r_k which is the negative of the gradient which is the negative of the gradient. So, $A X_k$. So, $A x$ plus; I am sorry $A x$ naught minus b is the gradient I believe this is the actual gradient. So, it this has to be a negative sign in here.

Sorry negative sign in here therefore, at k is equal to 0, I know r_0 , I can compute α_0 $r_0^T r_0$ transpose r_0 divided by $r_0^T r_0$ transpose a transpose that is the optimal step length at that particular step X_{k+1} is equal to X_k plus $\alpha_k r_k$. So, that is the iterate. So, I move from k to $k+1$, I want to be able to test for convergence if the convergent test passes exit otherwise you update r_k that is called the residual update this is the gradient at the point X_{k+1} the gradient the negative of the gradient of the point X_{k+1} is related to negative the gradient at the point X_k and a correction term and a correction term.

So, from the definition of r_k one can very readily verify this residual update. So, these 4 steps together gives you the framework for the optimization algorithm. So, the optimization algorithm now generates a sequence of iterates x_1, x_2, \dots, X_k and. So, on and we have already made sure using the greedy principle X_k is closer to x^* than x_0 was that is the basic idea that is the basic idea so; that means, I am moving towards my goal that is the basic idea of the algorithm the algorithm is extremely simple very easy to implement this algorithm is called the gradient algorithm or the steepest descent algorithm.

(Refer Slide Time: 35:51)

ORTHOGONALITY OF RESIDUALS

- Recall that the residual at x_{k+1} is

$$\begin{aligned}
 r_{k+1} &= b - Ax_{k+1} \\
 &= b - A(x_k + \alpha_k r_k) \\
 &= r_k - \alpha_k A r_k \text{ - The residual update}
 \end{aligned}$$
- Also $r_k^T r_{k+1} = r_k^T (r_k - \alpha_k A r_k)$

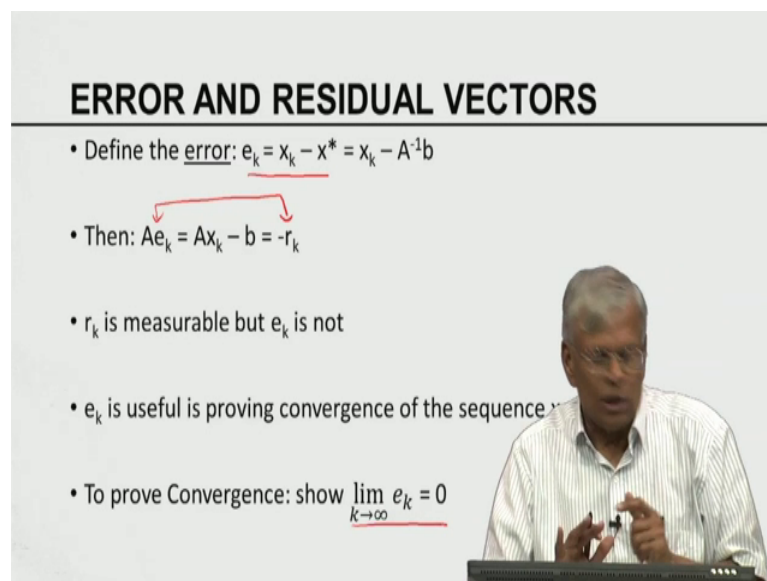
$$\begin{aligned}
 &= r_k^T r_k - \alpha_k r_k^T A r_k \\
 &= 0
 \end{aligned}$$
- That is, $r_{k+1} \perp r_k$
- Convergence question: When is $\lim_{k \rightarrow \infty} x_k = x^* = A^{-1}b$?

Now, I would like to talk about some of the properties of the residual the residual at X_k plus 1 sorry the residual at X_k plus 1 is given by r_{k+1} which is b minus $A X_{k+1}$, but X_{k+1} is given by this; therefore, r_{k+1} is given by this formula this is

called the residual update that you have seen in the previous step 4 of the algorithm. So, here is the verification of the correctness of the step 4. Now I would like to be able to take the inner product of x_k and r_{k+1} ; you can readily see from this calculation the inner product is 0. So, what does it mean 2 successive search directions are orthogonal 2 successive search directions are orthogonal please remember r_k is the direction of search at X_k .

So, with these properties; now I would like to be able to ask the fundamental question while I know X_k is moving towards x^* what happens to X_k as k becomes large. So, what is the limit of X_k we under what condition the limit of X_k will be equal to x^* where x^* is $A^{-1}b$. So, that is the question that is a convergence question when is the limit equal to the minimum; if I can show in the limit the X_k tends to x^* , then I have guaranteed convergence of that algorithms.

(Refer Slide Time: 37:30)



ERROR AND RESIDUAL VECTORS

- Define the error: $e_k = x_k - x^* = x_k - A^{-1}b$
- Then: $Ae_k = Ax_k - b = -r_k$
- r_k is measurable but e_k is not
- e_k is useful in proving convergence of the sequence
- To prove Convergence: show $\lim_{k \rightarrow \infty} e_k = 0$

So, let me summarize what we have done so far; we introduced the notion of descent direction we introduced the Norden of steepest descent direction we then said that given an operating point and a descent direction, I would like to be able to find the value the step length for which I get the maximum decrease. So, I move from point to point base point to base point to base point guaranteeing that along the chosen at the chosen point along the chosen direction I have a paint the maximum decrease possible I have implemented by greedy strategy the divide and conquer strategy essentially tells you I

search for the minimum along successive gradient direction and the previous analysis essentially tells you successive gradient directions successive are directions of search are mutually orthogonal.

So, with that now we have the burden to show that while it goes towards the minimum it will indeed hit the minimum as the number of iterates k grows unbounded. So, that is what is called convergence proof the in order to prove the convergence. Now I am going to introduce another term called the error term the error is $x_k - x^*$; I do not know the numerical value of x^* , but I only know its form $a^{-1}b$. So, if I multiply a by e_k I get $A(x_k - x^*) = Ax_k - b$ is equal to $-r_k$. So, what does it mean the error and the residual are related to each other by this relation $a e_k$ is equal to $-r_k$.

We have already known the minimum occurs when the residual the norm of the residual is 0 when the norm of the residual is 0 at the minimum the norm of the error must also be 0; therefore, I could utilize either r_k or e_k to analyze my convergence the only difference is r_k is measurable e_k is not measurable why e_k is not measurable because I do not x^* even though e_k is not measurable he said very; it is very useful to use e_k in my theoretical analysis. So, r_k is measurable, but not e_k ; e_k is useful in proving convergence of a sequence to prove convergence instead of showing x_k tends to x^* it is enough to show e_k tends to 0 as k tends to infinity that is an equivalent way of proving convergence.

(Refer Slide Time: 40:21)

ENERGY NORM OF THE ERROR e_k

- Define

$E(x_k) = f(x_k) - f(x^*)$
- Setting $b = Ax^*$ and simplifying

$E(x_k) = \frac{1}{2}(x_k - x^*)^T A (x_k - x^*)$
 $= \frac{1}{2} e_k^T A e_k = \frac{1}{2} \|e_k\|_A^2 > 0$
 unless $e_k = 0$
- $E(x_k)$ is a measure of how far x_k is from x^*
- Since A is SPD, $E(x_k) = 0$ if and only if $x_k = x^*$

A - SPD

$x^T A x > 0 \quad x \neq 0$
 $= 0 \quad x = 0$

In order to prove convergence those of you who are involved in application only they can simply take the algorithm program and apply it, but we would like to go a bit further we would like to be able to provide a complete analysis of this algorithm. So, in order to be able to prove convergence; I am going to define an energy function the energy function is $e f X k$ which is equal to f of $X k$ minus f of x star the value of the function.

At $X k$ minus the minimum value setting b is equal to $A x$ star and simplifying it can be shown a of $X k$ after simplification becomes one half of $e k$ transpose $a e k$; you may recall from our module on matrices this is called an energy norm. So, this is one half of the square of the energy norm of the error that is what E of $X k$ is. So, E of $X k$ is always greater than 0 unless $e k = 0$ why is that $e a$ is a possible definite quadratic form from the definition of quadratic form; we already know x transpose $A x$ is always greater than 0 for all x naught equal to 0 which is equal to 0 only when x is equal to 0 that is the definition half a being positive definite.

So, since a is positive definite E of $X k$ being the energy norm E of $X k$ is a good measure of how far I am away from the minimum if E of $X k$ is 0 I am at the minimum if it is not 0 I am not at the minimum. So, it is a kind of a meter that tells you how far away I am from the minimum. So, E of $X k$ is the measure of how far $X k$ is from x star again I want to reinforce E of $X k$ is 0 only when $X k$ is equal to x star.

(Refer Slide Time: 42:24)

A FRAME WORK FOR CONVERGENCE PROOF

- Evaluate $E(x)$ along the trajectory and prove that $E(x_k)$ is a decreasing function of k
- Since $E(x_k)$ is bounded below by zero, prove that $E(x_k) \rightarrow 0$ as $k \rightarrow \infty$
- This framework is due to A. Lyapunov and has come to be known as the Lyapunov method

So, what is the basic idea I want to be able to. So, what is the proof of framework proof of the convergence I want to evaluate the E of x_k along the trajectory; I want to be able to evaluate E of x_k along the trajectory and prove x is a decreasing function of k .

E of x_k is bounded below by 0. So, I have a positive function which is bounded below by 0. So, if I can prove E of x_k tends to 0 as k tends to infinity I would have proved convergence this framework of trying to use the energy function to prove convergence is an idea which is it is an old idea is due to a famous Russian mathematician called Lyapunov and has come to be called Lyapunov technique. So, that we are going to be talking about is the convergence proof that is directly related to the fundamental principles that Lyapunov introduced towards the turn of the last century about 1919; in the early 1900s.

(Refer Slide Time: 43:28)

A RECURSIVE RELATION FOR $E(x_k)$

- $E(x_{k+1}) = f(x_{k+1}) - f(x^*)$
- Substituting $x_{k+1} = x_k + \alpha_k r_k$ and simplifying with $b = A^{-1}x^*$, it follows:

$$E(x_{k+1}) = \beta_k E(x_k) \quad (*)$$

$$\beta_k = \left[1 - \frac{(r_k^T r_k)^2}{(r_k^T A r_k)(r_k^T A^{-1} r_k)} \right]$$

16

So, we are going to talk about the recursive relation for e of x_k . So, E of x_{k+1} because I am interested in trying to evaluate E of x_k along the trajectory I am going to first want to compute E of x_{k+1} and related to E of x_k substituting x_{k+1} is equal to x_k plus $\alpha_k r_k$ and simplifying and remembering that b is equal to a inverse x^* because x^* is equal to a inverse; it follows that. So, I should have said the following; I am sorry this there is an error here sorry; this should have been x^* is equal to a inverse b I am sorry; we will make the correction.

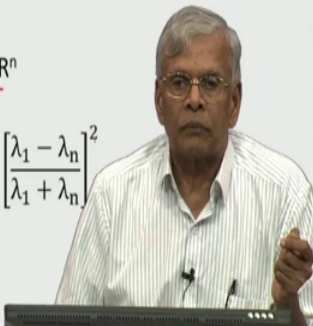
So, with the x^* is equal to $a^{-1}b$ it follows. In fact, I would like to recommend all of you to be able to undertake this simplification is a non trivial simplification it can be shown that $E(x^k)$ is equal to β^k times $E(x^0)$ where β is given by this quantity β^k is given by this quantity. So, I have now related A to $E(x^{k+1})$ and β^k is the multiplying constant.

So, what does this tell you if you can show β^k is less than one that essentially tells you $E(x^k)$ is less than $E(x^{k+1})$ is less than $E(x^k)$. So, the whole proof of convergence now rests in our showing that β^k is a constant which is less than one if it is less than one from this equation x^* we would approve essentially would have put a convergence. So, convergence now boils down to showing that β^k is less than one?

(Refer Slide Time: 45:43)

KANTOROVICH INEQUALITY

- Let $A \in \mathbb{R}^n$ be SPD
- Let $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n > 0$ be the eigenvalues of A
- Kantorovich inequality states: for any $y \in \mathbb{R}^n$

$$\frac{(y^T y)^2}{(y^T A y)(y^T A^{-1} y)} \geq 1 - \left[\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n} \right]^2$$


This quantity β^k less than one is analyzed through a very well known inequality called Kantorovich inequality. In fact, it is Kantorovich again another Russian mathematician who is the first one to prove the convergence of gradient algorithm. So, now, I am going to quote the Kantorovich inequality which is a very powerful inequality very useful inequality there are several different versions of this inequality in the literature I am talking about the simplest possible version of Kantorovich inequality it goes somewhat like this let A be a symmetric positive definite matrix let λ_1, λ_n be the n Eigen values since A is a SPD even λ_n the smallest value is greater than 0.

Kantorovich inequality essentially says that for any vector y in \mathbb{R}^n , $y^T A y$ divided by $y^T A^{-1} y$ is always greater than $1 - \frac{\lambda_1}{\lambda_n}$ or $1 - \frac{\lambda_1}{\lambda_n}$ divided by $1 + \frac{\lambda_1}{\lambda_n}$ whole square. The proof of this is very well known. You can get several different versions of the proof by looking at Wikipedia for example or many textbooks on optimization theory. I think it is an interesting exercise to prove this inequality now we are going to take this for granted. So, if I substitute y is equal to r_k . So, please remember this inequality is true for any y .

(Refer Slide Time: 47:37)

UPPER BOUND ON β_k : CONDITION NUMBER OF A

- Combining:
- $$\beta_k = \left[1 - \frac{(r_k^T r_k)^2}{(r_k^T A r_k)(r_k^T A^{-1} r_k)} \right]$$

$r_k = y$ $\mathcal{K}_2(A) = \frac{\lambda_1}{\lambda_n}$

$$\leq \left[\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n} \right]^2 = \left[\frac{\frac{\lambda_1}{\lambda_n} - 1}{\frac{\lambda_1}{\lambda_n} + 1} \right]^2 = \left[\frac{\mathcal{K}_2(A) - 1}{\mathcal{K}_2(A) + 1} \right]^2 = \beta < 1$$
- $\mathcal{K}_2(A) = \frac{\lambda_1}{\lambda_n}$ = condition number of A
 ≥ 1 when A is SPD

Now, what is the quantity I mean I am interested in r_k ; r_k is any vector I have no idea what r_k will be except that it is a negative the gradient at the point x_k therefore by combining Kantorovich inequality and β_k . So, β_k has a value from the previous slide we know this is the value of β_k by identifying r_k is equal to y in Kantorovich inequality we readily follow that β_k is less than this quantity that essentially comes from Kantorovich inequality.

I can divide the numerator and denominator by λ_n that gives you $\frac{\lambda_1}{\lambda_n} - 1$ by $\frac{\lambda_1}{\lambda_n} + 1$ the numerator is less than the denominator not only that please remember \mathcal{K}_2 is called the condition number is a spectral condition number of the matrix A please remember \mathcal{K}_2 of A is the ratio λ_1 by λ_n the range ratio of the largest to the smallest Eigenvalue;

therefore, β_k is less than or equal to $\frac{\kappa^2 - 1}{\kappa^2 + 1}$.

I am going to call this quantity as β and it is very trivial to verify β is less than 1. So, β_k is uniformly bounded by β which is less than one. So, for all k β_k is less than 1 if and also I want you to recognize that the condition number is always greater than or equal to 1 when SPD. So, you can readily see the notion of a condition number now critically use in the proof of convergence of the gradient method.

(Refer Slide Time: 49:19)

CONVERGENCE OF $E(x_k)$

- Hence

$$E(x_{k+1}) \leq \beta E(x_k) \text{ and } \beta < 1$$
- Iterating

$$E(x_k) \leq \beta^k E(x_0) \rightarrow 0 \text{ as } k \rightarrow \infty$$
- Hence, $\lim_{k \rightarrow \infty} E(x_k) = 0$ and $\lim_{k \rightarrow \infty} x_k = x^*$

Therefore combining all these x_k plus E of x_{k+1} is less than or equal to β times E of x_k . Now, please remember the I have replaced β_k by β ; β_k is less than β ; β is less than 1. Therefore, if I iterate this I get this expression E of x_k is less than or equal to β^k times E of x_0 .

β^k being less than one goes to 0 as k goes to infinity therefore, E of x_k as k goes to infinity goes to 0 if E of x_0 is finite that implies limit of x_k goes to x^* and hence convergence. So, by exploiting the greedy nature and by concentrating on the energy norm of the error instead of the residual we can always see the relation between the previous slides because we have related the energy norm of the error through r_k and I am by combining cleverly the Kantorovich inequality we are now able to prove convergence. So, what does this mean this means gradient method for quadratic functions indeed will converge starting from any point.

(Refer Slide Time: 50:42)

SUMMARY – MAIN THEOREM

- If $f(x) = \frac{1}{2}x^T Ax - b^T x + c$, and A is SPD then the gradient algorithm, starting from any x_0 , Converges to the minimum as $k \rightarrow \infty$
- However, the rate of Convergence depends on β which in turn depends only on the condition number $\kappa_2(A)$ of A and not on n , the dimension of the space

$$\beta = \frac{(\kappa_2(A) - 1)^2}{(\kappa_2(A) + 1)^2}$$

So, that is the main theorem or a main summary if f of x is equal to 1 half of x transpose $A x$ minus b transpose x plus c in A is SPD then the gradient. So, what is that we have achieved the gradient algorithm starting from any point x naught converges to the minimum as k goes to infinity that is a guaranteed that is what is called convergence theorem it is all the people who use gradient algorithm to do minimization of quadratic functions they rely on the power of this theorem. So, here comes the basic idea of combining cover ways to measure how far I am away from the minimum we are able to using greedy principles we are showing not only we are moving closer towards the minimum, but also indeed I will be on the top of the minimum as k equal to infinity.

So, once I approved convergent the next one is how fast I go towards convergence that relates to what is called the rate of convergence please understand the rate of convergence depends on beta; if beta is very small beta to the power k grows a goes to 0 very fast. For example, if this is k if this is beta to the power k if beta is 0.9 it goes to 0 slowly this is point beta is equal to 0.9.

Let us say, but if beta is equal to 0.1. So, beta smaller the value of beta faster the convergence larger; the value of beta slower the convergence, the convergence rate depends on the value that the beta takes now please remember converge the beta essentially depends on the condition number please go back what is the value of beta; beta is essentially we can we can see from this slide I will I will cut it now beta is

essentially beta is equal to kappa 2 of a minus 1 divided by kappa 2 of a plus 1 whole square.

So, if kappa 2 is very large let us say 10 thousand the numerator will be 9999 divided by 10000 and one that ratio is very close to 1. Therefore, beta will be very large and the other hand if beta is 10, the numerator is 9. The denominator is 11, beta is much smaller than the previous case. It comes down very fast. Therefore, ultimately the rate of convergence is controlled by only the condition number the rate of convergence depends on the condition number and that is the condition number of a is fixed because given a matrix a; the condition number of fixed those. So, the rate of convergence well its fixed turns on the condition number of a. So, that is an important conclusion that comes out of this.

(Refer Slide Time: 54:05)

ESTIMATION OF THE NUMBER OF ITERATIONS

- For what value of k:

$$\frac{E(x_k)}{E(x_0)} \leq \beta^k = \varepsilon = 10^{-d}$$

$E(x_k) \leq \beta^k E(x_0)$
- Solving $\beta^k = 10^{-d} \Rightarrow k^* = \left\lceil \frac{d}{\log_{10} \beta^{-1}} \right\rceil$

$\lceil x \rceil$ - CEILING OF x
 = SMALLEST INTEGER GREATER THAN x
- That is, for a given β , in k^* iterations

$\boxed{\frac{E(x_k)}{E(x_0)} \leq 10^{-d}}$ ✓

Now, I would like to be able to bring a practical question the method says as k goes to infinity I will converge at k goes to infinity E of x; x of k becomes 0, but if I am doing arithmetic in a final precision in a finite precision machine what is that I am looking for we already know E of X k; we already know E of X k is equal to beta to the power k is less than or equal to beta to the power k of E of x naught.

Therefore, E of X k divided by E of x naught is less than or equal to beta to the power k I am not in I do not want to wait until beta to the power k goes to 0, but I would like to be

able to make β to the power k less than ϵ where ϵ is 10 to the power of minus d .

So, what is d in a single precision arithmetic I cannot believe I cannot compute anything beyond 5-6 decimal accuracy in a double precision arithmetic I cannot compute anything more than 10 I am I am sorry about thirteen fourteen decimal accuracy. So, by picking d to be 6; I can think of a single precision arithmetic I can by picking d is equal to 13-14. I can think of double precision arithmetic. So, whatever be I let d be a coefficient I am going to use in my judgment as to when to exist exit out of the algorithm. So, let setting β to the power k equal to 10 to the power of d by taking logarithms on both sides for a given d my k^* is given by d over \log of 10 of one over β .

β is less than 1; $1/\beta$ is greater than one. So, logarithm of a number greater than one is positive d is positive and this is called this. This is called this ceiling of x ceiling of x and its value is equal to the smallest integer greater than x smallest integer greater than x . So, k^* is an integer which is the smallest integer yeah the smallest integer greater than that and for a given β k^* in case.

So, what does it mean for a given β in k^* iterations the ratio of the energy of the error at time k to the ratio the energy at time 0 will be less than or equal to 10 to the power of minus d . So, I this is the measure actual measure one would use in practical applications. So, what does it mean by picking k^* which depends on d ; I can pre-compute the number of iterations one would need to do to be able to get closer to the minimum.

(Refer Slide Time: 57:31)

DEPENDENCE OF k^* ON β AND $\mathcal{K}_2(A)$		
$\mathcal{K}_2(A)$	β	k^*
1	0	-
10	0.66942	40
100	0.960788	403
1000	0.996008	4030
10^4	0.9996	40288

Now, I am going to give you a feel for the dependence of k on β and κ . So, \mathcal{K}_2 is one the κ ; when the condition number is one for which matrix condition number is one identity matrix β is 0 that generally does not happen. So, we are not interested in cases where the function the matrix A symmetric positive definite matrix is quadratic form with identity matrix. So, so let us consider other cases when κ is 10 β is given by 0.6 is 942 k^* is forty likewise when κ is 10 to the power of 4 β is 9996 I would indeed need about forty thousand iterations.

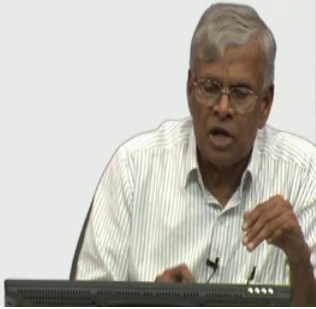
So, you can readily see the measure the power of the measure given by κ ; the power of the measure given by κ as κ increases the number of iterations needed also increases and this is for the value of d is equal to 6. This is for a value of d is equal to 6. So, this essentially tells you by please specifying the accuracy with which we want to decide the minimum and for a given κ we can estimate the number of iterations needed to be able to achieve. So, once I know I actually the desired accuracy once I know the number of iterations then that could be used to be able to decide what is the total amount of time that is needed to be able to go to the minimum use in the greedy in such algorithm.

So, that is a sense this table essentially provides the summary of the performance of the gradient algorithm in trying to minimize quadratic objective functions.

(Refer Slide Time: 59:20)

EXAMPLE IN \mathbb{R}^2

- Consider $A = \begin{bmatrix} 1 & 0 \\ 0 & \lambda \end{bmatrix}$, with $\lambda \geq 1$
- $f(x) = \frac{1}{2}x^T A x = \frac{1}{2}(x_1^2 + \lambda x_2^2)$ ←
- $= \frac{x_1^2}{(\sqrt{2})^2} + \frac{x_2^2}{(\sqrt{2\lambda})^2}$ ←
- The minimum of $f(x)$ occurs at $x^* = (0, 0)^T$
- $\nabla f(x_k) = Ax = (x_1, \lambda x_2)^T = -r(x)$
- Set $x_0 = (\lambda, 1)^T$
- Verify $\alpha_0 = \frac{r_0^T r_0}{r_0^T A r_0} = \frac{2}{1+\lambda}$
- $x_1 = x_0 + \alpha_0 r_0 = \frac{\lambda-1}{\lambda+1} \begin{bmatrix} \lambda \\ 1 \end{bmatrix}$ ←



I am now going to illustrate this by an example consider a matrix A which is $\begin{bmatrix} 1 & 0 \\ 0 & \lambda \end{bmatrix}$; λ is a parameter f of x is equal to one half of $x^T A x$ for this the function is given by this function is given by this expression this it can be rewritten as an equation for an ellipse the minimum value of f of the minimum value of f of x occurs at x^* which is $(0, 0)^T$; I can compute the gradient of this function the gradient is given by this I am going to start my x_0 is equal to $(\lambda, 1)^T$ as the as my initial point just for the fun of it I am going to start there; I could verify that α_0 is given by this which is given by this.

So, x_1 is given by this vector. So, I am actually computing the progress of the iteration from step 0 to step one using this specific example I have given all the values I would like to encourage you to be able to verify each of these values.


(Refer Slide Time: 60:33)

EXAMPLE IN R^2 - CONTINUED

- Continuing

$$x_k = \left(\frac{\lambda-1}{\lambda+1}\right)^k \begin{bmatrix} \lambda \\ (-1)^k \end{bmatrix} \rightarrow 0 \text{ as } k \rightarrow \infty$$

- When $\lambda = 4$, $x_k = (0.6)^k \begin{bmatrix} 4 \\ (-1)^k \end{bmatrix}$
- Zig-Zag behavior: Iterates x_0, x_1, x_2, \dots exhibit oscillatory behavior which slows the convergence



24

Continuing it can be shown that if I start at x naught which is equal to λ 1 my X_k will be given by this no please understand this vector is fixed the constant that multiplies the vector λ minus 1 divided by λ plus 1 to the power k that that number is less than one it is that number to the power k . So, that goes to 0 therefore, X_k goes to 0 as k goes to infinity therefore, they have already verified the convergence when λ is equal to 4 as a particular example I have already proved convergence for any λ for a specific λ is equal to 4 X_k is given by this.

So, 0.6 to the power k goes to 0 as k goes to infinity. So, X_k goes to infinity. Now what is the basic idea here this behavior has a zigzag behavior what does this mean look at this now the first component is the first component is 0.6 to the power k 4 the second component is minus 1 to the power k . So, when k is even the second component is positive when k is odd the second component is negative. So, what does this mean at even iterates are above the x axis audit rates are below the x axis. So, it goes zigzag like this with decreasing amplitude. So, if this is if this is x naught if this is the origin this is x star the iterates zigzags across the x axis.

So, the iterates exhibit an oscillatory behavior which essentially which is essentially responsible for the slow convergence you have seen in the previous table that when n when the $kappa$ is at the order of 10 to the power of 4. It takes 40000 iteration; why it takes 40000 iteration because the iterates has an oscillatory exhibit the iterates exhibit an

oscillatory behavior. So, instead of moving directly towards this it keeps oscillating and making progress towards that the progress towards the minimum is little slower that is an inherent behavior of the gradient algorithm the zigzag behavior the presence of oscillatory behavior is what is responsible for the slow nature the convergence of the gradient algorithm in cases that is exhibited by this specific problem.

(Refer Slide Time: 63:11)

1-D SEARCH – GENERAL CASE

- Given an operating point x , a descent direction p , the optimal step length α is obtained by minimizing

$$g(\alpha) = f(x + \alpha p)$$
- Solve

$$\frac{dg}{d\alpha} = [\nabla f(x + \alpha p)]^T p = 0 \quad \rightarrow (*)$$
- When f is quadratic $\Rightarrow g$ is quadratic and $(*)$ is linear in α
- When f is not quadratic, $(*)$ can be solved only numerically

25

Now, I would like to talk about I would like to talk about the 1 D search you remember we have to decide alpha k given an operating point x and a descent direction p the optimal step length is obtained by minimizing g of alpha I am going to go; I am going back to one of the problems we decided earlier. So, what is that we have sent we would like to be able to get the derivative of g with respect to alpha which is given by this expression I want to solve that when f when f is quadratic g is quadratic and star is linear in alpha when f is not quadratic g can be solved only numerically. So, what does it mean I am now thinking of extensions of the gradient algorithm to non quadratic functions?

So, this is what we have already proven, but in principle not all functions we are called upon to minimize our quadratic. So, when you are dealing with minimization of a non quadratic function odd function which are highly non-linear and apply the gradient method this method of choosing the step length parameter alpha is a little bit more involved because this can be solved only numerically because it is not linear I would like to I would like to caution I would like to bring that caution to the forefront right now.

(Refer Slide Time: 64:37)

QUADRATIC APPROXIMATION TO $g(\alpha)$

- Compute the following values of $g(\alpha)$:
 $g(0) = f(x)$
 $g(1) = f(x + p)$
 $\frac{dg(0)}{d\alpha} = [\nabla f(x)]^T p$
- Let $m(\alpha) = a\alpha^2 + b\alpha + c$ be a quadratic approximation to $g(\alpha)$

26

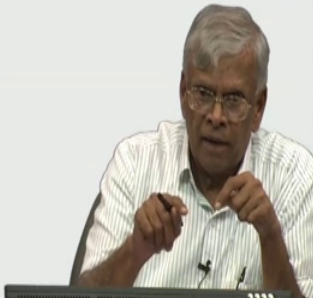
So, in that case what we do we essentially compute a quadratic approximation g of 0 is f of x g of 1 is f of x plus p the gradient of g with respect to α is given by that?

So, if I have 3 pieces of information about the function g of α . So, what is g of α g of α is the slice of f of x along the direction r_k starting at the point x_k . So, I am trying to fit a model which is a quadratic model that approximates g of α I do not know g of α precisely because of non-linear function I am going to bring a non-linear approximation to g of x g of x has 3 unknown parameters a b and c , I have given 3 different pieces of information about g of α by using these 3 of pieces of information

(Refer Slide Time: 65:31)

QUADRATIC APPROXIMATION TO $g(\alpha)$

- Set $m(0) = g(0) = c$ ✓
 $m(1) = g(1) = a + b + c$ ✓
 $m'(0) = \frac{dm(\alpha)}{d\alpha} \Big|_{\alpha=0} = \frac{dg(\alpha)}{d\alpha} \Big|_{\alpha=0} = (2a\alpha + b) \Big|_{\alpha=0} = b$ ✓
- Hence $a = \frac{g(1) - g(0) - g'(0)}{2}$
 $b = g'(0)$
 $c = g(0)$
- Setting $\frac{dm(\alpha)}{d\alpha} = 0 \Rightarrow$ optimal step length
 $\alpha = -\frac{b}{2a} = -\frac{g'(0)}{2[g(1) - g(0) - g'(0)]}$ ✓



I can essentially estimate the parameters a b and c. So, m of 0 is g of 0 which is c m of one is g of 1 which is a plus b plus c m prime the derivative of at 0 is given by 2 a alpha plus b.

(Refer Slide Time: 65:49)

A LOOK BACK

- Gradient method Converges only asymptotically even for Quadratic functions
- Is finite time Convergence feasible at least theoretically?
- The conjugate direction/conjugate gradient methods can in principle achieve this goal for Quadratic functional

28

Sorry $2a\alpha + b$ $2a\alpha + b$, there is b. Therefore, I know I now know what you see I now know what is a plus b plus c. I now know what is b. So, I can now compute a is given by this b is given by this c is given by this. These are all functions of g and g is the slice of f of x along the direction r k g is known. So, I can compute the

values a , b and c once I compute the value of a and b , b and c while I do not know the actual g of x I have a quadratic approximation to g of α .

I can minimize the model the quadratic model by setting the gradient of m of α with respect to α is equal to 0 and I get the optimal structuring parameter given by this expression. So, you would generally use this expression when you use gradient method for a general non-linear function, but you would use the ratio of the r_k transpose r_k divided by r_k transpose a_k for α_k when the function f is a quadratic function.

(Refer Slide Time: 67:15)

QUADRATIC APPROXIMATION TO $g(\alpha)$

- Compute the following values of $g(\alpha)$:

$g(0) = f(x)$
 $g(1) = f(x + p)$
 $\frac{dg(0)}{d\alpha} = [\nabla f(x)]^T p$
- Let $m(\alpha) = a\alpha^2 + b\alpha + c$ be a quadratic approximation to $g(\alpha)$

So, we have now talked about generalization of the application of the gradient method to problems where the function may not be a quadratic function. So, you are look back a summary of search gradient method converges only asymptotically even for quadratic functions. So, that is the important thing.

Gradient method converges for quadratic functions, but it converges asymptotically; asymptotically means what as iterations go to infinity in practice we may not have all the time that is needed to be able to wait for convergence. So, we would like to be able to cut off the convergence at a desired place. So, we would like to be able to see in when the magnitude of the residual becomes smaller than 10 to the power of minus d where d is 6 or 10 or fourteen depending on the kind of accuracy you want, which case the total number of iterations needed.

One can pre-compute according to the table based on which is which is fundamentally depend on the condition number of the matrix A and all those good results are only for quadratic functions and the key thing I want to emphasize is that the convergence is asymptotic. So, that behooves they ask a question is finite time convergence at all possible theoretically the answer is the well known conjugate gradient method and the conjugate direction idea can be used in principle to achieve this goal of finite time convergence at least for quadratic functionals.

So, what is the what is the summary here the gradient algorithms are very good for quadratic functions they have asymptotic convergence we can pre-compute the number of iterations needed to get the desired accuracy those are all the fundamental properties of gradient algorithm and that sets the limit of the power of the gradient algorithm once we have understood this asymptotic nature of convergence of gradient algorithm, we would like to ask ourselves a question is there a way for us to be able to start from an arbitrary place to be able to get to the minimum infinite time that is the ultimate desire in the design of any optimization minimization algorithm.

So, we would like to be able to explore at least theoretically if it is feasible talk you cannot do anything in practice if it is not theoretically possible. So, this exploration of finite time convergence at least in the theoretical sense is a question that arises from the analysis of the gradient algorithm the answer to this question does that exist an algorithm in principle that can converge to the minimum in finite time the answer is yes a class of method called conjugate direction conjugate gradient methods is one that theoretically provides this framework of finite convergence that the next topic which we will pursue in our next lecture.

Thank you.