Set 12: Unconstrained Optimization and Linear Systems

Kyle A. Gallivan

Department of Mathematics

Florida State University

Foundations of Computational Math 1

Fall 2012

Unconstrained Smooth Optimization

Problem 12.1. Given $f(x) : \mathbb{R}^n \to \mathbb{R}$, solve

$$\min_{x \in \mathbb{R}^n} f(x)$$

to find a local minimizer.

- global convergence desirable
- superlinear convergence rate very desirable
- robustness desirable
- efficiency desirable
- both $n = 1$ and $n > 1$ are of practical interest

Solutions

Definition 12.1. The following minimizers are of interest:

- The point $x^* \in \mathbb{R}^n$ is a global minimizer if $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$.
- The point $x^* \in \mathbb{R}^n$ is a local minimizer if $f(x^*) \leq f(x)$ for all $x \in \mathcal{N}_{x^*} \subset \mathbb{R}^n$ where $\mathcal{N}_{x^*}$ is a neighborhood of x^*.

Solving Linear System via Optimization

The very optimistic scenario is solving the system $Ax = b$ by doing the following:

- Create a function $f : \mathbb{R}^n \to \mathbb{R}^+$ which has a unique minimum value $f(x_{min})$ where $x_{min} = A^{-1}b$.
- Create a fixed point iteration with fixed point x_{min} with each step:
  - Pick a direction p_k, choose a scale $\alpha_k > 0$:

$$x_{k+1} = x_k + \alpha_k p_k$$

  - Find α_k that minimizes $f(x_k + \alpha p_k)$ via a simple analytical solution.

Symmetric Positive Definite Systems

If $A \in \mathbb{R}^{n \times n}$ is symmetric positive definite,

- A defines an inner product on $\mathbb{R}^n$
- Any inner product induces a vector norm:

$$\|v\|_A^2 = v^T A v$$

- We have:

  Solving $Ax = b$ is equivalent to minimizing a quadratic functional that defines a convex $f(x)$. That is,

$$f(\alpha x_1 + (1-\alpha)x_0) \leq \alpha f(x_1) + (1-\alpha) f(x_0), \quad 0 \leq \alpha \leq 1$$

A-norm Minimization

$$\hat{x} = A^{-1}b \text{ and } \phi(x) = \|x - \hat{x}\|_A^2$$

$$\phi(x) = (x - A^{-1}b)^T A(x - A^{-1}b)$$

$$= (x^T A - b^T A^{-T} A)(x - A^{-1}b)$$

$$= x^T Ax - x^T AA^{-1}b - b^T A^{-T} Ax + b^T A^{-T} AA^{-1}b$$

$$= x^T Ax - x^T b - b^T x + b^T A^{-1}b = x^T Ax - 2b^T x + b^T A^{-1}b$$

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

$$\operatorname*{argmin}_{x \in \mathbb{R}^n} \phi(x) = \operatorname*{argmin}_{x \in \mathbb{R}^n} Q(x)$$

Stepsize Selection

Theorem 12.1. *Suppose $x_k \in \mathbb{R}^n$ be the current guess at x and $p_k \in \mathbb{R}^n$ is the direction in which the next step is to be taken. The scale α_k that minimizes $Q(x_{k+1})$ where*

$$x_{k+1} = x_k + \alpha_k p_k \quad \textit{is} \quad \alpha_k = \frac{p_k^T r_k}{p_k^T A p_k}$$

Proof.

$$\begin{aligned} Q(\alpha) &= \frac{1}{2}(x_k + \alpha p_k)^T A(x_k + \alpha p_k) - b^T(x_k + \alpha p_k) \\ &= \frac{1}{2}\alpha^2 p_k^T A p_k - \alpha r_k^T p_k - b^T x_k + \frac{1}{2} x_k^T A x_k \\ Q'(\alpha) &= \alpha p_k^T A p_k - r_k^T p_k \end{aligned}$$

□

Steepest Descent

Lemma 12.2. *If $Q(x) : \mathbb{R}^n \to \mathbb{R}$ then the direction of most rapid descent on the surface $Q(x)$ at some point x^* is the negative gradient*

$$-\nabla Q(x^*) = \begin{pmatrix} -\frac{\partial Q}{\partial \xi_1}(x^*) \\ \vdots \\ -\frac{\partial Q}{\partial \xi_n}(x^*) \end{pmatrix}$$

$\nabla Q(x^*)$ is the external normal of the tangent plane of the surface defined by $Q(x)$. We have

$$-\nabla Q(x) = b - Ax = r$$

which is the residual vector.

Example

Let $A = A^T$ be 2×2. Note $\alpha_{12} = \alpha_{21}$.

$$Q(x) = \frac{1}{2} \begin{pmatrix} \xi_1 & \xi_2 \end{pmatrix} \begin{pmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} - \begin{pmatrix} \beta_1 & \beta_2 \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix}$$

$$= \frac{1}{2} \left(\xi_1 \alpha_{11} \xi_1 + \xi_1 \alpha_{12} \xi_2 + \xi_2 \alpha_{21} \xi_1 + \xi_2 \alpha_{22} \xi_2 \right) - \left(\beta_1 \xi_1 + \beta_2 \xi_2 \right)$$

$$\frac{\partial Q}{\partial \xi_1} = \alpha_{11} \xi_1 + \frac{1}{2} \left(\alpha_{12} + \alpha_{21} \right) \xi_2 - \beta_1$$

$$\frac{\partial Q}{\partial \xi_2} = \alpha_{22} \xi_2 + \frac{1}{2} \left(\alpha_{12} + \alpha_{21} \right) \xi_1 - \beta_2$$

Steepest Descent

- example of a line search for unconstrained optimization
- negative gradient, i.e., steepest descent direction, always used
- stepsize α_k selection varies for different problems
- Direction selected is always locally fastest.
- There is often a better direction to consider based on more "global" information.

Steepest Descent for $Ax = b$

- For $Ax = b$ the cost function of $\|x - A^{-1}b\|_A^2$ is used.
- The negative gradient is the residual vector.
- Steepest descent solves a series of one dimensional optimization problems, i.e., constrained to a line in $\mathbb{R}^n$.
- Analytical solution available and used in algorithm.
- $r_{k+1} \perp r_k$ but no global relationship is maintained between them.
- SD can converge very slowly.
- Consider level curves and iterates for example with $n = 2$.

Steepest Descent

A is symmetric positive definite

x_0 arbitrary; $r_0 = b - Ax_0$; $v_0 = Ar_0$

do $k = 0, 1, \ldots$ until convergence

$$\alpha_k = \frac{r_k^T r_k}{r_k^T v_k}$$
$$x_{k+1} \leftarrow x_k + r_k \alpha_k$$
$$r_{k+1} \leftarrow r_k - v_k \alpha_k$$
$$v_{k+1} \leftarrow A r_{k+1}$$

end

Preconditioned Steepest Descent

A, M are symmetric positive definite

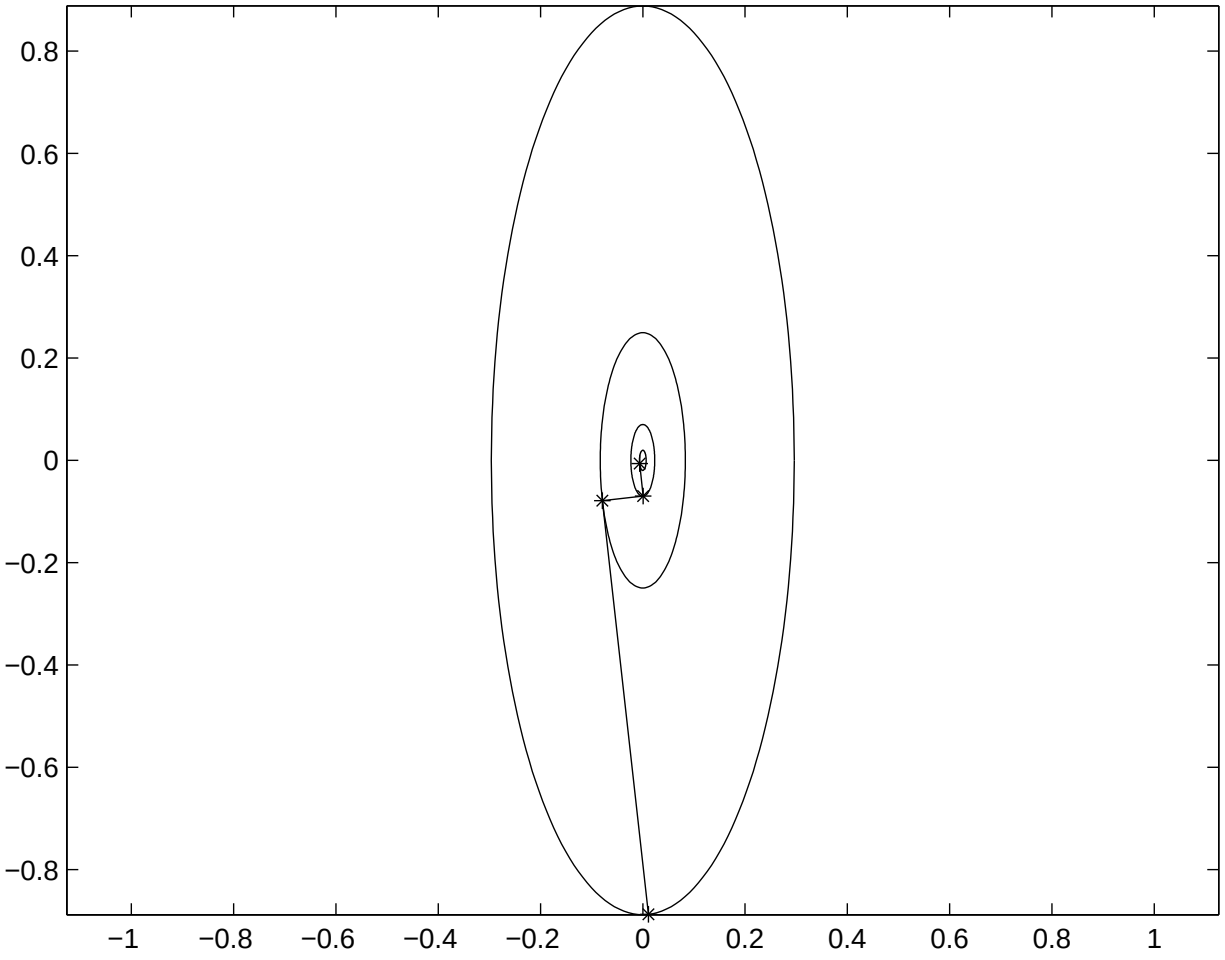
x_0 arbitrary; $r_0 = b - Ax_0$; solve $Mz_0 = r_0$

do $k = 0, 1, \ldots$ until convergence

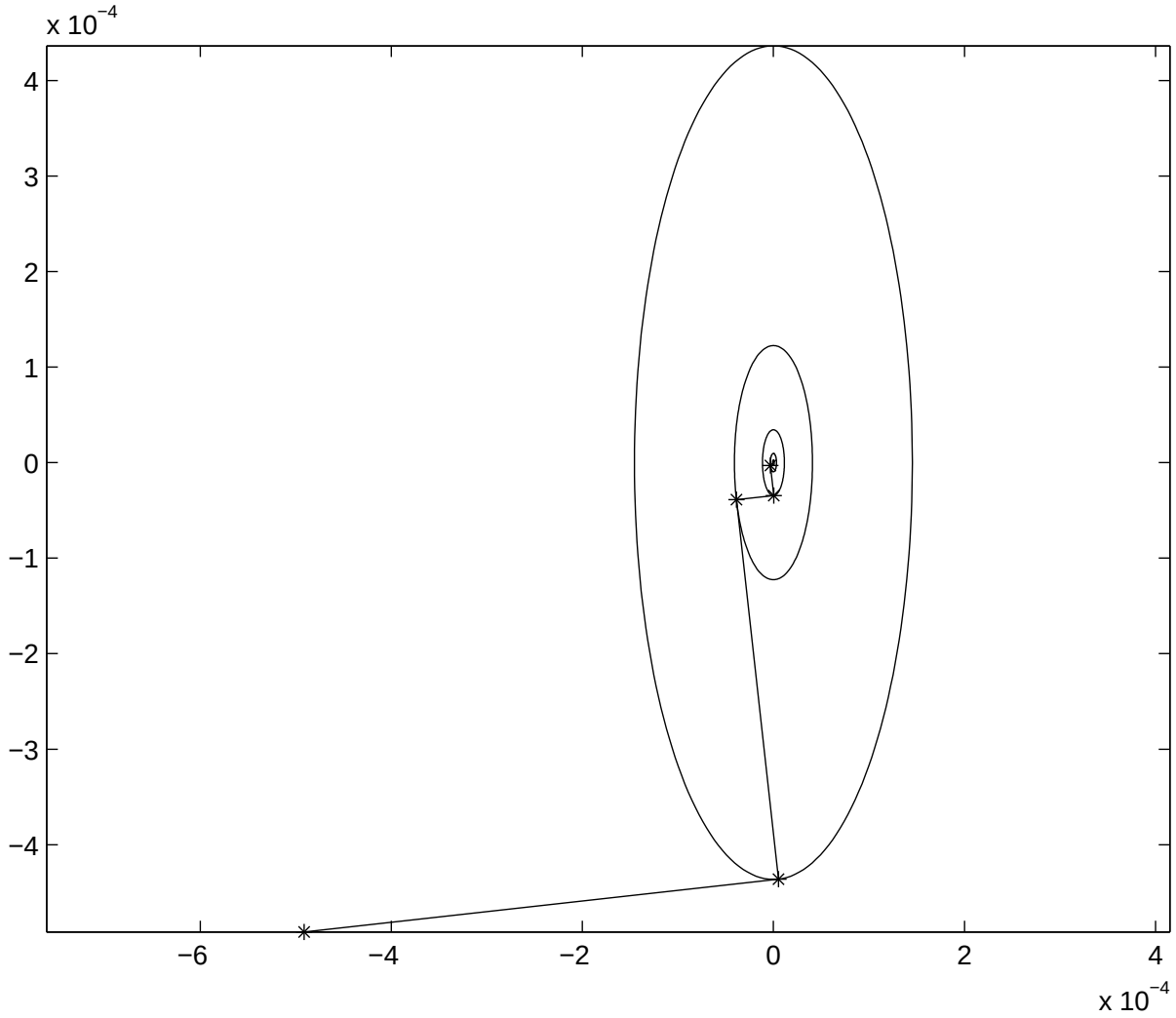
$$
\begin{aligned}
w_k &= Az_k \\
\alpha_k &= \frac{z_k^T r_k}{r_k^T w_k} \\
x_{k+1} &\leftarrow x_k + z_k \alpha_k \\
r_{k+1} &\leftarrow r_k - w_k \alpha_k \\
&\text{solve } Mz_{k+1} = r_{k+1}
\end{aligned}
$$

end

Iterations 2 to 5 for $n = 2$, $\lambda_1 = 9$, $\lambda_2 = 1$



Iterations 7 to 11 for $n = 2$, $\lambda_1 = 9$, $\lambda_2 = 1$, note scale difference



Convergence Rates from Level Curves

Convergence is a function of the eccentricity of the ellipses that define the level curves of Q.

You move in the direction of r until the point at which the line $x_k + \alpha r_k$ is tangent to a level curve. This must be the minimum in that direction.

Very eccentric curves cause the search to ping-pong across the low "valley" rather than following the bottom of the valley.

Convergence Rates

- A is symmetric positive definite
- $\|A\|_2 = \lambda_{max}$ and $\|A^{-1}\|_2 = \lambda_{min}^{-1}$
- $\kappa(A)_2 = \lambda_{max}/\lambda_{min}$
- error is damped based on eccentricity

$$\|e^{(k+1)}\|_A \leq \frac{\kappa(A)_2 - 1}{\kappa(A)_2 + 1}\|e^{(k)}\|_A$$

- note similarity to earlier analysis of stationary method
- preconditioning with some M can improve the convergence considerably by changing the eigenvalues.

Conjugate Directions for $Ax = b$

- $< x,\ y >_A = x^T Ay$ is an inner product.
- $< e,\ e >_A = \|e\|_A^2$ used to define cost $Q(x)$, where $e = x - A^{-1}b$.
- $< x,\ y >_A$ defines conjugacy or A-orthogonality.
- $p_0,\ \ p_1, \ldots,\ p_k$ are A-orthogonal if

$$< p_i,\ p_j >_A \begin{cases} = 0 & \text{if } i \neq j \\ \neq 0 & \text{if } i = j \end{cases}$$

Conjugate Directions for $Ax = b$

- A-orthogonality is a global constraint we relate to the underlying notion of a series of 1-dimensional optimization problems to get
  - efficiency per step
  - minimization of the error, i.e., solving the system $Ax = b$
- Many algorithms possible based on conjugate directions.

Successive Minimization CD

- A-orthogonal directions $p_0, \cdots, p_{n-1} \rightarrow$ bases for subspaces.
- Finite termination since $span(p_0, \cdots, p_{n-1}) = \mathbb{R}^n$.
- Solve a series of minimization problems with solutions x_k

$$\min_{x \in \mathcal{S}_{k-1}} \|x - A^{-1}b\|_A = \min_{x \in \mathcal{S}_{k-1}} Q(x)$$

$$Q(v) = 0.5 v^T A v - v^T b$$

$$\mathcal{S}_{k-1} = x_0 + span(p_0, \cdots, p_{k-1}) = x_0 + \mathcal{R}(P_{k-1}) \subseteq \mathbb{R}^n$$

$$P_i = \begin{pmatrix} p_0 & \cdots & p_i \end{pmatrix} \in \mathbb{R}^{n \times i+1}$$

Successive Minimization CD

Start with a 1-d minimization problem:

$$\min_{\alpha>0} \phi_0(\alpha) = \min_{\alpha>0} Q(x_0 + \alpha p_0) = \min_{x \in \mathcal{S}_0} Q(x)$$

Known solution:

$$x_1 = x_0 + \alpha_0 p_0, \quad \alpha_0 = \frac{p_0^T r_0}{p_0^T A p_0}$$

$$x_1 \in \mathcal{S}_0$$

Successive Minimization CD

$$x_k = \operatorname*{argmin}_{x \in \mathcal{S}_{k-1}} Q(x)$$

- The dimension of the constraint variety increases by one on each step.
- $x_0, x_1, \ldots, x_{k-1}$ are the best approximations to $A^{-1}b$ on known varieties.
- Can those solutions be used to find x_k?
- What yields acceptable efficiency per step?

Successive Minimization CD

- Consider minimizing $Q(v)$ with $v \in S_{k-1}$.
- $\mathcal{S}_{k-1}$ is defined by $\mathcal{S}_{k-2}$ and some new direction p.
- Assume p is arbitrary except that it has some part outside $\mathcal{S}_{k-2}$.
- We have for some $y \in \mathbb{R}^{k-1}$

$$\begin{aligned} v &= x_0 + P_{k-2}y + \alpha p \\ &= z + \alpha p \end{aligned}$$

- y and α are free for the minimization given a fixed p.
- y is free since we may have to alter the components in $\mathcal{S}_{k-2}$, i.e., x_{k-1} may no longer be the proper vector in $\mathcal{S}_{k-2}$'s directions.

Successive Minimization CD

It is straightforward to show that

$$
\begin{aligned}
Q(v) &= 0.5v^T Av - v^T b \\
&= Q(z) + \frac{\alpha^2}{2} p^T Ap - \alpha p^T b + \alpha p^T Az \\
&= Q(z) + \frac{\alpha^2}{2} p^T Ap - \alpha p^T b + \alpha p^T A(x_0 + P_{k-2} y) \\
&= Q(z) + \frac{\alpha^2}{2} p^T Ap - \alpha p^T (b - Ax_0) + \alpha p^T AP_{k-2} y \\
&= Q(z) + \frac{\alpha^2}{2} p^T Ap - \alpha p^T r_0 + \alpha p^T AP_{k-2} y
\end{aligned}
$$

Successive Minimization CD

- By definition, $Q(z)$ is minimized by x_{k-1}, the current approximation to x.
- The term
$$\frac{\alpha^2}{2} p^T A p - \alpha p^T r_0$$
is minimized by considering α only.
- the last term in $Q(v)$ requires both y and α be determined together, i.e., the previous solution, x_{k-1} has no relevance in general.
- Can we remove this term?

Successive Minimization CD

- p was an arbitrary direction up to now.
- Choose p_{k-1} to be any vector such that it is A-orthogonal to $p_0, \cdots, p_{k-2}$ so

$$p_{k-1}^T A P_{k-2} y = 0_{k-1}^T$$

- The last term is eliminated.
- Therefore, by enforcing the global condition of A-orthogonality on the directions $p_0,\ p_1, \ldots, p_{k-1}$,

  we have decoupled the problem on $\mathcal{S}_{k-1}$ to get x_k into a solved problem and a one-dimensional problem

Successive Minimization CD

- Given such a p_{k-1} we can determine the optimal α as

$$\alpha_{k-1} = \frac{p_{k-1}^T r_0}{p_{k-1}^T A p_{k-1}} = \frac{p_{k-1}^T r_{k-1}}{p_{k-1}^T A p_{k-1}}$$

- We can always (in exact arithmetic) produce an A-orthogonal p_{k-1} such that $p_{k-1}^T r_{k-1} \neq 0$ as long as we have not yet solved the system by producing a residual that is 0.

Conjugate Direction Algorithm

Algorithm: CD

Given x_0, let $r_0 = b - Ax_0 = p_0$

loop until convergence

choose p_{k-1} A-orthogonal to $p_0, \cdots, p_{k-2}$

$$\alpha_{k-1} = \frac{p_{k-1}^T r_{k-1}}{p_{k-1}^T A p_{k-1}}$$

$$x_k = x_{k-1} + \alpha_{k-1} p_{k-1}$$

$$r_k = b - Ax_k$$

end loop

Conjugate Gradients

- We still have to specify how to choose a particular p_{k-1}.
- The efficiency of the method also needs to be consdiered and various properties used to reduce the complexity of an iteration.
- One very popular version that accomplishes all of these is the **conjugate gradient algorithm** (CG).
- There are **many** ways to derive CG.
- **Idea:** Combine steepest descent direction with $A-$orthogonality to get p_{k-1}.

Conjugate Gradient Algorithm Derivation

CG chooses p_{k-1} to be the vector that minimizes

$$\|p - r_{k-1}\|_2$$

over all $p \in \mathbb{R}^n$ that are orthogonal to $\mathcal{R}(AP_{k-2})$ where

$$P_{k-2} = \begin{bmatrix} p_0 & p_1 & \dots & p_{k-2} \end{bmatrix}$$

The first condition is based on the notion that r_{k-1} is the steepest gradient direction but we know we must modify it to be A-orthogonal with the earlier directions to accelerate convergence by using the global perspective of $A-$orthogonality.

Conjugate Gradient Algorithm Derivation

We already know how to characterize such a vector. If $z_{min} \in \mathbb{R}^{k-1}$ solves the least squares problem

$$\min_z \|r_{k-1} - AP_{k-2}z\|_2$$

The residual at z_{min} is orthogonal to $\mathcal{R}(AP_{k-2})$ and is the closest such vector to r_{k-1}.

$$p_{k-1} = r_{k-1} - AP_{k-2}z_{min}$$

Efficient Production of Direction Vector

- The A-orthogonality of p_j and their relationship to the residuals can be used to show that the least squares problem defining the next direction vector p_k has significant structure.
- p_k is a linear combination of r_k and only p_{k-1}.
- Specifically,

$$p_k = r_k + \beta_{k-1} p_{k-1}$$

$$\beta_{k-1} = r_k^T r_k / r_{k-1}^T r_{k-1}$$

- This is a key point in the efficiency of CG. x_k, p_k, and r_k are all given by vector triads, i.e., there is no need to combine all previous p_j and r_j.

Conjugate Gradient Efficient Form

Algorithm: CG

x_0 arbitrary; $r_0 = b - Ax_0$; $p_0 = r_0$

$k = 0, 1, \cdots$

$$v_k = Ap_k$$
$$\alpha_k = r_k^T r_k / p_k^T v_k$$
$$x_{k+1} = x_k + \alpha_k p_k$$
$$r_{k+1} = r_k - \alpha_k v_k$$
$$\beta_k = r_{k+1}^T r_{k+1} / r_k^T r_k$$
$$p_{k+1} = r_{k+1} + \beta_k p_k$$

end

Congugate Gradient Method

- Note the low complexity per step.
- This method can be derived from other points of view (including the view in the text).
- It is the standard algorithm to solve large sparse symmetric positive definite linear systems and sparse linear least squares solvers.
- It is the basis for two families of nonsymmetric sparse linear system solvers.
- In practice, it must be preconditioned to work well.
- It can be applied as an accerleration method to other solvers.
- It can be used in very abstract settings, e.g., optimization in vector spaces of functions when solving PDEs.

Useful CG Facts

- The direction vectors p_k, $k = 0, \cdots$ are mutually A-orthogonal.
- The residuals, r_k, $k = 0, \cdots$ are mutually orthogonal
- r_k is orthogonal to $\mathcal{R}(P_{k-1})$. (Galerkin condition)
- Two bases for the space in which x_k resides:

$$\mathcal{S}_k = x_0 + span(p_0, \cdots, p_{k-1}) = x_0 + span(r_0, \cdots, r_{k-1})$$

Convergence of CG

- Efficient step in terms of computations and space like SD.
- In exact arithmetic it is a finite algorithm since

$$x_k = x_0 + \alpha_0 p_0 + \cdots + \alpha_{k-1} p_{k-1}$$

  and $(p_0, \ldots, p_{n-1})$ is a basis for $\mathbb{R}^n$, i.e., the algorithm determines $e^{(0)} = x - x_0$ in terms of this basis.
- This is not good enough and the main convergence results bound $e^{(k)} = x_k - A^{-1}b$.
- CG's behavior under finite arithmetic is fairly complicated and is also the subject of much rigorous literature and folklore.

Convergence and the A-norm

Theorem 12.3. *For CG, $e^{(k)}$ is bounded in terms of, κ, the condition number of A and the initial error $e^{(0)}$ by*

$$\|e^{(k)}\|_A \leq 2\alpha^k \|e^{(0)}\|_A$$

$$\alpha = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$$

Convergence and the 2-norm

Theorem 12.4. *For CG, $e^{(k)}$ is bounded in terms of, κ, the condition number of A and the initial error $e^{(0)}$ by*

$$\|e^{(k)}\|_2 \leq 2\sqrt{\kappa}\alpha^k\|e^{(0)}\|_2$$

$$\alpha = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$$

Eigenvalues and Convergence

Eigenvalues play an important role in characterizing the convergence of CG.

Theorem 12.5. *If A has m distinct eigenvalues, i.e., there are m values $\mu_1, \cdots, \mu_m$ such that*

- *for $1 \leq i \leq n$ there exists j such that $\lambda_i = \mu_j$*
- *for $1 \leq j \leq m$ there exists at least one i such that $\lambda_i = \mu_j$*

then CG converges in at most m steps.

Rules of Thumb

Heuristically we have the following statements:

- CG converges quickly in the A-norm if $\kappa \approx 1$. This implies the spread of eigenvalues is getting small and therefore it "looks" like there are fewer distinct values. Alternatively, it says the steepest descent level curves are going to circles.
- If A is close to a rank r update to the identity then CG is almost converged after r steps

Preconditioning

- Finite termination and the distinct eigenvalue convergence theorems typically do not yield satisfactory convergence in practice.
- It is necessary to alter the system in order to improve the convergence rate.
- We transform the coefficient matrix to have, effectively, fewer distinct eigenvalues.
- There is a tradeoff in the cost of transforming the system – or **preconditioning** the system – and the resulting improvement in performance.

Preconditioned Conjugate Gradient

A and M are symmetric positive definite matrices.

x_0 arbitrary; $r_0 = b - Ax_0$;

solve $Mz_0 = r_0$; $p_0 = z_0$

$k = 0, 1, \cdots$

$$v_k = Ap_k$$
$$\alpha_k = r_k^T z_k / p_k^T v_k$$
$$x_{k+1} = x_k + \alpha_k p_k$$
$$r_{k+1} = r_k - \alpha_k v_k$$

solve $Mz_{k+1} = r_{k+1}$

$$\beta_k = r_{k+1}^T z_{k+1} / r_k^T z_k$$
$$p_{k+1} = z_{k+1} + \beta_k p_k$$

end

Preconditioners

- diagonal or block diagonal (Jacobi or block Jacobi)
- Symmetric Gauss-Seidel and Symmetric SOR
- Approximate inverse: $\|I - M^{-1}A\|_F$ is minimized.
- Polynomial preconditioning: $A^{-1} \approx P(A)$
- Incomplete Cholesky

See text Section 4.3.2 for more discussion of preconditioners.