A Summary of Methods of Solution for Linear Systems

Nate DeMaagd, Kurt O'Hearn

MTH 499-02

April 23, 2013

Outline

- Types of methods

Outline

- Types of methods
- Direct methods and examples

Outline

- Types of methods
- Direct methods and examples
- Indirect methods and examples

Recall: For the linear system $Ax = b$ with $A_{m\times n}$:

Recall: For the linear system $Ax = b$ with $A_{m \times n}$:

System Type	Possible Number of Solutions
Square ($m = n$)	None, Unique
Overdetermined ($m > n$)	None, Unique
Underdetermined ($m < n$)	None, Infinite

Recall: For the linear system $Ax = b$ with $A_{m \times n}$:

System Type	Possible Number of Solutions
Square ($m = n$)	None, Unique
Overdetermined ($m > n$)	None, Unique
Underdetermined ($m < n$)	None, Infinite

Types of Methods

- Direct methods

Recall: For the linear system $Ax = b$ with $A_{m\times n}$:

System Type	Possible Number of Solutions
Square ($m = n$)	None, Unique
Overdetermined ($m > n$)	None, Unique
Underdetermined ($m < n$)	None, Infinite

Types of Methods

- Direct methods
    - Execute a predetermined number of computations to produce a result

Recall: For the linear system $Ax = b$ with $A_{m\times n}$:

System Type	Possible Number of Solutions
Square ($m = n$)	None, Unique
Overdetermined ($m > n$)	None, Unique
Underdetermined ($m < n$)	None, Infinite

Types of Methods

- Direct methods
  - Execute a predetermined number of computations to produce a result
  - Methods: compute A^{-1}, transform A using factorizations/pivoting

Recall: For the linear system $Ax = b$ with $A_{m\times n}$:

System Type	Possible Number of Solutions
Square ($m = n$)	None, Unique
Overdetermined ($m > n$)	None, Unique
Underdetermined ($m < n$)	None, Infinite

Types of Methods

- Direct methods
  - Execute a predetermined number of computations to produce a result
  - Methods: compute A^{-1}, transform A using factorizations/pivoting
- Indirect methods

Recall: For the linear system $Ax = b$ with $A_{m \times n}$:

System Type	Possible Number of Solutions
Square ($m = n$)	None, Unique
Overdetermined ($m > n$)	None, Unique
Underdetermined ($m < n$)	None, Infinite

Types of Methods

- Direct methods
  - Execute a predetermined number of computations to produce a result
  - Methods: compute A^{-1}, transform A using factorizations/pivoting
- Indirect methods
  - Generate a sequence of intermediate results which (hopefully) produce the desired final result

Recall: For the linear system $Ax = b$ with $A_{m\times n}$:

System Type	Possible Number of Solutions
Square ($m = n$)	None, Unique
Overdetermined ($m > n$)	None, Unique
Underdetermined ($m < n$)	None, Infinite

Types of Methods

- Direct methods
  - Execute a predetermined number of computations to produce a result
  - Methods: compute A^{-1}, transform A using factorizations/pivoting
- Indirect methods
  - Generate a sequence of intermediate results which (hopefully) produce the desired final result
  - Methods:

Recall: For the linear system $Ax = b$ with $A_{m \times n}$:

System Type	Possible Number of Solutions
Square ($m = n$)	None, Unique
Overdetermined ($m > n$)	None, Unique
Underdetermined ($m < n$)	None, Infinite

Types of Methods

- Direct methods
  - Execute a predetermined number of computations to produce a result
  - Methods: compute A^{-1}, transform A using factorizations/pivoting
- Indirect methods
  - Generate a sequence of intermediate results which (hopefully) produce the desired final result
  - Methods:
    - General: Richardson, Jacobi, Gauss-Seidel, SOR

Recall: For the linear system $Ax = b$ with $A_{m \times n}$:

System Type	Possible Number of Solutions
Square ($m = n$)	None, Unique
Overdetermined ($m > n$)	None, Unique
Underdetermined ($m < n$)	None, Infinite

Types of Methods

- Direct methods
  - Execute a predetermined number of computations to produce a result
  - Methods: compute A^{-1}, transform A using factorizations/pivoting
- Indirect methods
  - Generate a sequence of intermediate results which (hopefully) produce the desired final result
  - Methods:
    - General: Richardson, Jacobi, Gauss-Seidel, SOR
    - Symmetric, positive definite: steepest descent, conjugate gradient

Recall: For the linear system $Ax = b$ with $A_{m\times n}$:

System Type	Possible Number of Solutions
Square ($m = n$)	None, Unique
Overdetermined ($m > n$)	None, Unique
Underdetermined ($m < n$)	None, Infinite

Types of Methods

- Direct methods
  - Execute a predetermined number of computations to produce a result
  - Methods: compute A^{-1}, transform A using factorizations/pivoting
- Indirect methods
  - Generate a sequence of intermediate results which (hopefully) produce the desired final result
  - Methods:
    - General: Richardson, Jacobi, Gauss-Seidel, SOR
    - Symmetric, positive definite: steepest descent, conjugate gradient

Note: methods can be general or exploit certain matrix characteristics

Developing A Simple Iterative Method

Developing A Simple Iterative Method

- Want to solve: $Ax = b$ for x with $A \in \mathbb{R}^{n \times n}$, A invertible

Developing A Simple Iterative Method

- Want to solve: $Ax = b$ for x with $A \in \mathbb{R}^{n \times n}$, A invertible
- Introduce an invertible "splitting matrix" Q and rearrange to get

$$Qx = (Q - A)x + b$$

Developing A Simple Iterative Method

- Want to solve: $Ax = b$ for x with $A \in \mathbb{R}^{n\times n}$, A invertible
- Introduce an invertible "splitting matrix" Q and rearrange to get

$$Qx = (Q - A)x + b$$

- Now define our iterative process as

$$Qx^{(k)} = (Q - A)x^{(k-1)} + b$$

where $k > 1$ denotes the k^{th} step in the process

Developing A Simple Iterative Method

- Want to solve: $Ax = b$ for x with $A \in \mathbb{R}^{n \times n}$, A invertible
- Introduce an invertible "splitting matrix" Q and rearrange to get

$$Qx = (Q - A)x + b$$

- Now define our iterative process as

$$Qx^{(k)} = (Q - A)x^{(k-1)} + b$$

where $k > 1$ denotes the k^{th} step in the process
  - Want: each successive iteration to produce a better approximation for x (i.e., converge)

Developing A Simple Iterative Method

- Want to solve: $Ax = b$ for x with $A \in \mathbb{R}^{n \times n}$, A invertible
- Introduce an invertible "splitting matrix" Q and rearrange to get

$$Qx = (Q - A)x + b$$

- Now define our iterative process as

$$Qx^{(k)} = (Q - A)x^{(k-1)} + b$$

  where $k > 1$ denotes the k^{th} step in the process
  - Want: each successive iteration to produce a better approximation for x (i.e., converge)
  - Also want: algorithms which guarantee convergence after satisfying some conditions

Developing A Simple Iterative Method

- Want to solve: $Ax = b$ for x with $A \in \mathbb{R}^{n \times n}$, A invertible
- Introduce an invertible "splitting matrix" Q and rearrange to get

$$Qx = (Q - A)x + b$$

- Now define our iterative process as

$$Qx^{(k)} = (Q - A)x^{(k-1)} + b$$

  where $k > 1$ denotes the k^{th} step in the process
  - Want: each successive iteration to produce a better approximation for x (i.e., converge)
  - Also want: algorithms which guarantee convergence after satisfying some conditions
  - To achieve these ends, we seek Q such that:
    - $\|x - x^{(k)}\| \to 0$ rapidly, and
    - $\left[x^{(k)}\right]$ is easy to compute

Developing A Simple Iterative Method

- Want to solve: $Ax = b$ for x with $A \in \mathbb{R}^{n\times n}$, A invertible
- Introduce an invertible "splitting matrix" Q and rearrange to get

$$Qx = (Q - A)x + b$$

- Now define our iterative process as

$$Qx^{(k)} = (Q - A)x^{(k-1)} + b$$

  where $k > 1$ denotes the k^{th} step in the process
  - Want: each successive iteration to produce a better approximation for x (i.e., converge)
  - Also want: algorithms which guarantee convergence after satisfying some conditions
  - To achieve these ends, we seek Q such that:
    - $\|x - x^{(k)}\| \to 0$ rapidly, and
    - $\left[x^{(k)}\right]$ is easy to compute
  - Note: often the initial vector $x^{(0)}$ is an estimate of the solution or arbitrary $(x = 0)$

Why This Iterative Process Works:

Why This Iterative Process Works:

- Recall iterative process $Qx^{(k)} = (Q - A)x^{(k-1)} + b$

Why This Iterative Process Works:

- Recall iterative process $Qx^{(k)} = (Q - A)x^{(k-1)} + b$
- Letting $k \to \infty$, solution is $Qx = (Q - A)x + b$

Why This Iterative Process Works:

- Recall iterative process $Qx^{(k)} = (Q - A)x^{(k-1)} + b$
- Letting $k \to \infty$, solution is $Qx = (Q - A)x + b$
- Assume Q and A nonsingular. So, Q^{-1} and A^{-1} exist and

$$x^{(k)} = (I - Q^{-1}A)x^{(k-1)} + Q^{-1}b$$

Why This Iterative Process Works:

- Recall iterative process $Qx^{(k)} = (Q - A)x^{(k-1)} + b$
- Letting $k \to \infty$, solution is $Qx = (Q - A)x + b$
- Assume Q and A nonsingular. So, Q^{-1} and A^{-1} exist and

$$x^{(k)} = (I - Q^{-1}A)x^{(k-1)} + Q^{-1}b$$

- Taking limit of this gives solution

$$x = (I - Q^{-1}A)x + Q^{-1}b$$

Why This Iterative Process Works:

- Recall iterative process $Qx^{(k)} = (Q - A)x^{(k-1)} + b$
- Letting $k \to \infty$, solution is $Qx = (Q - A)x + b$
- Assume Q and A nonsingular. So, Q^{-1} and A^{-1} exist and

$$x^{(k)} = (I - Q^{-1}A)x^{(k-1)} + Q^{-1}b$$

- Taking limit of this gives solution

$$x = (I - Q^{-1}A)x + Q^{-1}b$$

- Thus $x^{(k)} - x = (I - Q^{-1}A)(x^{(k-1)} - x)$

Why This Iterative Process Works:

- Recall iterative process $Qx^{(k)} = (Q - A)x^{(k-1)} + b$
- Letting $k \to \infty$, solution is $Qx = (Q - A)x + b$
- Assume Q and A nonsingular. So, Q^{-1} and A^{-1} exist and

$$x^{(k)} = (I - Q^{-1}A)x^{(k-1)} + Q^{-1}b$$

- Taking limit of this gives solution

$$x = (I - Q^{-1}A)x + Q^{-1}b$$

- Thus $x^{(k)} - x = (I - Q^{-1}A)(x^{(k-1)} - x)$
- Select vector norm and subordinate norm so that by using the norm and the recursive definition

$$||x^{(k)} - x|| \leq ||I - Q^{-1}A||^k \; ||x^{(0)} - x||$$

Why This Iterative Process Works:

- Recall iterative process $Qx^{(k)} = (Q - A)x^{(k-1)} + b$
- Letting $k \to \infty$, solution is $Qx = (Q - A)x + b$
- Assume Q and A nonsingular. So, Q^{-1} and A^{-1} exist and

$$x^{(k)} = (I - Q^{-1}A)x^{(k-1)} + Q^{-1}b$$

- Taking limit of this gives solution

$$x = (I - Q^{-1}A)x + Q^{-1}b$$

- Thus $x^{(k)} - x = (I - Q^{-1}A)(x^{(k-1)} - x)$
- Select vector norm and subordinate norm so that by using the norm and the recursive definition

$$||x^{(k)} - x|| \leq ||I - Q^{-1}A||^k \; ||x^{(0)} - x||$$

- Thus, if $||I - Q^{-1}A|| < 1$, then $\lim_{k \to \infty} ||x^{(k)} - x|| = 0$

More General Conditions for Iterative Method Convergence

More General Conditions for Iterative Method Convergence

Theorem

The spectral radius of a matrix A, $\rho(A) = \max_{1 \leq i \leq n} |\lambda_i|$, satisfies

$$\rho(A) = \inf_{\|\cdot\|} \|A\|.$$

More General Conditions for Iterative Method Convergence

Theorem

The spectral radius of a matrix A, $\rho(A) = \max_{1 \le i \le n} |\lambda_i|$, *satisfies*

$$\rho(A) = \inf_{\|\cdot\|} \|A\|.$$

Theorem

For the linear system $Ax = b$ *with* A *invertible, define the iteration formula*

$$x^{(k)} = Gx^{(k-1)} + c.$$

The sequence $\left[x^{(k)}\right]$ *will converge to* $(I - G)^{-1}c$ *provided that* $\rho(G) < 1$.

More General Conditions for Iterative Method Convergence

Theorem

The spectral radius of a matrix A, $\rho(A) = \max_{1 \le i \le n} |\lambda_i|$, *satisfies*

$$\rho(A) = \inf_{\|\cdot\|} \|A\|.$$

Theorem

For the linear system $Ax = b$ *with* A *invertible, define the iteration formula*

$$x^{(k)} = Gx^{(k-1)} + c.$$

The sequence $\left[x^{(k)}\right]$ *will converge to* $(I - G)^{-1}c$ *provided that* $\rho(G) < 1$.

Corollary

The iteration formumla

$$Qx^{(k)} = (Q - A)x^{(k-1)} + b$$

will produce a convergent sequence provided that $\rho(I - Q^{-1}A) < 1$.

Method	Q	Iteration Formula: $x^{(k)} = (I - Q^{-1}A)x^{(k-1)} + Q^{-1}b$
Richardson	I	$x^{(k)} = (I - A)x^{(k-1)} + b = x^{(k-1)} + r^{(k-1)}$
Jacobi	D	$x^{(k)} = (I - D^{-1}A)x^{(k-1)} + D^{-1}b$
Gauss-Seidel	L	$x^{(k)} = (I - L^{-1}A)x^{(k-1)} + L^{-1}b$

where

- D: diagonal matrix where $d_{ii} = a_{ii}$
- L: lower triangular matrix where $l_{ij} = a_{ij}, i \geq j$