MATH 3795
Lecture 3. Solving Linear Systems 1

Dmitriy Leykekhman

Fall 2008

Goals

- Review of basic linear algebra (matrix computations).
- Solution of simple linear systems.
- Gaussian elimination.

Matrix-Vector Multiplication

Let $A \in R^{m \times n}$ and $x \in R^n$. The i-th component of the matrix-vector product $y = Ax$ is defined by

$$y_i = \sum_{j=1}^{n} a_{ij} x_j \tag{1}$$

i.e., y_i is the dot product (inner product) of the i-th row of A with the vector x.

Matrix-Vector Multiplication

Let $A \in R^{m \times n}$ and $x \in R^n$. The i-th component of the matrix-vector product $y = Ax$ is defined by

$$y_i = \sum_{j=1}^{n} a_{ij} x_j \tag{1}$$

i.e., y_i is the dot product (inner product) of the i-th row of A with the vector x.

$$\begin{pmatrix} \vdots \\ \hline y_i \\ \hline \vdots \end{pmatrix} = \begin{pmatrix} \vdots & \vdots & & \vdots \\ \hline a_{i1} & a_{i2} & & a_{in} \\ \hline \vdots & \vdots & & \vdots \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

Matrix-Vector Multiplication

Another useful point of view is to look at entire vector $y = Ax$

$$\begin{pmatrix} y_1 \\ \vdots \\ \hline y_i \\ \hline \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & & a_{1n} \\ \vdots & \vdots & & \vdots \\ \hline a_{i1} & a_{i2} & & a_{in} \\ \hline \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

$$= x_1 \begin{pmatrix} a_{11} \\ \vdots \\ \hline a_{i1} \\ \hline \vdots \\ a_{m1} \end{pmatrix} + x_2 \begin{pmatrix} a_{12} \\ \vdots \\ \hline a_{i2} \\ \hline \vdots \\ a_{m2} \end{pmatrix} + \cdots + x_n \begin{pmatrix} a_{1n} \\ \vdots \\ \hline a_{in} \\ \hline \vdots \\ a_{mn} \end{pmatrix}.$$

Thus, y is a linear combination of the columns of matrix A.

Matrix-Matrix Multiplication

If $A \in \mathbb{R}^{m \times p}$ and $B \in \mathbb{R}^{p \times n}$, then $AB = C \in \mathbb{R}^{m \times n}$ defined by

$$c_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj},$$

i.e. the ij-th element of the product matrix is the dot product between i-th row of A and j-th column of B.

$$\begin{pmatrix} & & \\ & c_{ij} & \\ & & \end{pmatrix} = \begin{pmatrix} \hline a_{i1} & \cdots & a_{ip} \\ \hline \end{pmatrix} \begin{pmatrix} & \Big| & b_{1j} & \Big| & \\ & \Big| & \vdots & \Big| & \\ & \Big| & b_{pj} & \Big| & \end{pmatrix}$$

Thus, y is a linear combination of the columns of matrix A.

Matrix-Matrix Multiplication

Another useful point of view is to look at j-th column of C

$$\begin{pmatrix} c_{1j} \\ \vdots \\ c_{ij} \\ \vdots \\ c_{mj} \end{pmatrix} = b_{1j} \begin{pmatrix} a_{11} \\ \vdots \\ a_{i1} \\ \vdots \\ a_{m1} \end{pmatrix} + b_{2j} \begin{pmatrix} a_{12} \\ \vdots \\ a_{i2} \\ \vdots \\ a_{m2} \end{pmatrix} + \cdots + b_{nj} \begin{pmatrix} a_{1n} \\ \vdots \\ a_{in} \\ \vdots \\ a_{mn} \end{pmatrix}.$$

Thus, j-th column of C is a linear combination of the columns of matrix A.

Matrix-Matrix Multiplication (cont.)

Sometimes it is useful to consider matrices partitioned into blocks. For example,

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \quad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

with $m_1 + m_2 = m$, $p_1 + p_2 = p$, and $n_1 + n_2 = n$.

Matrix-Matrix Multiplication (cont.)

Sometimes it is useful to consider matrices partitioned into blocks. For example,

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \quad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

with $m_1 + m_2 = m$, $p_1 + p_2 = p$, and $n_1 + n_2 = n$.

This time $C = AB$ can be expressed as

$$C = \begin{pmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{21} + A_{22}B_{21} & A_{12}B_{12} + A_{22}B_{22} \end{pmatrix}.$$

Matrix-Matrix Multiplication (cont.)

For example, if

$$A = \left(\begin{array}{cc|c} 1 & 2 & 3 \\ 4 & 5 & 6 \\ \hline 7 & 8 & 9 \end{array}\right) \quad B = \left(\begin{array}{c|c} 1 & 2 \\ 3 & 4 \\ \hline 5 & 6 \end{array}\right),$$

then

Matrix-Matrix Multiplication (cont.)

For example, if

$$A = \left(\begin{array}{cc|c} 1 & 2 & 3 \\ 4 & 5 & 6 \\ \hline 7 & 8 & 9 \end{array}\right) \quad B = \left(\begin{array}{c|c} 1 & 2 \\ 3 & 4 \\ \hline 5 & 6 \end{array}\right),$$

then

This time $C = AB$ can be expressed as

$$C = \left(\begin{array}{c|c} \begin{pmatrix} 1 & 2 \\ 4 & 5 \end{pmatrix}\begin{pmatrix} 1 \\ 3 \end{pmatrix} + \begin{pmatrix} 3 \\ 6 \end{pmatrix} 5 & \begin{pmatrix} 1 & 2 \\ 4 & 5 \end{pmatrix}\begin{pmatrix} 2 \\ 4 \end{pmatrix} + \begin{pmatrix} 3 \\ 6 \end{pmatrix} 6 \\ \hline \begin{pmatrix} 7 & 8 \end{pmatrix}\begin{pmatrix} 1 \\ 3 \end{pmatrix} + 9 \cdot 5 & \begin{pmatrix} 7 & 8 \end{pmatrix}\begin{pmatrix} 2 \\ 4 \end{pmatrix} + 9 \cdot 6 \end{array}\right)$$

$$= \left(\begin{array}{c|c} 22 & 28 \\ 49 & 64 \\ \hline 76 & 100 \end{array}\right).$$

Basic Linear Algebra Subroutines (BLAS)

- There are many mathematically equivalent ways to organize basic linear algebra tasks, such as matrix-vector multiplications, matrix-matrix-multiplications, etc.

Basic Linear Algebra Subroutines (BLAS)

- There are many mathematically equivalent ways to organize basic linear algebra tasks, such as matrix-vector multiplications, matrix-matrix-multiplications, etc.
- Which one is best depends depends on the programming language and the type of computer used (parallel vs. serial, memory hierarchies,...).

Basic Linear Algebra Subroutines (BLAS)

- There are many mathematically equivalent ways to organize basic linear algebra tasks, such as matrix-vector multiplications, matrix-matrix-multiplications, etc.
- Which one is best depends depends on the programming language and the type of computer used (parallel vs. serial, memory hierarchies,...).
- Write code in terms of Basic Linear Algebra Subroutines (BLAS)
  BLAS level 1: vector operations such as $x + y$.
  BLAS level 2: matrix vector operations such as Ax.
  BLAS level 3: matrix matrix operations such as AB.

Basic Linear Algebra Subroutines (BLAS)

- There are many mathematically equivalent ways to organize basic linear algebra tasks, such as matrix-vector multiplications, matrix-matrix-multiplications, etc.
- Which one is best depends depends on the programming language and the type of computer used (parallel vs. serial, memory hierarchies,...).
- Write code in terms of Basic Linear Algebra Subroutines (BLAS)
  BLAS level 1: vector operations such as $x + y$.
  BLAS level 2: matrix vector operations such as Ax.
  BLAS level 3: matrix matrix operations such as AB.
- Use Automatically Tuned Linear Algebra Software (ATLAS) http://math-atlas.sourceforge.net/ or other tuned (to the computer system of interest) BLAS.

Solution of Linear Systems

- Let $A \in \mathbb{R}^{m \times n}$ and $x \in \mathbb{R}^n$. Then Ax is a linear combination of the columns of A.
  Hence $Ax = b$ has a solution if $b \in \mathbb{R}^m$ can be written as a linear combination of the columns of A.

Solution of Linear Systems

- Let $A \in \mathbb{R}^{m \times n}$ and $x \in \mathbb{R}^n$. Then Ax is a linear combination of the columns of A.
  Hence $Ax = b$ has a solution if $b \in \mathbb{R}^m$ can be written as a linear combination of the columns of A.
- **Solvability**:
  $Ax = b$ is solvable for every $b \in \mathbb{R}^m$ iff the columns of A span R^m (necessary $n \geq m$).

Solution of Linear Systems

- Let $A \in \mathbb{R}^{m \times n}$ and $x \in \mathbb{R}^n$. Then Ax is a linear combination of the columns of A.
  Hence $Ax = b$ has a solution if $b \in \mathbb{R}^m$ can be written as a linear combination of the columns of A.
- **Solvability**:
  $Ax = b$ is solvable for every $b \in \mathbb{R}^m$ iff the columns of A span R^m (necessary $n \geq m$).
- **Uniqueness**:
  If $Ax = b$ has a solution, then the solution is unique iff the columns of A are linearly independent (necessary $n \leq m$).

Solution of Linear Systems

- Let $A \in \mathbb{R}^{m \times n}$ and $x \in \mathbb{R}^n$. Then Ax is a linear combination of the columns of A.
  Hence $Ax = b$ has a solution if $b \in \mathbb{R}^m$ can be written as a linear combination of the columns of A.
- **Solvability**:
  $Ax = b$ is solvable for every $b \in \mathbb{R}^m$ iff the columns of A span R^m (necessary $n \geq m$).
- **Uniqueness**:
  If $Ax = b$ has a solution, then the solution is unique iff the columns of A are linearly independent (necessary $n \leq m$).
- Hence, for any $b \in \mathbb{R}^m$, the system $Ax = b$ has a unique solution iff $n = m$ and the columns of A are linearly independent.

Transpose of a Matrix

- Let

$$A = \left(\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ \hline 7 & 8 & 9 \end{array}\right) \qquad .$$

Transpose of a Matrix

- Let

$$A = \left(\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ \hline 7 & 8 & 9 \end{array}\right) \quad \text{then} \quad A^T = \left(\begin{array}{cc|c} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{array}\right).$$

- If $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times k}$, then

$$(AB)^T = B^T A^T.$$

More generally,

$$(A_1 A_2 \ldots A_j)^T = A_j^T \ldots A_2^T A_1^T.$$

Transpose of a Matrix

- Let

$$A = \left(\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ \hline 7 & 8 & 9 \end{array}\right) \quad \text{then} \quad A^T = \left(\begin{array}{cc|c} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{array}\right).$$

- If $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times k}$, then

$$(AB)^T = B^T A^T.$$

  More generally,

$$(A_1 A_2 \dots A_j)^T = A_j^T \dots A_2^T A_1^T.$$

- If $A \in \mathbb{R}^{n\times n}$ is invertible, then A^T is invertible and

$$(A^T)^{-1} = (A^{-1})^T.$$

  We will write A^{-T}.

Solution of Triangular Systems

- A matrix $L \in \mathbb{R}^{n \times n}$ is called *lower triangular matrix* if all matrix entries above the diagonal are equal to zero, i.e., if

$$l_{ij} = 0 \quad \text{for} \quad j > i.$$

Solution of Triangular Systems

- A matrix $L \in \mathbb{R}^{n \times n}$ is called *lower triangular matrix* if all matrix entries above the diagonal are equal to zero, i.e., if

$$l_{ij} = 0 \quad \text{for} \quad j > i.$$

- A matrix $U \in \mathbb{R}^{n \times n}$ is called *upper triangular matrix* if all matrix entries below the diagonal are equal to zero, i.e., if

$$l_{ij} = 0 \quad \text{for} \quad i > j.$$

Solution of Triangular Systems

- A matrix $L \in \mathbb{R}^{n \times n}$ is called *lower triangular matrix* if all matrix entries above the diagonal are equal to zero, i.e., if

$$l_{ij} = 0 \quad \text{for} \quad j > i.$$

- A matrix $U \in \mathbb{R}^{n \times n}$ is called *upper triangular matrix* if all matrix entries below the diagonal are equal to zero, i.e., if

$$l_{ij} = 0 \quad \text{for} \quad i > j.$$

- A Linear system with lower (upper) triangular matrix can be solved by forward substitution (backward substitution).

Solution of Triangular Systems

Example

Consider the system

$$\begin{pmatrix} 2 & 0 & 0 \\ 1 & 4 & 0 \\ 4 & 3 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 2 \\ 5 \end{pmatrix}$$

Forward substitution gives

$$x_1 = 4/2 = 2, \quad x_2 = (2 - 1 \cdot 2)/4 = 0, \quad x_3 = (5 - 4 \cdot 2 - 3 \cdot 0)/3 = -1.$$

Solution of Triangular Systems

Example

Consider the system

$$\begin{pmatrix} 2 & 4 & 2 \\ 0 & -2 & 4 \\ 0 & 0 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 2 \\ 6 \end{pmatrix}$$

Backward substitution gives

$$x_3 = 6/3 = 2, \quad x_2 = (2-4\cdot 2)/(-2) = 3, \quad x_1 = (4-2\cdot 2-4\cdot 3)/2 = -6.$$

Solution of Triangular Systems. Back substitution.

Solution of Upper Triangular Systems (Row-Oriented Version).
Input: Upper triangular matrix $U \in \mathbb{R}^{n \times n}$, right hand side vector $b \in \mathbb{R}^n$.
Output: Solution $x \in \mathbb{R}^n$ of $Ux = b$.
Mathematically,

$$x_i = \left(b_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) / u_{ii}, \quad \text{if} \quad u_{ii} \neq 0.$$

MATLAB code that overwrites b with the solution to $Ux = b$.

```
if all(diag(u)) == 0
        disp('the matrix is singular')
else
    b(n) = b(n)/U(n,n);
    for i = n-1:-1:1
        b(i)= (b(i) - U(i,i+1:n)*b(i+1:n))/U(i,i);
    end
end
```

Solution of Triangular Systems. Back substitution.

Solution of Upper Triangular Systems (Column-Oriented Version).
Input: Upper triangular matrix $U \in \mathbb{R}^{n \times n}$, right hand side vector $b \in \mathbb{R}^n$.
Output: Solution $x \in \mathbb{R}^n$ of $Ux = b$.
MATLAB code that overwrites b with the solution to $Ux = b$.

```
if all(diag(u)) == 0
        disp('the matrix is singular')
else

    for  j = n:-1:2
            b(j) = b(j)/U(j,j) ;
            b(1:j-1) = b(1:j-1) - b(j)*U(1:j-1,j);
    end

    b(1) = b(1)/U(1,1);
end
```

Gaussian Elimination

Gaussian elimination for the solution of a linear system transforms the system $Ax = b$ into an equivalent system with upper triangular matrix.

This is done by applying three types of transformations to the augmented matrix $(A|b)$.

- Type 1: Replace an equation with the sum of the same equation and a multiple of another equation;
- Type 2: Interchange two equations; and
- Type 3: Multiply an equation by a nonzero number.

Once the augmented matrix $(A|b)$ is transformed into $(U|y)$, where U is an upper triangular matrix, we can use the techniques discussed previously to solve this transformed system $Ux = y$.

Gaussian Elimination (cont.)

Consider a linear system $Ax = b$ with

$$A = \begin{pmatrix} 1 & 2 & -1 \\ 2 & 1 & 0 \\ -1 & 2 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}.$$

The augmented system is

$$\left(\begin{array}{ccc|c} 1 & 2 & -1 & 0 \\ 2 & 1 & 0 & 2 \\ -1 & 2 & 2 & 1 \end{array}\right).$$

Step 1.

$$\left(\begin{array}{ccc|c} 1 & 2 & -1 & 0 \\ 0 & -3 & 2 & 2 \\ 0 & 4 & 1 & 1 \end{array}\right) \qquad \begin{array}{l} (1) \\ (2) \leftarrow (2) - 2 * (1) \\ (3) \leftarrow (3) + 1 * (2). \end{array}$$

Gaussian Elimination (cont.)

Consider a linear system $Ax = b$ with

$$A = \begin{pmatrix} 1 & 2 & -1 \\ 2 & 1 & 0 \\ -1 & 2 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}.$$

The augmented system is

$$\left(\begin{array}{ccc|c} 1 & 2 & -1 & 0 \\ 2 & 1 & 0 & 2 \\ -1 & 2 & 2 & 1 \end{array}\right).$$

Step 2.

$$\left(\begin{array}{ccc|c} 1 & 2 & -1 & 0 \\ 0 & -3 & 2 & 2 \\ 0 & 0 & 11/3 & 11/3 \end{array}\right) \qquad \begin{array}{l} (1) \\ (2) \\ (3) \leftarrow (3) + 4/3 * (2). \end{array}$$

Solving the triangular system we obtain $x_3 = 1$, $x_2 = 0$, $x_1 = 1$.

Gaussian Elimination

We need to modify Gaussian elimination for two reasons

- improve numerical stability (change how we perform pivoting)
- make it more versatile (leads to LU-decomposition)

Gaussian Elimination (cont.)

Partial pivoting: In step i, find row j with $j > i$ such that $|a_{ji}| \geq |a_{ki}|$ for all $k > i$ and exchange rows i and j. Such numbers a_{ji} we call pivots. Again consider the augmented system

$$\left(\begin{array}{ccc|c} 1 & 2 & -1 & 0 \\ 2 & 1 & 0 & 2 \\ -1 & 2 & 2 & 1 \end{array} \right).$$

Step 1.

$$\left(\begin{array}{ccc|c} 2 & 1 & 0 & 2 \\ 1 & 2 & -1 & 0 \\ -1 & 2 & 2 & 1 \end{array} \right) \qquad \begin{array}{l} (1) \leftarrow (2) \\ (2) \leftarrow (1) \\ (3). \end{array}$$

Gaussian Elimination (cont.)

Partial pivoting: In step i, find row j with $j > i$ such that $|a_{ji}| \geq |a_{ki}|$ for all $k > i$ and exchange rows i and j. Such numbers a_{ji} we call pivots.
Again consider the augmented system

$$\left(\begin{array}{ccc|c} 1 & 2 & -1 & 0 \\ 2 & 1 & 0 & 2 \\ -1 & 2 & 2 & 1 \end{array} \right).$$

Step 2.

$$\left(\begin{array}{ccc|c} 2 & 1 & 0 & 2 \\ 0 & 3/2 & -1 & -1 \\ 0 & 5/2 & 2 & 2 \end{array} \right) \qquad \begin{array}{l} (1) \\ (2) \leftarrow (2) - 1/2 * (1) \\ (3) \leftarrow (3) - (-1/2) * (2). \end{array}$$

Gaussian Elimination (cont.)

Partial pivoting: In step i, find row j with $j > i$ such that $|a_{ji}| \geq |a_{ki}|$ for all $k > i$ and exchange rows i and j. Such numbers a_{ji} we call pivots. Again consider the augmented system

$$\left(\begin{array}{ccc|c} 1 & 2 & -1 & 0 \\ 2 & 1 & 0 & 2 \\ -1 & 2 & 2 & 1 \end{array}\right).$$

Step 3.

$$\left(\begin{array}{ccc|c} 2 & 1 & 0 & 2 \\ 0 & 5/2 & 2 & 2 \\ 0 & 3/2 & -1 & -1 \end{array}\right) \qquad \begin{array}{l} (1) \\ (2) \leftarrow (3) \\ (3) \leftarrow (2). \end{array}$$

Gaussian Elimination (cont.)

Partial pivoting: In step i, find row j with $j > i$ such that $|a_{ji}| \geq |a_{ki}|$ for all $k > i$ and exchange rows i and j. Such numbers a_{ji} we call pivots. Again consider the augmented system

$$\left(\begin{array}{rrr|r} 1 & 2 & -1 & 0 \\ 2 & 1 & 0 & 2 \\ -1 & 2 & 2 & 1 \end{array} \right).$$

Step 4.

$$\left(\begin{array}{rrr|r} 2 & 1 & 0 & 2 \\ 0 & 5/2 & 2 & 2 \\ 0 & 0 & -\frac{11}{8} & -\frac{11}{8} \end{array} \right) \quad \begin{array}{l} (1) \\ (2) \\ (3) \leftarrow (3) - \frac{3/2}{5/2} * (2). \end{array}$$

Solving the triangular system we obtain $x_3 = 1$, $x_2 = 0$, $x_1 = 1$.

Gaussian Elimination

Why do we need partial pivoting?

Gaussian Elimination

Why do we need partial pivoting?

It doesn't matter if we use exact arithmetic. Without partial pivoting with floating point arithmetic the method can be unstable

Gaussian Elimination

Why do we need partial pivoting?

It doesn't matter if we use exact arithmetic. Without partial pivoting with floating point arithmetic the method can be unstable

How expensive is Gaussian elimination?

Gaussian Elimination

Why do we need partial pivoting?

It doesn't matter if we use exact arithmetic. Without partial pivoting with floating point arithmetic the method can be unstable

How expensive is Gaussian elimination?

Using

$$\sum_{j=1}^{n} j^2 = \frac{n(n+1)(2n+1)}{6}$$

we can calculate that for large n the number of flops in the Gaussian elimination with partial pivoting approximately equal to $2n^3/3$.