Math 381 - Fall 2022

Jay Newby

University of Alberta

Week 8

Last Week

1. Review of linear algebra
2. Properties of matrices
3. Special matrices
4. Linear systems
5. Vector and matrix norms

This Week

1. Direct methods for solving linear systems
    - Forward and Backward Substitution
    - Gaussian elimination
    - Gaussian elimination with partial pivoting (briefly)
    - LU decomposition
    - Condition number of a matrix
    - Error bounds
    - Backward stability analysis

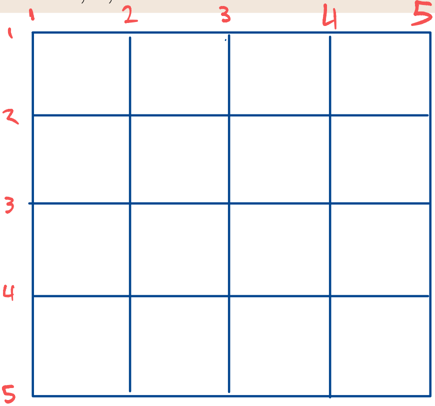
Overview

Direct solvers

Let $A \in \mathbb{R}^{n\times n}$ be nonsingular and let $b \in \mathbb{R}^n$. We want to discuss algorithms that solve (up to rounding error) the linear system of equations $Ax = b$ for the vector x. We will discuss error analysis and stability analysis of the above problem.

Motivation: the curse of dimensionality $\nabla^2 u = f$

Suppose we want to solve a PDE that involves independent variable $x \in \mathbb{R}^d$, where $d = 1, 2, 3$.



Assume m^d grid points. Many numerical schemes will result in a linear system, where the matrix is $n \times n$ with $n = m^d$.

Motivation: the curse of dimensionality

Assume m^d grid points. Many numerical schemes will result in a linear system, where the matrix is $n \times n$ with $n = m^d$.

Example: $m = 10^2$, $d = 3$

* matrix is $n \times n$, $n = m^d = 10^6$
* has n^2 entries, $n^2 = m^{2d} = 10^{12}$
* 1 float64 = 8 bytes

memory $= 10^{13}$ bytes $= 10$TB

* Solving a linear system requires n^2 flops

$n^3 = m^{6d} = 10^{18}$ flops

* suppose a CPU can do $10^{10} \frac{\text{flops}}{\text{sec}}$

$\frac{10^{18}}{10^{10}} = 10^8$ sec ≈ 3 years

Some specialized methods for the 2D Poisson problem

A wide range of methods exist. Methods tailored to take advantage of structure of the matrix are better than general methods (like Gaussian elimination).

- Direct methods $O(m^6)$ worst case $\approx$ 3 min
  - LU decomposition $O(n^3) = O(m^{3d})$
  - Nested dissection (for 2D stencils) $O(m^3)$
  - Fast 2D Poisson solvers $O(m^2 \log(m))$ $\approx$ 10 μs
- Iterative methods
  - Jacobi and Gauss-Seidel $O(m^4 \log(m))$
  - Conjugate Gradient $O(m^3)$
  - Multigrid $O(m^2)$

Let's start with algorithms for triangular matrices

Example

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ & a_{2,2} & a_{2,3} \\ & & a_{3,3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$a_{33} x_3 = b_3 \Rightarrow x_3 = \frac{b_3}{a_{33}}$$

$$a_{22} x_2 + a_{23} x_3 = b_2 \Rightarrow x_2 = \frac{b_2 - a_{23} x_3}{a_{22}}$$

$$a_{11} x_1 = b_3 - a_{12} x_2 - a_{13} x_3 \qquad x_3 = \frac{b_3 - a_{12} x_2 - a_{13} x_3}{a_{11}}$$

Backward substitution for upper triangular matrices

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ & a_{2,2} & a_{2,3} & \cdots & a_{2,n} \\ & & a_{3,3} & \cdots & a_{3,n} \\ & & & \ddots & \vdots \\ & & & & a_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

$O(n^2)$ flops

Backward substitution formula (recursive sequence)

For $x_n, x_{n-1}, x_{n-2}, \ldots, x_1$

for k in arange(n)[::-1]

$$x_n = \frac{b_n}{a_{n,n}}$$

$$x_k = \frac{b_k - \sum_{j=k+1}^{n} a_{kj} x_j}{a_{kk}}$$

Forward substitution for lower triangular matrices

Forward substitution formula (recursive sequence)

For $x_1, x_2, x_3, \ldots, x_n$

$$\begin{bmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ \vdots & & \ddots & \\ a_{n1} & \cdots & \cdot & a_{nn} \end{bmatrix}$$

$$x_1 = \frac{b_1}{a_{1,1}}$$

$$x_k = \frac{b_k - \sum_{j=1}^{k-1} a_{kj} x_j}{a_{kk}}$$

LU decomposition

The LU decomposition

Let $A \in \mathbb{R}^{n \times n}$ be nonsingular. The LU decomposition of A is given by $A = LU$ where

$$L = \begin{bmatrix} 1 & & & & \\ l_{2,1} & 1 & & & \\ l_{3,1} & l_{3,2} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ l_{n,1} & l_{n,2} & \cdots & l_{n,n-1} & 1 \end{bmatrix}, \qquad U = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & \cdots & u_{1,n} \\ & u_{2,2} & u_{2,3} & \cdots & u_{2,n} \\ & & u_{3,3} & \cdots & u_{3,n} \\ & & & \ddots & \vdots \\ & & & & u_{n,n} \end{bmatrix}$$

unit lower triangular

Solving a linear system with the LU decomposition

Claim:

Let $A \in \mathbb{R}^{n\times n}$ be nonsingular and let $b \in \mathbb{R}^n$. Given the LU decomposition of A is given by $A = LU$. The solution to the equation $Ax = b$ is found in two steps:

1. Forward substitution to solve $Ly = b$ for y
2. Backward substitution to solve $Ux = y$ for x

The operation count is $O(n^2)$

Computing LU is $O(n^3)$ flops

Proof: $Ax = b \Rightarrow LUx = b$. Since $Ux = y$

$\Rightarrow Ly = b$

Step (1) $Ly = b$

Step (2) $Ux = y$

Computing the determinant with the LU decomposition

Claim:

Given square matrices $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times n}$ we have that

$$\det(AB) = \det(A)\det(B).$$

Claim:

Given square matrices $A \in \mathbb{R}^{n\times n}$ that is either upper or lower triangular (or diagonal), we have that

$$\det(A) = \prod_{i=1}^{n} a_{ii}.$$

Another fact:

$$\det(A) = \prod_{i=1}^{n} \lambda_i$$

Proof:

$$\left| \begin{matrix} a_{11} & \cdots & \cdots & a_{1n} \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{nn} \end{matrix} \right| = a_{11} \left| \begin{bmatrix} a_{22} & \cdots & a_{2n} \\ & a_{33} & \vdots \\ & & \ddots & \\ & & & a_{nn} \end{bmatrix} \right| = \cdots$$

Computing the determinant with the LU decomposition

Claim:

Given $A \in \mathbb{R}^{n \times n}$ and an LU decomposition $A = LU$, we have that

$$\det(A) = \prod_{i=1}^{n} u_{ii}.$$

Proof:

$$\det(A) = \det(LU) = \det(L)\det(U)$$
$$= \det(U) = \prod_{i=1}^{n} u_{ii}$$

1

n-1 flops

Gaussian Elimination

Example

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

Ex 1

$$\left[\begin{array}{ccc|c} 1 & -1 & 3 & 2 \\ 1 & 1 & 0 & 4 \\ 3 & -2 & 1 & 1 \end{array}\right] \begin{array}{c} \\ -r_1^T \\ \\ \end{array} \quad \left[\begin{array}{ccc|c} 1 & -1 & 3 & 2 \\ 0 & 2 & -3 & 2 \\ 3 & -2 & 1 & 1 \end{array}\right] \begin{array}{c} \\ \\ -3r_1^T \end{array} \quad \left[\begin{array}{ccc|c} 1 & -1 & 3 & 2 \\ 0 & 2 & -3 & 2 \\ 0 & 1 & -2 & 0 \end{array}\right]$$

Ex 2

$$\begin{bmatrix} r_1^T \\ \hline r_2^T \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} b_1 \\ \hline b_2 \end{bmatrix} \qquad r_1^T x = b_1 \ ; \ r_2^T x = b_2$$

$$\begin{bmatrix} r_1^T \\ \hline r_2^T - \alpha r_1^T \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = \boxed{\begin{matrix} b_1 \\ \hline b_2 - \alpha b_1 \end{matrix}} \qquad (r_2^T - \alpha r_1^T)x = r_2^T x - \alpha r_1^T x = b_2 - \alpha b_1$$

Gaussian Elimination for tridiagonal matrices

$$\begin{bmatrix} a_1 & c_1 & & & & \\ b_2 & a_2 & c_2 & & & \\ & b_3 & a_3 & c_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & c_{n-1} \\ & & & & b_n & a_n \end{bmatrix} - \frac{b_2}{a_1} r_1^T$$

$$\begin{bmatrix} a_1 & c_1 & & & & \\ 0 & \underbrace{a_2 - \frac{b_2}{a_1} c_1}_{=u_2} & c_2 & & & \\ & b_3 & a_3 & c_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & c_{n-1} \\ & & & & b_n & a_n \end{bmatrix} - \frac{b_3}{u_2} r_2^T$$

Goal: compute U

$$\begin{bmatrix} u_1 & c_1 & & \\ & u_2 & \ddots & \\ & & \ddots & c_{n-1} \\ & & & u_n \end{bmatrix}$$

need u_i

$u_1 = a_1$

$u_2 = a_2 - \frac{b_2}{u_1} c_1$

$\vdots$

$u_i = f(u_{i-1})$

Gaussian Elimination general case

$$A = LU$$

$$\Rightarrow L^{-1}A = U$$

Transforms A into U

$$L_{n-1}L_{n-2}\cdots L_2 L_1 A = U$$