BST 234: Lab - 8

Divy Kangeyan

March 28, 2018

Forward and backward error

- **Forward Error**: Let x be a real number and let $f : \mathbb{R} \to \mathbb{R}$ be a function. If $\hat{y}$ is a real number that is an approximation to y = f(x), then forward error in $\hat{y}$ is the difference $\Delta y = \hat{y} - y$
- **Backward Error**: Let x be a real number and let $f : \mathbb{R} \to \mathbb{R}$ be a function. If $\hat{y}$ is a real number that is an approximation to y = f(x), that is $\hat{y} = f(\hat{x})$ for some real number $\hat{x}$ then backward error is the difference $\Delta x = \hat{x} - x$

Forward and backward error

- Question: sin(x) can be approximated by following expression $sin(x) = x - \frac{x^3}{3!}$. Calculate backward and forward error for sin(0.1). Hint: arcsin(0.09983333) $=$ 0.09999991

Matrix Norm

- Defn: $||A|| = max_{x \neq 0} \frac{||Ax||}{||x||}$
- Depending on the vector norm being used matrix norm would differ
- Matrix norm corresponding to vector 1-norm $||A||_1 = max_j \sum_{i=1}^{n} |a_{ij}|$
- Matrix norm corresponding to vector ∞-norm $||A||_\infty = max_i \sum_{i=1}^{n} |a_{ij}|$
- Some matrix norm identities in Python

Condition number

- Defn: $cond(A) = ||A|| \times ||A^{-1}||$
- if A is singular cond(A) $= \infty$
- Geometric interpretation: Ratio of maximum stretching to maximum shrinking a matrix does to any nonzero vector
- Python demonstration with some condition number identities

Solving systems of linear equations - Gaussian Elimination

- For $Ax = b$ where A is a non-singular matrix and b is an arbitrary vector, there exists a unique solution for x
- Easier to solve is A is a triangular matrix
- For non-triangular matrix A, it can be solved by factorization i.e. LU factorization

LU Factorization Algorithm

- For matrix A: set $U = A$ and $L = I$
- $n = rank(A)$
- ```
  for i = 1:n-1
     for j = i+1:n
        L[j,i] = U[j,i]/U[i,i]
        U[j,i:n] = U[j,i:n] - L[j,i] U[i,i:n]
     end
  end
  ```
- Homework: LU/Gaussian elimination with pivoting
  ```

SPD Matrices and Cholesky factorization

- Defn for SPD: $n \times n$ real symmetric matrix M is positive definite if $z^T Mz > 0$ for every non-zero column vector z of n real numbers.
- If a matrix A is SPD it's LU decomposition will be: $A = LL^T$
- If a matrix is SPD then performing Cholesky factorization reduces the time and memory requirements compared to usual LU decomposition

Cholesky factorization

for an SPD matrix A set $L = I$

for k = 1:n

  L[k, k] = (A[k,k] - $\sum_{j=1}^{k-1}(L[k,j])^2)^{\frac{1}{2}}$

  for i = k+1:n

    L[i,k] $= \frac{1}{L[k,k]}(A[i,k] - \sum_{j=1}^{k-1} L[i,j] \times L[k,j])$

  end

end

Homework: Python implementation