CS 371 Notes

Thomas Liu

April 13, 2024

Contents

1 Chapter 1: Floating Point Arithmetic

Numerical algorithm and computers are operating on finite precision arithmetic
We don't have the totality of $\mathbb{R}$ ar our disposal $\rightarrow$ only a tiny position of it

1.0.1 Definition: Approximation

Let $\hat{x}$ be an approximation to a real number x. Two fundamental measures of $\hat{x}$:

- absolute error: $\triangle x = x - \hat{x}$
- relative error: $\delta x = \dfrac{x - \hat{x}}{x}$

1.1 Section 1.1 Floating Point number System

1.1.1 Defintion: Floating Point System

A floating point system $F \subset \mathbb{R}$ is subset of real numbers of following form:

$$Z = \pm(0.x_1 x_2 \cdots x_m)_b \times b^{\pm(y_1 y_2 \cdots y_e)_b}$$

where $0 \leq x_i \leq b - 1$, $0 \leq y_i \leq b - 1$, $\forall 1 \leq i \leq m$, $1 \leq j \leq e$
three parameters of F

- base: b_f
- mantissa: m_f
- exponent: e_f

$$F[b = b_f, m = m_f, e = e_f]$$

1.1.2 Definition: Normalized

A floating point number in $F \subset \mathbb{R}$

$$Z = \pm(0.x_1 x_2 \cdots x_m)_b \times b^{\pm(y_1 y_2 \cdots y_e)_b}$$

is normalized when $x_1 \geq 1$

1.1.3 General Formula

$$\begin{aligned} &(a_n a_{n-1} \cdots a_1 a_0 a_{-1} \cdots a_m)_b \\ =&a_n b^n + a_{n-1} b^{n-1} + \cdots a_1 b^1 + a_0 b^0 + a_{-1} b^{-1} + \cdots + a_{-m} b^{-m} \end{aligned}$$

1.1.4 Definition: Machine Epsilon

The distance from 1.0 to the next largest (normalized) floating point number is called machine epsilon, denoted by $\epsilon\ mach$

$$1 = (0.10 \cdots 0)_b \times b^{(0 \cdots 01)_b}$$

$$next = (0.10 \cdots 01)_b \times b^{(0 \cdots 01)_b}$$

$$\epsilon\ mach = (0.0 \cdots 01)_b \times b^{(0 \cdots 01)_b} = b^{-m} \times b = b^{1-m}$$

- number m is also called precision
- $\epsilon\ mach$ is also called machine precision
- $\epsilon\ mach = b^{1-m}$
- important: formula $\epsilon\ mach = b^{1-m}$ is subject to slight change in practical (single / double formats)

1.1.5 Definition: Subnormal Numbers

The system F can be extended by including (filling the gap) subnormal numbers which are represented by:

$$\pm(0.0x_2 \cdots x_m)_b \times b^{-(b_{-1}, b_{-1}, \cdots, b_{-1})_b}$$

where $0 \leq x_2, x_3 \cdots, x_m \leq b-1$, and $(0.0x_2 \cdots x_m)_b \neq 0$

- closest to zero normalized numbers: $\pm(0.10 \cdots 0)_b \times b^{-(b_{-1}, b_{-1}, \cdots, b_{-1})_b}$
- subnormal numbers are closer to 0 than normalized numbers
- if we denote the smallest nonnormalized positive number as λ, then subnormal numbers fill the gap between 0 and λ with the same spacing between λ and $b\lambda$

1.2 Rounding, Overflow, Underflow

1.2.1 Definition: rounding

Let $G \subset \mathbb{R}$ denote all read numbers of the form

$$z = \pm(0.x_1 \cdots x_m)_b \times b^y \qquad y \in \mathbb{Z}$$

For $\forall x \in \mathbb{R}$, then $fl(x)$ denotes an element of G nearest to x, and the transformation $x \to fl(x)$ is called rounding

1.2.2 Definition: Overflow, Underflow

We say $fl(x)$ overflows if $|fl(x)| > \max\{|z| : z \in F\}$ and $fl(x)$ underflow if $0 < |fl(x)| < \min\{|z| : 0 \neq z \in F\}$

1.2.3 Theorem: Unit Roundoff

Every real number x lying in the range (such that $fl(x)$ is normalized in F) of F can be rounded to an element in F with a relative error no larger than $u = \frac{1}{2}\epsilon\ mach$
Mathematically, if $x \in \mathbb{R}$ lies in the range of F, then

$$fl(x) = x(1+\delta), |\delta| < u = \frac{1}{2}\epsilon\ mach$$

1.3 Standard Floating Point Systems

- single precision format (32-bit (4-bytes) memory)

$$\text{s} \mid m = 23 \text{ bits} \mid e = 8 \text{ bits}$$

  where s is sign bit of mantissa, $s = 1$ means negative, $s = 0$ means positive

- we have $2^8 = 256$ exponents from 0 to 255 $\rightarrow [0, 255]$

- want a range of signal exponents

- convertion:
  they are subtracted by a bias 127 $\rightarrow [-127, 128]$

- $e = (0 \cdots 0)_2 \rightarrow -127$ & $e = (1 \cdots 1) \rightarrow 128$, special numbers (non-normalized)

exponent	mantissa= 0	mantissa$\neq$ 0
$(00000000)_2$	$\pm$zero	subnormal numbers
$(11111111)_2$	$\pm$infinity	NaN (not a number)

- when $e \in [-126, 127]$, when string normalized values, instead of wasting a bit on storing the leading $x_1 - 1$, this is assumed

- a general formula

$$x_0|x_1 \cdots x_{23}|y_1 \cdots y_8 \rightarrow (-1)^{x_0} \times (1 \cdot x_1 x_2 \cdots x_{23})_2 \times 2^{(y_1 \cdots y_8)_2 - 127}$$

1.3.1 Double Precision Format (64-bit (8-bytes) memory)

we omit details about double precision except:

1. it's matlab default

2. 10^{-16} is a special number for (modern) numerical analyst since $eps(\text{"}double\text{"}) \approx 2.2204 \times 10^{-16}$. When we have an error of $\approx 10^{-16}$ from our algorithm we are happy

1.4 Floating Point Operations

1.4.1 Definition

Floating point addition $\oplus$ is defined by

$$\forall x, y \in \mathbb{R} : x \oplus y = fl(fl(x) + fl(y))$$

subtraction $\ominus$, multiplication $\otimes$ & division $\oslash$ can be similarly defined

1.4.2 Proposition

For any floating point number systems F, $x \oplus y = (fl(x) + fl(y))(1 + s)$ with $|s| \leq u$, the unit roundoff. same applies to $\ominus, \otimes, \oslash$

1.5 Condition of a Mathematical Problem

Consider a problem P with input $\vec{x}$ and output (exact) $\vec{z} = f_p(\vec{x})$

1.5.1 Definition: Conditioing of P

- P is said to be well-conditioned w.r.t. the absolute error if small change $\triangle\vec{x}$ in $\vec{x}$ results in small changes $\triangle\vec{z}$ in $\vec{z}$
- P is said to be ill-conditioned w.r.t. the absolute error if small change $\triangle\vec{x}$ in $\vec{x}$ results in large changes $\triangle\vec{z}$ in $\vec{z}$
- similarly define with w.r.t. the relative error

1.5.2 Vector Norms

Suppose V is a vector space over $\mathbb{R}$. Then $|\cdot|$ is a vector norm on V iff $||\vec{v}|| \geq 0$, and

- $||\vec{v}|| = 0$ iff $\vec{v} = \vec{0}$
- $||\lambda\vec{v}|| = |\lambda|||\vec{v}||$ $\forall\vec{v} \in V$, $\forall\lambda \in \mathbb{R}$
- $||\vec{u} + \vec{v}|| \leq ||\vec{u}|| + ||\vec{v}||$ $\forall\vec{u}, \vec{v} \in V$ (triangle inequality)

1.5.3 Condition Number of a Problem

- The condition number of a problem P w.r.t. the absolute error is given by the absolute condition number
$$\kappa_A = \frac{||\triangle\vec{z}||}{||\triangle\vec{x}||}$$
- The condition number of a problem P w.r.t. the relative error is given by the relative condition number
$$\kappa_R = \frac{||\triangle\vec{z}||/||\vec{z}||}{||\triangle\vec{x}||/||\vec{x}||}$$

2 Chapter 2: Road Finding

2.1

2.1.1 Intermediate Value Theorem

if $f(x)$ is continuous on a closed interval $[a, b]$ and $c \in [f(a), f(b)]$, then $\exists x^* \in [a, b]$ such that $f(x^*) = c$

2.1.2 Corollary

If $f(a) \cdot f(b) < 0$ for a closed interval $[a, b]$, then $[a, b]$ will contain at least one root x^* as long as $f(x)$ is continuous

2.2 Form Algorithm for Root Finding

2.2.1 Bisection method.

Theorem 2.3 If $f(x)$ is continuous on $[a_0, b_0]$ such that $f(a_0) \cdot f(b_0) \le 0$ then the interval $[a_k, b_k]$, defined by

$$[a_k, b_k] := \begin{cases} \left[a_{k-1}, \frac{a_{k-1}+b_{k-1}}{2}\right] & \text{if } f(a_{k-1}) \cdot f\left(\frac{a_{k-1}+b_{k-1}}{2}\right) \le 0 \\ \left[\frac{a_{k-1}+b_{k-1}}{2}, b_{k-1}\right] & \text{if } f(a_{k-1}) \cdot f\left(\frac{a_{k-1}+b_{k-1}}{2}\right) > 0 \end{cases}$$

Then, it holds that $f(a_k) \cdot f(b_k) \le 0$ for any $k \ge 1$.

a_{k-1} $\qquad \frac{a_{k-1}+b_{k-1}}{2}$ $\qquad b_{k-1}$

$f\left(\frac{a_{k-1}+b_{k-1}}{2}\right) f(a_{k-1}) \le 0 \Rightarrow a_k \qquad b_k$

$f\left(\frac{a_{k-1}+b_{k-1}}{2}\right) f(a_{k-1}) > 0 \Rightarrow \qquad a_k \qquad b_k.$

Algorithm 2.1 (Bisection method)

Input: $f(x)$, $[a,b]$, tolerance τ (under the condition that $f(a)\cdot f(b) \le 0$)

Output: x, an approximant of x^* $\quad (f(x^*)=0)$

```
while |b-a| > τ      OR: |f((a+b)/2)| > τ
      c = (a+b)/2
      if f(a) * f(c) ≤ 0
           b = c
      else
           a = c
      end if
end while
x = (a+b)/2.
```

2.2.2. Fixed point iteration

<u>Definition 2.1</u> We say that x^* is a fixed point of $g(x)$ if $g(x^*) = x^*$

<u>Problem</u> Root finding of $f(x)$ is equivalent to finding the fixed point of

$$g(x) \triangleq x - f(x)$$

$f(x^*)=0 \Rightarrow x^* - f(x^*) = x^*$

$g(x^*)=x^* \Rightarrow x^* - f(x^*) = x^* \Rightarrow f(x^*) = 0$

* The fixed point iteration goes like this:

$$x_{n+1} = g(x_n) \quad n = 0, 1, 2, 3 \ldots$$

Algorithm 2.2. (Fixed point iteration)

Input: $g(x)$, x_0, tolerance τ

Output: x, an approximant of x^*

$i = 0$

repeat

  $i = i + 1$

  $x[i] = g(x[i-1])$

until $|x[i] - x[i-1]| < \tau$

$x = x[i]$

Convergence? If $|g'(x^*)| < 1$ and x_0 is "close enough" to x^*, alg. 2.2 will converge to x^*

more on this in §2.4

2.2.3. Newton's method

Consider the Taylor series of $f(x^*)$ about an initial estimate x_0:

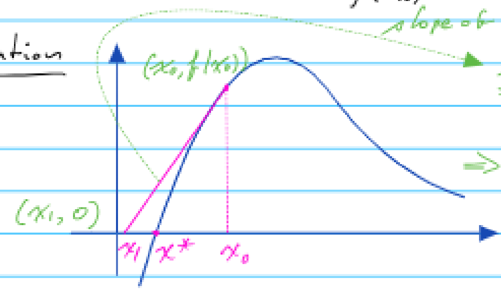
$$0 = f(x^*) = f(x_0) + f'(x_0)(x^* - x_0) + \mathcal{O}((x^* - x_0)^2)$$

$$\Rightarrow \quad 0 = f(x^*) \approx f(x_0) + f'(x_0)(x^* - x_0)$$

$$\Rightarrow \quad 0 = f(x_0) + f'(x_0)(x_1 - x_0)$$

i.e. $$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

<u>Visualization</u>



slope of tangent line

$$\frac{f(x_0) - 0}{x_0 - x_1} = f'(x_0)$$

$$\Rightarrow \quad x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Thus, we have iteration:

$$0 = f(x_{k-1}) + f'(x_{k-1})(x_k - x_{k-1})$$

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

```
Algorithm 2.3 (Newton's Method)
    Input: f(x), f'(x), x0, tolerance τ
    Output: x, an approximant for x* (f(x*)=0)
    i = 0
    x[0] = x0
    repeat
        i = i+1
        if f'(x[i-1]) = 0 stop
        x[i] = x[i-1] - f(x[i-1]) / f'(x[i-1])
    until |x[i] - x[i-1]| < τ OR |f(x[i])| < τ
    x = x[i]
```

2.3 Intro of Convergence Analysis

2.3.1 Error at Iteration

For a sequence $\{x_i\}_{i=0}^{\infty}$ and point x^*, the error at iteration i is

$$e_i = x_i - x^*$$

2.3.2 Order of Convergence

The sequence $\{x_i\}_{i=0}^{\infty}$ converges to x^* with order of convergence q iff

1. $\{x_i\}_{i=0}^{\infty}$ converges to x^*
2. $|e_{i+1}| = c_i|e_i|^q$ where $\lim_{i\to\infty} c_i = c^*$ for some constant $c^* \in (0,\infty)$
3. (special case): when $q = 1$, we call the convergence linear and we require $c^* < 1$, which is also called the rate of convergence

Method	Guaranteed Convergence	Order	Knowledge of $f'(x)$
Bisection	yes	linear	nope
Fixed-point	not always, depend on $g(x)$ and x_0	linear	nope
Newton	not always, depend on $f(x)$ and x_0	quadratic	yes
Secant	not always, depend on $f(x), x_0$ and x_1	$\frac{1+\sqrt{5}}{2}$	nope

2.4 Convergence Theory of the Root Finding Algorithm

2.4.1 Bisection Method

Consider the sequence $\{L_i\}_{i=1}^{\infty}$ with $L_i = |b_i - a_i|$ and $x_i = \dfrac{a_i + b_i}{2}$

- $L_{i+1} = \frac{1}{2} L_i$
- $|e_n| \leq L_n \leq (\dfrac{1}{2})^n L_0 = (\dfrac{1}{2})^n (b_0 - a_0)$

2.4.2 Fixed Point Iteration

Suppose g is continuous on $[a, b]$. Then g is said to be a contraction on $[a, b]$ if there exists a constant $L \in (0, 1)$ such that

$$|g(x) - g(y)| \leq L|x - y| \quad \forall x, y \in [a, b]$$

Proposition

If $g(x)$ is differentiable on $[a, b]$ with $|g'(x)| < 1$ $\forall x \in [a, b]$. Then $g(x)$ is a contraction on $[a, b]$ with

$$L = \max_{x \in [a,b]} |g'(x)|$$

Definition: Mean Value Theorem

If $g(x)$ is continuous on $[a, b]$ and differentiable on (a, b), then $\exists c \in (a, b)$ such that the tangent at c is parallel to the secant line connecting $(a, g(a))$ and $(b, g(b))$

$$g'(c) = \frac{g(b) - g(a)}{b - a}$$

Theorem: Cantraction Mapping Theorem

Let g be continuous on $[a, b]$ and assume that

- $g(x) \in [a, b]$
- $g(x)$ is contraction on $[a, b]$

Then

- g has a unique fixed point x^* in the interval $[a, b]$
- the sequence $\{x_k\}$ defined by

  $$x_{k+1} = g(x_k)$$

  converges to x^* as $k \to \infty$ for any starting point x_0 in $[a, b]$

Corollary: Convergence of Fixed Point Iteration

Let $g'(x)$ be continuous on $[a,b]$ and assume that

- $g(x) \in [a,b]$
- $\max\limits_{x\in[a,b]} |g'(x)| < 1$

Then

1. g has a unique fixed point x^* in the interval $[a,b]$
2. the sequence $\{x_k\}_{k=0}^{\infty}$ defined by
$$x_{k+1} = g(x_k)$$
converges to x^* as $k \to \infty$ for any starting point x_0 in $[a,b]$
3. the sequence $\{x_k\}_{k=0}^{\infty}$ converges with
$$|e_{k+1}| = c_k|c_k|$$
and
$$\lim_{k\to\infty} c_k = |g'(x^*)|$$

If $|g'(x^*)| \in (0,1)$, we have linear convergence
If $g'(x^*) = 0$, we have faster convergence

Divergence of Fixed Point Iteration

Let $g'(x)$ be continuous on $[a,b]$ and $g(x)$ has a unique fixed point x^* on $[a,b]$
If $|g'(x^*)| > 1$, then the sequence $\{x_k\}_{k=0}^{\infty}$ diverges for any starting point x_0

2.4.3 Newton's Method

Convergence of Newton's Method

If $f(x^*) = 0$, $f'(x^*) \neq 0$ and f, f', f'' are all continuous in $I_\delta = [x^* - \delta, x^* + \delta]$ with x_0 sufficiently close to x^* then the sequence $\{x_k\}_{k=0}^{\infty}$ converges quadratically to x^* with

$$|e_{k+1}| = c_k|e_k|^2$$

where $\lim\limits_{k\to\infty} c_k = \dfrac{|f''(x^*)|}{|2f'(x)|}$

2.4.4 Secant Method

Convergence of Secant Method

If $f(x^*) = 0$, $f'(x^*) \neq 0$ and f, f', f'' are all continuous in $I_\delta = [x^* - \delta, x^* + \delta]$ with x_0, x_1 sufficiently close to x^* then the sequence $\{x_k\}_{k=0}^{\infty}$ converges to x^* with order of convergence $q = \dfrac{1}{2}(1+\sqrt{5}) \approx 1.62$

$$|e_{k+1}| = c_k|e_k|^{\frac{1+\sqrt{5}}{2}} \,\&\, \lim_{k\to\infty} c_k = c^* > 0$$

3 Numerical Linear Algebra

3.1 Introduction

3.1.1 Determinant

The determinant of matrix $A \in \mathbb{R}^{n\times n}$ is given by

$$\det(A) = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} \det(A_{ij})$$

- i can be any number from 1 to n
- A_{ij} is the $(n-1)\times(n-1)$ matrix obtained by removing row i and column j from the original matrix A

3.1.2 Theorem: Existence and Uniqueness of Solution of $A\vec{x} = \vec{b}$

Case 1: $\det(A) \neq 0$, $\vec{x} = A^{-1}\vec{b}$ is the unique solution of $A\vec{x} = \vec{b}$
Case 2: $\det(A) = 0$

- if $\vec{b} \in \text{Range}(A)$ then $A\vec{x} = \vec{b}$ has infinite many solutions
- if $\vec{b} \notin \text{Range}(A)$ then $A\vec{x} = \vec{b}$ has no solutions

3.2 Guassian Elimination

3.2.1 LU Factorization

Gaussian Elimination

- Phase 1: reduce the matrix A to upper triangle form
- Phase 2: solve the reduced system

Definition

A matrix $A \in \mathbb{R}^{n\times n}$ with components a_{ij} is said to be

- upper-triangular: if $a_{ij} = 0$ for all $i > j$
- lower-triangular: if $a_{ij} = 0$ for all $i < j$

A is said to be triangular if it is eiter upper- or lower-triangular

Algorithm: Forward & Backward Substitutions

$A\vec{x}=\vec{b}$, $A\in\mathbb{R}^{n\times n}$, $\vec{b}\in\mathbb{R}^{n\times 1}$.

- A: upper triangular $\Rightarrow$ backward substitution.

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1,n-1} & a_{1,n} \\ 0 & a_{22} & \cdots & a_{2,n-1} & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & \cdots & 0 & a_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{pmatrix}$$

$$x_n = a_{n,n}^{-1} b_n$$
$$x_{n-1} = a_{n-1,n-1}^{-1} (b_{n-1} - a_{n-1,n} x_n)$$
$$x_{n-2} = a_{n-2,n-2}^{-1} (b_{n-2} - a_{n-2,n} x_n - a_{n-2,n-1} x_{n-1})$$
$$\vdots$$
$$x_i = a_{i,i}^{-1} \left(b_i - \sum_{k=i+1}^{n} a_{i,k} x_k \right)$$

- A: Lower triangular $\Longrightarrow$ Forward substitution.

$$\begin{pmatrix} a_{11} & 0 & \cdots & 0 & 0 \\ a_{21} & a_{22} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-1,1} & a_{n-1,2} & \cdots & a_{n-1,n-1} & 0 \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n-1} & a_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{pmatrix}$$

$$x_1 = a_{11}^{-1} b_1$$
$$x_2 = a_{22}^{-1} (b_2 - a_{21} x_1)$$
$$x_3 = a_{33}^{-1} (b_3 - a_{31} x_1 - a_{32} x_2)$$
$$\vdots$$
$$x_i = a_{i,i}^{-1} \left(b_i - \sum_{k=1}^{i-1} a_{i,k} x_k \right)$$

Inversion Property

$L_i = m_i^{-1}$ can be obtained from m_i by swapping the signs of the off-diagonal elements

Combination Property

$L = L_1 L_2 \cdots L_{n-1} = m_1^{-1} m_2^{-1} \cdots m_{m-1}^{-1}$
L can be obtained by placing all off-diagonal elements of L_i in the corresponding position in L

LU Factorization/Decomposition

For $A \in \mathbb{R}^{n\times n}$
LU factorization may be computed as follow

$$\begin{aligned}
A^{(1)} &= A \\
A^{(2)} &= m_1 A^{(1)} \\
A^{(3)} &= m_2 A^{(2)} = m_2 m_1 A^{(1)} \\
&\vdots \\
A^{(n)} &= m_{n-1} \cdots m_{(2)} m_{(1)} A^{(1)}
\end{aligned}$$

m_j is a matrix where the diagonal is 1, $c_{ij} = -\dfrac{a_{ij}}{a_{ji}}$ for $j + 1 \leq i \leq n$

3.2.2 Guassian Elimination (full version)

- phase 1: decompose $A = LU$, $LU\vec{x} = \vec{b}$
- phase 2: solve $L\vec{y} = \vec{b}$ for $\vec{y}$ by forward substitution
- phase 3: solve $U\vec{x} = \vec{y}$ for $\vec{x}$ by backward substitution

3.2.3 Pivoting

Definition

$P \in \mathbb{R}^{n\times n}$ is a permutation matrix iff P is obtained from the identity matrix by swapping any number of rows

Theorem

For all $A \in \mathbb{R}^{n\times n}$ there exists a permutation matrix P, a unit lower triangular matrix L and an upper triangular matrix U such that

$$PA = LU$$

Corollary

If A is nonsingular then $A\vec{x} = \vec{b}$ can be solved by LU factorization applied to PA

Algorithm and Computational Cost

```
Algorithm 3.3 Phase 1 : A = LU.
    L = diag(1)   : identity matrix.
    U = A
    for p = 1 : n-1
        for r = p+1 : n
            t = - U(r,p) / U(p,p)
            U(r,p) = 0
            for c = p+1 : n
                U(r,c) = U(r,c) + t U(p,c)
            end for
            L(r,p) = -t
        end for
    end for
```

Algorithm 3.3 Phase 2: $L\vec{y} = \vec{b}$ forward substitution

```
y = b
for r = 2 : n
    for c = 1 : r-1
        y(r) = y(r) - L(r,c) * y(c)
    end for
end for
```

Algorithm 3.3 Phase 3: $U\vec{x} = \vec{y}$ backward substitution

```
x = y
for r = n : -1 : 1
    for c = r+1 : n
        x(r) = x(r) - U(r,c) * x(c)
    end for
    x(r) = x(r) / U(r,r)
end for
```

3.3 Conditioning of $A\vec{x} = \vec{b}$ and stability of Gaussian Elimination

- condition of the mathematical problem $A\vec{x} = \vec{b}$

$$\vec{x} = f_p(A, \vec{b}) = A^{-1}\vec{b}$$

  - absolute condition number

$$\mathcal{K}_A = ||\triangle\vec{x}||/||\triangle(A, \vec{b})||$$

  - relative condition number

$$\mathcal{K}_R = \frac{||\triangle\vec{x}||}{||\vec{x}||}/\frac{||\triangle(A, \vec{b})||}{||(A, \vec{b})||}$$

3.3.1 Matrix Norm

Definition

The natural matrix p-norm:

$$||A||_p = \sup_{||vx||_p \neq 0} \frac{||A\vec{x}||_p}{||\vec{x}||_p}$$

Theorem

$||A||_p$ is a norm $\forall A \in \mathbb{R}^{m\times n}$

- $||A||_p \geq 0$, $||A||_p = 0$ iff $A = 0$
- $||\alpha A||_p = |\alpha|||A||_p$ $\forall \alpha \in \mathbb{R}$
- $||A + B||_p \leq ||A||_p + ||B||_p$ $\forall B \in \mathbb{R}^{m\times n}$

Proposition

- $||A\vec{x}||_p \leq ||A||_p||\vec{x}||_p$
- $||AB||_p \leq ||A||_p||B||_p$

Proposition

- $||A||_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{ij}|$

Singular Value Decomposition

* Given $A \in \mathbb{R}^{n\times n}$, a singular value decomposition (SVD) of A is a factorization:

$$A = U \Sigma V^T$$

where
- $U \in \mathbb{R}^{n\times n}$ is orthogonal $(UU^T = U^TU = I)$
- $V \in \mathbb{R}^{n\times n}$ is orthogonal
- $\Sigma \in \mathbb{R}^{n\times n}$ is diagonal

$$\Sigma = \begin{pmatrix} \sigma_1 & & 0 \\ & \sigma_2 & \\ 0 & & \ddots & \\ & & & \sigma_n \end{pmatrix}$$

- the diagonal entries σ_j of Σ are nonnegative and in nondecreasing order
$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$$
- the singular values $\{\sigma_j\}_{j=1}^n$ are uniquely determined.

3.3.2 Condition number of a matrix

The condition number of a matrix A ($\det(A) \neq 0$) is

$$\mathcal{K}(A) = ||A||||A^{-1}||$$

$$\mathcal{K}_p(A) = ||A||_p||A^{-1}||_p$$

3.3.3 Proposition

For $A \in \mathbb{R}^{n\times n}$, $\det(A) \neq 0$ and $p = 2$ for vector and matrix norms in question

$$\mathcal{K}_2(A) = ||A||_2||A^{-1}||_2 = \frac{\sqrt{1}(A)}{\sqrt{n}(A)}$$

3.3.4 Gaussian Elimination with Partial Pivoting

Essentially, during every step of the LU factorization, ie steps where we compare m_i, we rearrange the rwo such that we get the largest pivoting element (in absolute value)

3.4 Interative Methods for Solving $A\vec{x} = \vec{b}$

3.4.1 Definition

$A \in \mathbb{R}^{n\times n}$ is a square matrix iff the number of nonzero elements in A is "much smaller" than n^2, or equally, "most" of the elements of A are zero

3.4.2 Iterative Method

For solving $A\overrightarrow{x} = \overrightarrow{b}$, general stationary itertive method takes the form

$$\overrightarrow{x}^{(k+1)} = G\overrightarrow{x}^{(k)} + \overrightarrow{c}$$

where

- G is called iteration matrix
- if G does not depend on k, we have stationary iterative method

3.4.3 Definition

TL residual of a linear system $A\overrightarrow{x} = \overrightarrow{b}$ for some vector $\overrightarrow{u}$ is given by $\overrightarrow{r} = \overrightarrow{b} - A\overrightarrow{u}$

3.4.4 Definition: Stationary Iteration Based on Matrix Splitting

The iteration for solving $A\overrightarrow{x} = \overrightarrow{b}$ based on splitting $A = M + N$ is given by

$$G = -M^{-1}N$$

3.4.5 Jacobi Method

Algorithm 3.5 (Jacobi iterations)

initial guess $\vec{x}^{(0)}$

$k=0$; $\vec{r}^{(0)} = \vec{b} - A\vec{x}^{(0)}$

while $\|\vec{r}^{(k)}\|_2 / \|\vec{r}^{(0)}\|_2 > \tau_{rel}$ do

for $i = 1 \sim n$

$\sigma = 0$

for $j = 1 \sim n$

if $j \neq i$

$\sigma = \sigma + a_{ij} x_j^{(k)}$

end

end

$x_i^{(k+1)} = (b_i - \sigma) / a_{ii}$

end

$\vec{r}^{(k+1)} = \vec{b} - A\vec{x}^{(k+1)}$; $k = k+1$

end

3.4.6 Gauss-Seidel Method

Algorithm 3.6 (Guass-Seidel iterations)

initial guess $\vec{x}$; initial residual: $\vec{r}^{(0)} = \vec{b} - A\vec{x}$

$\vec{r} = \vec{r}^{(0)}$

while $\|\vec{r}\|_2 / \|\vec{r}^{(0)}\|_2 > \tau_{rel}$ do

  for $i = 1 \sim n$

    $\sigma = 0$

    for $j = 1 \sim n$

      if $j \neq i$

        $\sigma = \sigma + a_{ij} x_j$

      end

    end

    $x_i = (b_i - \sigma)/a_{ii}$

  end

  $\vec{r} = \vec{b} - A\vec{x}$

end

3.4.7 Successive Over-Relaxation (SOR)

- matrix spliting: $A = M + N$ where

$$M = \frac{1}{w}D + L$$

$$N = (1 - \frac{1}{w})D + U$$

w: relaxation factor ($w = 1$ is Gauss-Seidel) method

- iteration matrix

$$G = (\frac{1}{w}D + L)^{-1}((\frac{1}{w} - 1)D - U)$$

- iteration:

$$\vec{x}^{(k+1)} = (\frac{1}{w}D + L)^{-1}((\frac{1}{w} - 1)D - U)\vec{x}^{(k)} + \frac{1}{w}(\frac{1}{w}D + L)^{-1}\vec{b}$$

Algorithm 3.7 (SOR Iterations)

```
initial guess x⃗; initial residual: r⃗(0) = b⃗ - A x⃗
r⃗ = r⃗(0)
while ||r⃗||2 / ||r⃗(0)||2 > τrel do
    for i = 1 ~ n
        σ = 0
        for j = 1 ~ n
            if j ≠ i
                σ = σ + aij xj
            end
        end
        xi = (1 - ω) xi + ω (bi - σ) / aii
    end
    r⃗ = b⃗ - A x⃗
end
```

3.5 Convergence of Iterative Methods

3.5.1 Definition: Spectral Radius of Matrix

Let $\{\lambda_1, \cdots, \lambda_n\}$ be the set of eigenvalus of $A \in \mathbb{R}^{n\times n}$. The spectral radius of A is given by

$$\rho(A) = \max\{|\lambda_1|, \cdots, |\lambda_n|\}$$

3.5.2 Theorem: Convergence of Iterative Methods

Let iterative method given by $\vec{x}^{(k+1)} = G\vec{x}^{(k)} + \vec{c}$ with initial guess $\vec{x}^{(0)}$. Then it converges for all $\vec{c} \in \mathbb{R}^n$ iff

$$\rho(G) < 1$$

3.6 Definition

$A \in \mathbb{R}^{n\times n}$ is strictly diagonally dominant if if for all $i = 1, \cdots, n$,

$$|a_{ii}| > \sum_{j=1, j\neq i}^{m} |a_{ij}|$$

3.6.1 Proposition

A strictly diagonally dominant matrix A is nonsingular

3.7 Theorem

Consider $A\vec{x} = \vec{b}$ and starting vector $\vec{x}^{(0)}$. Let $\{x^{(i)}\}_{i=0}^{\infty}$ be sequence generated by either Jacobi, Gauss-Seidel method. If A is strictly diagonally dominant, the sequence converges to the unique solution of $A\vec{x} = \vec{b}$

3.7.1 Proposition

For any natural matrix norm, $||\cdot||$, and a square matrix $G \in \mathbb{R}^{n\times n}$

$$\rho(G) \leq ||G||$$

4 Interpolation

4.1 Polynomial Interpolation

4.1.1 Vandermonde Matrix

Definition

Given $n+1$ discrete data point $\{(x_i, f_i)\}_{i=0}^{n}$ with $x_i \neq x_j$, for $i \neq j$, the interpolating polynomial P_n is given by

$$P_n(x) = a_0 + a_1 x + \cdots + a_n x^n$$

st $P_n(x_i) = f_i$ for $0 \leq i \leq n$

Algorithm

solver $(n+1) \times (n+1)$ linear system

$$\begin{cases} a_0 + a_1 x_0 + \cdots + a_n x_0^n = f_0 \\ a_0 + a_1 x_1 + \cdots + a_n x_1^n = f_1 \\ \vdots \\ a_0 + a_1 x_n + \cdots + a_n x_n^n = f_n \end{cases}$$

can be written as $V\vec{a} = \vec{f}$

Proposition

The determinant of V is

$$det(V) = \prod_{0 \leq i < j \leq n} (x_j - x_i)$$

Theorem

Interpolating polynomial $P_n(x) \in P_n$ given $\{(x_i, f_i)\}_{i=0}^n$ exists and unique

4.1.2 Lagrange Form

Lagrange Form of Interpolating Polynomial

$n+1$ Lagrange polynomial for a set of points $\{(x_i, f_i)\}_{i=0}^n$ are P_n polynomial that satisfy

$$l_i(x_j) \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

or

$$l_i(x_j) = f_{ij} \text{(kronecker symbol)}$$

- we can write out $l_i(x)$

$$l_i(x) = \prod_{j=0, j \neq i}^{n} \frac{x - x_j}{x_i - x_j}$$

- Lagrange form of iterpolating polynomial

$$P_n(x) = \sum_{i=0}^{n} f_i l_i(x)$$

4.1.3 Hermite Interpolation

Definition

Given $\{(x_i, f_i, f'_i)\}_{i=0}^n$, Hermite interpolating polynomial $H_n(x)$ is the polynomial that satisfies

$$p(x) \in P_{2n+1}$$

st

$$p(x_i) = f_i \quad \text{n+1 condition}$$
$$p'(x_i) = f'_i \quad \text{n+1 condition} \rightarrow \text{2n+1 degree}$$

thus

$$H_n(x) = \sum_{i=0}^{n} f_i h_i(x) + \sum_{i=0}^{n} f'_i \tilde{h}_i(x)$$

where

$$h_i(x) = [1 - 2l'_i(x_i)(x - x_i)](l_i(x))^2$$
$$\tilde{h}_i(x) = (x - x_i)(l_i(x))^2$$

Proposition

For given $\{(x_i, f_i, f'_i)\}_{i=0}^n$ where $x_i \neq x_j$, there exists unique interpolating polynomial $p(x) \in P_{2n+1}$ st

$$\begin{cases} p(x_i) = f_i \\ p'(x_i) = f'_i \end{cases}$$

and $p(x) = H_n(x)$

4.2 Piecewise Polynomial Interpolation

4.2.1 Piecewise Linear Interpolation

Definition

Define a set of polynomial $p_1^{[i]}(x)$, $1 \leq j \leq n$ where domain of $p_1^{[i]}(x)$ is $I_i = [x_{i-1}, x_i]$

$$p_1^{[i]}(x) = \frac{x - x_2}{x_{i-1} - x_2} f_{i-1} + \frac{x - x_{i-1}}{x_i - x_{i-1}} f_i$$

The interpolating piecewise polynomial $P_1(x)$ is equal to $p_1^{[i]}(x)$ on $I_i = [x_{i-1}, x_i]$ for all

4.2.2 Spline Interpolation

Definition

Given $\{(x_i, f_i)\}_{i=0}^{n}$, $p(x)$ is a degree k spline if

- $p(x)$ is piecewise P_k polynomial on each interval $I_i = [x_{i-1}, x_i]$, denote $p^{[i]}(x)$ as restriction of $p(x)$ on I_i
- $p^{[i]}(x_{i-1}) = f_{i-1}$ and $p^{[i]}(x_i) = f_i$
- for each iterior node x_j
$$\begin{cases} p^{[j]'}(x_j) = p^{[j]'}(x_j) \\ p^{[j]''}(x_j) = p^{[j]''}(x_j) \\ \vdots \\ p^{[j]^{(k-1)}}(x_j) = p^{[j](k-1)}(x_j) \end{cases}$$
-
  - free boundary: $p^{[1]''}(x_0) = 0$, $p^{[n]''}(x_n) = 0$
  - clamped boundary: specify the 1^{st} derivatives at the end with f_0', f_n': $p^{[1]'}(x_0) = f_0'$, $p^{[n]'}(x_n) = f_n'$
  - periodic boundary: if $f_0 = f_n$, we impose that first and second derivatives also math end points
$$p^{[1]'}(x_0) = p^{[n]'}(x_n)$$
$$p^{[1]''}(x_0) = p^{[n]''}(x_n)$$

4.3 Error Analysis for Polynomial Interpolation

4.3.1 Newton's Form

Proposition

For all $f[x_0, x_1, \cdots, x_n]$

- $$f[x_0, x_1] = \frac{f_1 - f_0}{x_1 - x_0}$$
- $$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$
- $$f[x_0, x_1, \cdots, x_n] = \frac{f[x_1, x_2, \cdots, x_n] - f[x_0, x_1, \cdots, x_{n-1}]}{x_n - x_0}$$

Definition: Newton's Form

Given $\{(x_i, f_i)\}_{i=0}^{n}$ with distinct x_i, Newton's form at interpolating polynomial of degree n of $f(x)$ can be written

$P_n(x) = f_0 + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2] + \cdots + (x - x_0)\cdots(x - x_{n-1})f[x_0, \cdots, x_n]$

4.3.2 Error Estimate of Polynomial Interpolation

Notation

For $\{x_i\}_{i=0}^{n}$, $\tau\{x_0, \cdots, x_n\}$ denotes the closed interval

$$[\min\{x_0, \cdots, x_n\}, \max\{x_0, \cdots, x_n\}]$$

Proposition

Let f be given real valued function withh n cts derivatiives. For $\{(x_i, f_i)\}_{i=0}^{n}$, there exists $\delta \in \tau\{x_0, \cdots, x_n\}$ such that

$$f[x_0, \cdots, x_n] = \frac{f^{(n)}(\delta)}{n!}$$

Theorem

Given $\{(x_i, f_i)\}_{i=0}^{n}$ with distinct x_i's and f, a real valued function with $n+1$ cts derivatiives on the interval $\tau_t = \delta\{t, x_0, \cdots, x_n\}$ with t some given real number where we evaluate the error of interpolation

There exists $\delta \in \tau_t$

$$f(t) - P_n(t) = (t - x_0)\cdots(t - x_n)\frac{f^{(n+1)}(\delta)}{(n+1)!}$$

where $P_n(t)$ is the degree n interpolating polynomial of $f(t)$ on $\{(x_i, f_i)\}_{i=0}^{n}$

Corollary

Let $p(x)$ be piecewise linear interpolating polynomial of $f(x)$ on $I = [x_0, x_n]$

Then for any $1 \leq i \leq n$ and $x_{i-1} < t < x_i$

$$|f(t) - p(t)| \leq \frac{(x_i - xi - 1)^2}{\delta} max_{x_{i-1} \leq x \leq x_i} |f^{(2)}(x)|$$

5 Numerical Integration

5.1 Quadrature faced on Interpolating Polynomial

5.1.1 Midpoint Rule

Let $P_0(x)$ be the constant interpolating polynomial of $f(x)$ on $I = [a, b]$ at $\dfrac{a+b}{2}$

$$\hat{I}_M(f) = I(P_0(x)) = \int_a^b P_0(x)dx = \int_a^b f(\frac{a+b}{2})dx = (b-a)f(\frac{a+b}{2})$$

Midpoint Rule

To approximate $\displaystyle\int_a^b f(x)dx$, the midpoint rule reads

$$\hat{I}_M(f) := (b-a)f(\frac{a+b}{2}) \xrightarrow{approx} I(f) = \int_a^b f(x)dx$$

The error of the rule is

$$E_M(f) := I(f) - \hat{I}_M(f)$$

Error Estimate of Midpoint Rule

Let f be given real valued function with 2 cts derivatives. Then there exists $\delta \in (a, b)$ st

$$|E_M(f)| = |\frac{(b-a)^3}{24} f''(\delta)|$$

5.1.2 Trapezoidal Rule

Let $P_z(x)$ be linear interpolating polynomial of $f(x)$ on $[a, b]$ with interpolating data point $\{(a, f(a)), (b, f(b))\}$

$$\hat{I}_T(f) = I(P_1(x)) = \frac{b-a}{2}(f(a) + f(b))$$

Trapezoidal Rule

To approximate $\displaystyle\int_a^b f(x)dx$, the midpoint rule reads

$$\hat{I}_T(f) := \frac{b-a}{2}(f(a) + f(b)) \xrightarrow{approx} I(f) = \int_a^b f(x)dx$$

The error of Trapezoidal rule is

$$E_T(f) := I(f) - \hat{I}_T(f)$$

Error Estimate of Trapezoidal Rule

Let f be given real valued function with 2 cts derivatives. Then there exists $\delta \in (a,b)$ st

$$|E_T(f)| = |\frac{(b-a)^3}{12} f''(\delta)|$$

5.1.3 Simpson's Rule

Let $P_2(x)$ be the 2^{nd} degree interpolating polynomial of $f(x)$ on $[a,b]$ with interpolating data point $\{(a, f(a)), (\frac{a+b}{2}, f(\frac{a+b}{2})), (b, f(b))\}$

$$\hat{I}_S(f) = I(P_2(x)) = \frac{b-a}{6}(f(a) + 4f(\frac{a+b}{2}) + f(b))$$

Simpson's Rule

To numerically approximate $\int_a^b f(x)dx$, the Simpson's rule reads

$$\hat{I}_S(f) = \frac{b-a}{6}(f(a) + 4f(\frac{a+b}{2}) + f(b)) \xrightarrow{approx} I(f) = \int_a^b f(x)dx$$

The error of Simpson's rule is

$$E_S(f) := I(f) - \hat{I}_S(f)$$

Error Estimate of Simpson's Rule

Let f be given real valued function with 4 cts derivatives. Then there exists $\delta \in (a,b)$ st

$$|E_S(f)| = |\frac{(b-a)^5}{2880} f^{(4)}(f)|$$

5.1.4 Degree of Precision

Degree of Precision of Quadrature

The following statements are equivalent

- $\hat{I}(f)$ has degree of precision m
- $E(f) = I(f) - \hat{I}(f) \equiv 0$ for any $f(x) \in P_m$
- $\hat{I}(f)$ integrates any polynomial $f(x) \in P_m$ exactly

Midpoint rule and trapezoidal rule have degree of precision 1, Simpson's rule has degree of precision 3

5.2 Composite Quadrature

Composite Trapezoidal Rule

Local error estimate:

$$|E_{CT}^{i}(f)| = |\frac{(x_i - x_{i-1})^3}{12} f''(q_i)| \quad q_i \in (x_{i-1}, x_i)$$

Global error estimate:

$$E_{CT}(f) = \mathcal{O}(h^2)$$

Composite Simpson's Rule

Global error estimate:

$$E_{CS}(f) = \mathcal{O}(h^4)$$

Composite Midpoint Rule

Global error estimate:

$$E_{CM}(f) = \mathcal{O}(h^2)$$

5.3 Gaussian Integration

Lemma

Given any distinct set of nodes $x_1, \cdots, x_n$ in $[-1, 1]$, one can find unique set of weights $w_1, \cdots, w_n$ such that the quadrature is at least degree of precision $n-1$

5.3.1 Orthogonal Polynomials

Definition

Consider space of all polynomial on $[-1, 1]$ denoted by P
Define inner product

$$< f, g >= \int_{-1}^{1} f(x)g(x)dx$$

Lemma

All n roots of orthogonal polynomial $p_n(x)$ reside in $(-1, 1)$ and all simple

Gaussian Legendre Quadrature

Let $\{x_i\}_{i=1}^n$ be roots of $p_n(x)$, and let $\{w_i\}_{i=1}^n$ be solution of system
Then

- $\{w_i\}_{i=1}^n$ ais of degree of precision $2n-1$
- no quadrature exceeds this order

6 Discrete Fourier Methods

6.1 Introduction

Complex Numbers and Complex Plane

Complex number $z = a + bi$ is defined as point in the xy-plane having Cartesian coordinates (a, b)
The plane is denoted $\mathbb{C}$ and called complex plane

Addition and Multiplication of Complex Number

- Addition: $z_1 + z_2$ is giving by

$$z_1 + z_2 = (a_1 + a_2) + i(b_1 + b_2)$$

- Multiplication is given by

$$z_1 \cdot z_2 = (a_1 + b_1 i)(a_2 + b_2 i) = (a_1 a_2 - b_1 b_2) + (a_1 b_2 + a_2 b_1)i$$

Euler's Formula

$$e^{i\theta} = \cos\theta + i \sin\theta$$

6.2 Fourier Series

Goal: Given a function on $[a, b]$, expand it in sum of sines and cosines

6.2.1 Fourier Series and Orthogonal Basis

Orthogonal Basis & Orthonormal Basis

A basis $B = \{v_1, \cdots, v_n\}$ is orthogonal basis iff

$$< v_i, v_j >= c_{ij}$$

where c_{ij} is nonzero iff $i = j$
When $c_{ij} = 1$, call B an orthonormal basis

Square Integrable Function

Define vector space of square integrable functions on $[-\pi, \pi]$ as

$$V = L^2([-\pi, \pi]) = \{f(x) | \int_{-\pi}^{\pi} f(x)^2 dx < \infty\}$$

The inner product $< \cdot, \cdot >: V \times V \to \mathbb{R}$ is defined as

$$< f, g >= \int_{-\pi}^{\pi} f(x)g(x)dx$$

Basis of Vector Space of Infinite Dimension

Say $\{v_i\}_{i=1}^{\infty}$ is a basis of infinite dimensional vector V if $\forall v \in V$, there exists unique sequence $\{c_i\}_{i=1}^{\infty}$ such that

$$v = \sum_{i=1}^{\infty} c_i v_i$$

Proposition

Let $V = \{f(x) | \int_{-\pi}^{\pi} f(x)^2 dx < \infty\}$ and $B = \{1, \cos(kx), \sin(kx)\}_{k=1}^{\infty}$. Then B is orthogonal basis of V

Fundamental Convergence Theorem for Fourier Series

Let $V = \{f(x) | \int_{-\pi}^{\pi} f(x)^2 dx < \infty\}$
Then for $\forall f(x) \in V$, there exists a unique set of coefficients $\{a_0, a_k b_k\}_{k=1}^{\infty}$ such that

$$f_n(x) = \frac{a_0}{2} = \sum_{k=1}^{n} [a_k \cos(kx) + b_k \sin(kx)]$$

converges to $f(x)$ as $n \to \infty$

Definition

The Fourier series of $f(x)$ on $[-\pi, \pi]$ is given by $g(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} [a_k \cos(kx) + b_k \cos(kx)]$ with

$$\begin{cases} a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(kx) dx & (k \geq 0) \\ b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(kx) dx & (k \geq 1) \end{cases}$$

6.2.2 Complex Form of the Fourier Series

Square Integrable Complex-valued Functions

Define vector space of square integrable functions on $[-\pi, \pi]$ as

$$V = L_2([-\pi, \pi]) = \{f(x) | \int_{-\pi}^{\pi} |f(x)|^2 dx < \infty\}$$

where $|f(x)|$ is modules of $f(x) \in \mathbb{C}$
Inner product $< \cdot, \cdot >: V \times V \to \mathbb{C}$ is defined as

$$< f, g >= \int_{-\pi}^{\pi} f(x) \overline{g(x)} dx$$

Proposition

Let $V = \{f(x) | \int_{-\pi}^{\pi} |f(x)|^2 dx < \infty\}$ and $B = \{\exp(ikx)\}_{k=-\infty}^{\infty}$
Then B is orthogonal basis of V

Definition

Fourier series of complex-valued $f(x)$ on $[-\pi, \pi]$ is given by

$$g(x) = \sum_{k=-\infty}^{\infty} c_k \exp(ikx)$$

with

$$c_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x)\overline{\exp(ikx)}dx = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) \exp(-ikx)dx$$

Proposition

The real and complex Fourier coefficients of a real function $f(x)$ obey:

- $\overline{c_k} = c_{-k}, a_{-k} = a_k, b_{-k} = -b_k$
- $a_k = 2Re(c_k), b_k = -2Im(c_k)$
- $b_0 = 0, c_0 = \frac{1}{2}a_0, c_k = \frac{1}{2}(a_k - ibk)$

Theorem

For a real function $f(x)$, its real and complex forms of Fourier series are equivalent: $g(x) = h(x)$

6.3 Discrete Fourier Transform

6.3.1 Approximation to Fourier Series Coefficients

N^{th} roots of unity

The N^{th} roots of unity are integer powers of $W_N = \exp(i\frac{2\pi}{N})$
They evenly divide unit circle on $\mathbb{C}$

DFT

The discrete Fourier transform of a discrete time signal $f[n]$ with $-\frac{N}{2} + 1 \leq n \leq \frac{N}{2}$ is

$$F[k] = DFT\{f[n]\} = \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f[n]W_N^{nk}$$

for $-\frac{N}{2} + 1 \leq k \leq \frac{N}{2}$

IDFT

The inverse discrete Fourier transform of a discrete frequency signal $F[k]$ with $-\frac{N}{2}+1 \leq k \leq \frac{N}{2}$ is

$$f[n = IDFT\{F[k]\}] = \sum_{k=-\frac{N}{2}-1}^{\frac{N}{2}} F[k]W_N^{nk}$$

where $-\frac{N}{2}+1 \leq n \leq \frac{N}{2}$

6.3.2 Properties of DFT

Properties of DFT

- $F[k] = F[k+sN]$ with $s \in \mathbb{Z}$
- if signal $f[n]$ is real
  - $Re(F[k])$ is even in k
  - $Im(F[k])$ is odd in k
  - $\overline{F[k]} = F[-k]$
  - $f[n]$ is even in n, then $Im(F[k]) = 0$
  - $f[n]$ is odd in n, then $Re(F[k]) = 0$

Dot Porduct in $\mathbb{C}$

$\forall \vec{x}, \vec{y} \in \mathbb{C}^n, < \vec{x}, \vec{y} >= \sum_{i=1}^{n} x_i \overline{y_i}$

Porposition

$\{\vec{F}_k\}_{k=0}^{N-1}$ forms orthogonal basis of $\mathbb{C}^n$

6.3.3 FFT