Introduction to Numerical Analysis I
Handout 12

1 Approximation of Functions

1.1 A Norm

Until now our approach in approximation of function was to find an interpolation. When we wanted to improve the error we used special interpolation points, that is roots of orthogonal polynomials.

We now learn another approach of approximation. Instead of interpolation condition $P_n(x_j) = f(x_j)$ we will attempt to find to minimize the error

$$||e(x)|| = ||P_n(x) - f(x)||$$

where the function $||\cdot||$ is defined as following

Definition 1.1. A norm over Vector Space V is a function $||\cdot|| : V \mapsto R$ that for each scalar $\lambda \in F$ and vector $v \in V$ satisfies the following conditions:

1. Positivity: $\|v\| \geq 0$, and also $\|v\| = 0$ iff $v = 0$
2. Homogeneity: $\|\lambda v\| = |\lambda| \|v\|$
3. Triangle Inequality: $\|v_1 + v_2\| \leq \|v_1\| + \|v_2\|$

Example 1.2.

1. For a vector in $v \in R^n$, including $\{v_j = f(x_j)\}_{j=0}^n$
   - L_1-norm $\|v\|_1 = \sum\limits_{i=1}^n |v_i|$
   - L_∞-norm $\|v\|_\infty = \max\limits_i |v_i|$
2. For real functions continuous in an interval $[a, b]$
   - L_1-norm $\|f\|_1 = \int\limits_a^b |f(x)|\, dx$
   - L_∞-norm $\|f\|_\infty = \max\limits_{a \leqslant x \leqslant b} |f(x)|$

Theorem 1.3 (Cauchy Schwartz inequality (C-S)).

$$|(x, y)|^2 \leqslant (x, x)(y, y)$$

Proof:

$$0 \leqslant \left(x - \frac{(x,y)}{(y,y)} y, x - \frac{(x,y)}{(y,y)} y\right) =$$

$$(x,x) - \frac{\overline{(x,y)}}{\overline{(y,y)}}(x,y) - \frac{(x,y)}{(y,y)}(y,x) + \left|\frac{(x,y)}{(y,y)}\right|^2 (y,y) =$$

$$= (x,x) - 2\frac{|(x,y)|^2}{(y,y)} + \frac{|(x,y)|^2}{(y,y)} = (x,x) - \frac{|(x,y)|^2}{(y,y)}$$

Theorem 1.4. Every Inner Product Induce a Norm $\|v\| = \sqrt{(v, v)}$

Proof: The conditions 1 and 2 implied by the definition of the inner product. To show that the 3rd condition is satisfied we use

$$|\mathrm{Re}\,(v_1, v_2)| \leqslant \underbrace{|(v_1, v_2)|}_{|\mathrm{Re}\,(v_1,v_2) + i\mathrm{Im}\,(v_1,v_2)|} \underset{\text{C-S}}{\leqslant} \|v_1\| \|v_2\|.$$

Thus

$$\|v_1 + v_2\| = (v_1 + v_2, v_1 + v_2) = (v_1, v_1) + (v_1, v_2) + \overline{(v_1, v_2)} +$$
$$(v_2, v_2) = (v_1, v_1) + 2\,\mathrm{Re}\,(v_1, v_2) + (v_2, v_2) \underset{\text{C - S}}{\leqslant}$$
$$\|v_1\| + 2\|v_1\|\|v_2\| + \|v_2\| = (\|v_1\| + \|v_2\|)^2$$

For the current discussion we interesting in two following L_2-norms

1. For a vector in $v \in R^n$, including $\{v_j = f(x_j)\}_{j=0}^n$, the inner product $(u, v) = \sum u_j v_j$ induces $\|v\|_2 = \sqrt{\sum\limits_{i=1}^n |v_i|^2}$
2. For real functions continuous in an interval $[a, b]$, the inner product $(f, g) = \int f\, \bar{g}\, dx$ induces $\|f\|_2 = \sqrt{\int\limits_a^b |f(x)|^2\, dx}$

1.2 Least Square Fit

Let $f(x)$ be real valued continuous function. We want to approximate it using function of the following form

$$g(x) = \sum_n c_n b_n(x),$$

where $b_n(x)$ is some basis. The error is given by $e(x) = f(x) - g(x) = f(x) - \sum\limits_n c_n b_n(x)$. We want to find the coefficients $c_1, \ldots, c_n$ such that $\|e(x)\|_2$ is minimal.

$$\|e\|_2^2 = g(c_1, ..., c_n) = \left(f - \sum_n c_n b_n, f - \sum_n c_n b_n\right) =$$

$$= (f, f) - \left(f, \sum_n c_n b_n\right) - \left(\sum_n c_n b_n, f\right) + \left(\sum_n c_n b_n, \sum_n c_n b_n\right) =$$

$$= \|f\| - 2\sum_n c_n (f, b_n) + \sum_n \sum_m c_n c_m (b_n, b_m)$$

In order to find the minimum we need to consider the derivatives, which gives

$$\frac{\partial g}{\partial c_j} = -2(f, b_j) + 2\sum_n c_n (b_n, b_j) = 0$$

Thus, we got for all j $(f, b_j) = \sum\limits_n c_n (b_n, b_j)$ One write it in a matrix form as

$$M(c_0, \cdots, c_n)^T = ((f, b_0), \cdots, (f, b_N))^T$$

where $M_{m,n} = (b_m, b_n)$.

We need to verify that the linear system is not singular, for which we will show that the homogeneous linear system $M\vec{c} = 0$ has only the trivial solution, that is $\vec{c} = \vec{0}$.

For the homogeneous system $M\vec{c} = 0$, we have

$$\sum_{n=1}^{N} c_n (b_n, b_j) = \left(\sum_{n=1}^{N} c_n b_n, b_j \right) = 0, \quad \forall j$$

that is, $\forall j$ the function $g(x) = \sum_{n=1}^{N} c_n b_n(x)$ is orthogonal to b_j. However, since $g \in Span\{b_n\}_{n=1}^N$, the orthogonality to all $\{b_n\}_{n=1}^N$ implies that $g = 0$. Since $\{b_n\}_{n=1}^N$ is linear independent we get $c_n = 0$, $\forall n$.

The minimality is due to the following, for any sequence ε_n:

$$\left\| f - \sum (c_n + \varepsilon_n) b_n \right\|^2 =$$

$$\left\| f - \sum c_n b_n + \sum c_n b_n - \sum (c_n + \varepsilon_n) b_n \right\|^2 =$$

$$\left\| f - \sum c_n b_n \right\|^2 + \left\| \sum c_n b_n - \sum (c_n + \varepsilon_n) b_n \right\|^2 +$$

$$2 \left(f - \sum c_n b_n, \sum c_n b_n - \sum (c_n + \varepsilon_n) b_n \right) \geqslant \left\| f - \sum_n c_n b_n \right\|^2$$

since

$$\left(f - \sum_n c_n b_n, \sum_n c_n b_n - \sum_n (c_n + \varepsilon_n) b_n \right) =$$

$$\left(\sum_n c_n b_n - f, \sum_n \varepsilon_n b_n \right) = \left(\sum_n c_n b_n, \sum_n \varepsilon_n b_n \right) - \left(f, \sum_n \varepsilon_n b_n \right)$$

$$= \sum_j \varepsilon_j \underbrace{\left\{ \sum_n c_n (b_n, b_j) - (f, b_j) \right\}}_{=0} = 0$$

Example 1.5. Given $(x_i, f(x_i)) = (1, 3.2), (2, 4.5), (3, 6.1)$ find a line that approximate the function. That is,

$$g = \alpha b_1 + \beta b_0 = \alpha \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \beta \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \qquad f = \begin{bmatrix} 3.2 \\ 4.5 \\ 6.1 \end{bmatrix}$$

Thus $\begin{bmatrix} 3 & 1+2+3 \\ 6 & 1+4+9 \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 3.2+4.5+6.1 \\ 3.2+9+18.3 \end{bmatrix}$ and therefore

$$a = \frac{13.8 \cdot 14 - 30.5 \cdot 6}{42 - 36} = 1.7 \quad b = \frac{30.5 \cdot 3 - 13.8 \cdot 6}{6} = 1.45$$

1.2.1 LSF Using Orthogonal Polynomials

The serious problem of the LSF method described above is that the matrix M is in general case is a full matrix, thus the numerical solution may be unstable.

Example 1.6. For example if we use inner product $(f, g) = \int_a^b f(x) g(x) dx$ with the standard polynomial basis $1, x, x^2, \ldots$. The entries of the matrix entries become $M_{ij} = (b_i, b_j) = \int_a^b x^{i+j} dx = \frac{x^{i+k+1}}{i+k+1}\Big|_a^b$ which give the Hilbert matrix denoted as $H_{n+1}(a, b)$, for example for $[a, b] = [0, 1]$ and $n = 4$ one get

$$H_5(0,1) = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{bmatrix}$$

The solution to the problem above is the use of orthogonal polynomial basis. In this case the matrix M become diagonal, so the coefficients are given $c_n = \frac{(f, b_n)}{(b_n, b_n)}$.

Example 1.7. $\{c_n\} = \left\{ \left(f, \frac{1}{\sqrt{2}}\right), \left(f, \sqrt{\frac{3}{2}} x\right), \left(f, \frac{1}{2}\sqrt{\frac{5}{2}}(3x^2 - 1)\right), \ldots \right\}$

1.2.2 Sensetivity to error

Consider $f(x)$ was measured with error $\|e(x)\| \leq \varepsilon$, that is $\tilde{f}(x) = f(x) + e(x)$. LSF have the following interesting property

$$\left\| f - \sum_n \left(\tilde{f}, b_n\right) b_n \right\| = \left\| f - \sum_n (f, b_n) b_n - \sum_n (e, b_n) b_n \right\| \leqslant$$

$$\leqslant \left\| f - \sum_n (f, b_n) b_n \right\| + \left\| \sum_n (e, b_n) b_n \right\|$$

$$= \left\| f - \sum_n (f, b_n) b_n \right\| + \Bigg\| \underbrace{LSF(e)}_{\approx e(x)} \Bigg\| \leqslant \left\| f - \sum_n (f, b_n) b_n \right\| + \varepsilon$$

that is $LSF(f) \approx LSF\left(\tilde{f}\right)$.

1.2.3 Discrete Fourier Transform as LSF

If we use a functional basis $b_n(x) = e^{i2\pi nx/L}$ the coefficients become

$$c_n = (f, b_n) = (f, e^{i2\pi nx/L}) = \frac{1}{L} \int_{-L/2}^{L/2} f(x) e^{-i2\pi nx/L} dx$$

become a Fourier Transform coefficients or a Discrete Fourier Transform coefficients:

$$c_n = (f, b_n) = \frac{1}{M} \sum_{m=-M/2}^{M/2} f(x_m) e^{-i2\pi nm/M} dx$$

This can be formulated with Vandermunde matrix of $\omega = e^{-2\pi i/N}$

$$c_n = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \ldots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \ldots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \ldots & \omega^{(N-1)^2} \end{pmatrix} \begin{pmatrix} x_0 \\ \vdots \\ x_n \end{pmatrix}$$

The DFT can be calculated very efficiently using an algorithm of Fast Furier Transform (FFT), here is how to use it in matlab:

```
xn = linspace(a,b,M);
cn = fft(f(xn));
cn = fftshift(cn);
cn = cn/M;
```

Use help fft and help fftshift to understand why.