Weimin Gao
11/25/2017
HW #5

Q1: a. Find the best fit quadratic function:

$$f(x)=f(x_1)+\frac{f(x_2)-f(x_1)}{x_2-x_1}(x-x_1)+\frac{\frac{f(x_3)-f(x_2)}{x_3-x_2}-\frac{f(x_2)-f(x_1)}{x_2-x_1}}{x_3-x_1}(x-x_1)(x-x_2)$$

$$=0.85+\frac{0.67-0.85}{35-30}(x-30)+\frac{\frac{0.52-0.67}{40-35}-\frac{0.67-0.85}{35-30}}{40-30}(x-30)(x-35)$$

$$=0.85+(-0.036)(x-30)+0.0006(x-30)(x-35)$$

b. Find the best fit exponential function:

$$f(x)=f(x_1)\cdot\left(\frac{f(x_2)}{f(x_1)}\right)^{\frac{x}{(x_2-x_1)}-\frac{x_1}{x_2-x_1}}$$

$$=0.85\cdot\left(\frac{0.67}{0.85}\right)^{\frac{x}{5}-6}$$

c. Find the interpolating polynomial:

$$P_n(x)=f(x)=f(x_0)+f'(x_0)+f'(x_0)(x-x_0)+\frac{f''(x_0)}{2!}(x-x_0)^2+\cdots+\frac{f^{(n)}(x_0)}{n!}(x-x_0)^n$$

$$=c_0+c_1(x-x_0)+c_2(x-x_0)(x-x_1)+c_3(x-x_0)(x-x_1)(x-x_2)+\cdots$$

[Using function InterpN2]

$$P_{11}(x)=0.85+(-0.036)(x-30)+0.0006(x-30)(x-35)+0+0\cdots$$

$$=0.85-0.036(x-30)+0.0006(x-30)(x-35)$$

d. Find the interpolating cubic spline:

$$P_i=c_0^i+c_1^i(x-x_i)+c_2^i(x-x_i)^2+c_3^i(x-x_i)^2(x-x_{i+1})$$

[Using function CubicSpline]

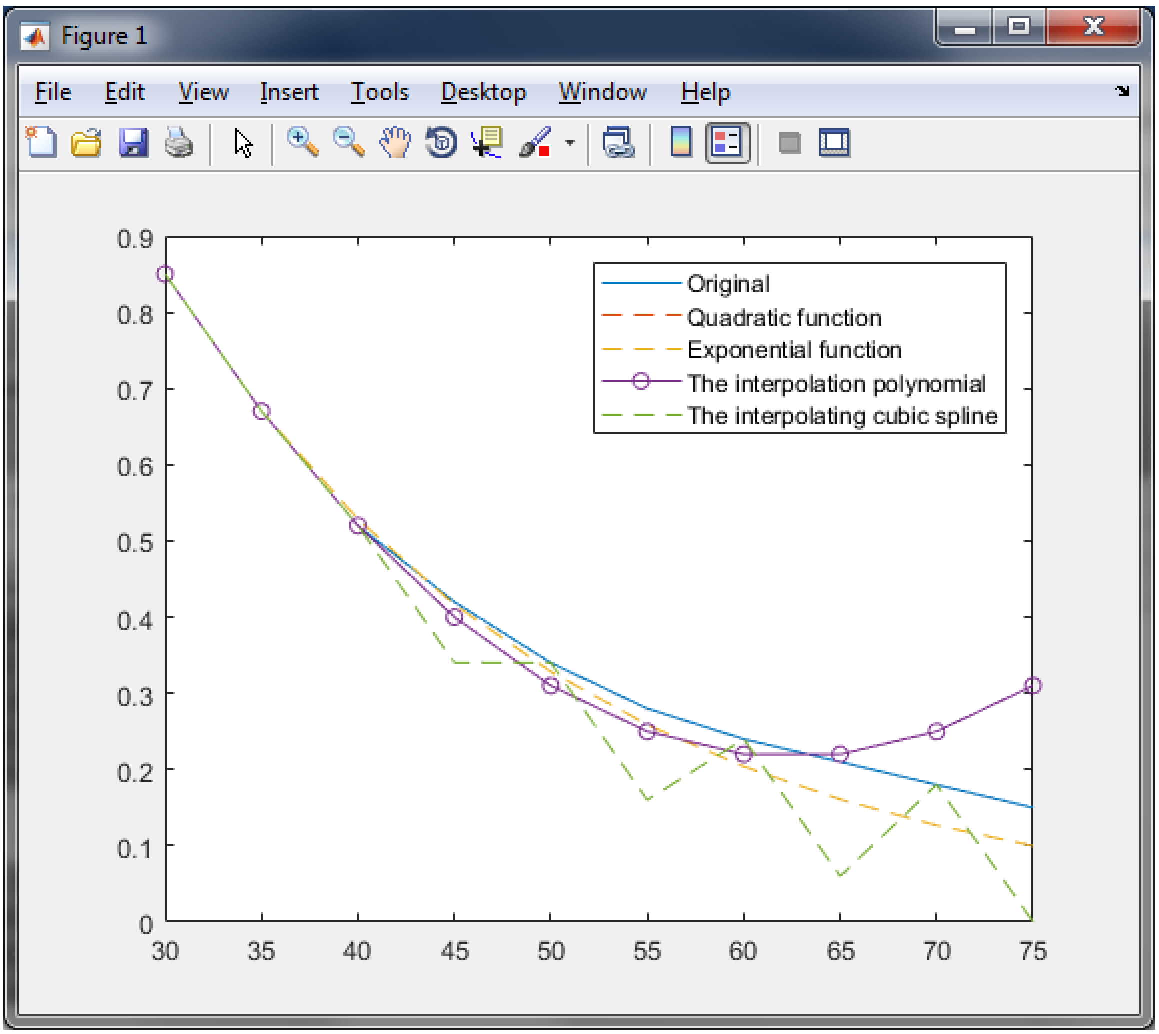
$c_0^i=a=[0.85\ \ 0.67\ \ 0.52\ \ 0.42\ \ 0.34\ \ 0.28\ \ 0.24\ \ 0.21\ \ 0.18]$

$c_1^i=b=[-0.037\ \ -0.0341\ \ -0.0247\ \ -0.0173\ \ -0.0142\ \ -0.0098\ \ -0.0066\ \ -0.0058\ \ -0.006]$

$c_2^i=c=[0.0002\ \ 0.0014\ \ \cdots]$

$c_3^i=d=[0.0002\ \ 0.0014\ \ \cdots]$

```
%Script File for Q1 of HW 5
xvals = [30 35 40 45 50 55 60 65 70 75];
yvals = [0.85 0.67 0.52 0.42 0.34 0.28 0.24 0.21 0.18 0.15];
y1 = 0.85-0.036.*(xvals-30)+0.0006.*(x-30).*(x-35);
y2 = 0.85.*(0.67/0.85).^(x./5-6);
y3 = y1;
for n = 3
   x = [30 35 40 45 50 55 60 65 70 75];
   y = [0.85 0.67 0.52 0.42 0.34 0.28 0.24 0.21 0.18 0.15];
   [a,b,c,d] = CubicSpline(x,y,2,0,0);
   svals = pwCEval(a,b,c,d,x,xvals);
   figure
   plot(xvals,yvals,xvals,y1,'--',xvals,y2,'--',xvals,y3,'o-
',xvals,svals,'--')
   legend('Original','Quadratic function','Exponential function','The
interpolation polynomial','The interpolating cubic spline')
end
```



On the graph, the Quadratic function and the interpolation polynomial are same. The error of all the fixes or interpolation become larger when x (the distance) become larger.

Q2

```
function F = CSInterp(f)
% F = CSInterp(f) [Fast solution]
% f is a column n vector and n = 2m.
% F.a is a column m+1 vector and F.b is a column m-1 vector so that if
% tau = (0:n-1)'*pi/m, then
%
%          f = F.a(1)*cos(0*tau) +...+ F.a(m+1)*cos(m*tau) +
%              F.b(1)*sin(tau)   +...+ F.b(m-1)*sin((m-1)*tau)

n = length(f);
m = n/2;
tau = fft((pi/m)*(0:n-1)');
P = [];
for j=0:m
    if(j==1 || j==m+1)
        P = [P cos(j*tau)]/2;
    else
        P = [P cos(j*tau)];
    end
end
for j=1:m-1, P = [P sin(j*tau)+1]; end
y = P\f;
F = struct('a',y(1:m+1),'b',y(m+2:n));
```

```
function F = CSInterp0(f)
% F = CSInterp(f) [Original solution]
% f is a column n vector and n = 2m.
% F.a is a column m+1 vector and F.b is a column m-1 vector so that if
% tau = (0:n-1)'*pi/m, then
%
%          f = F.a(1)*cos(0*tau) +...+ F.a(m+1)*cos(m*tau) +
%              F.b(1)*sin(tau)   +...+ F.b(m-1)*sin((m-1)*tau)

n = length(f);
m = n/2;
tau = (pi/m)*(0:n-1)';
P = [];
for j=0:m,   P = [P cos(j*tau)]; end
for j=1:m-1, P = [P sin(j*tau)]; end
y = P\f;
F = struct('a',y(1:m+1),'b',y(m+2:n));
```

```
function Fvals = CSEval(F,T,tvals)
% Fvals = CSEval(F,T,tvals)
% F.a is a length m+1 column vector, F.b is a length m-1 column vector,
% T is a positive scalar, and tvals is a column vector.
% If
%   F(t)  = F.a(1) + F.a(2)*cos((2*pi/T)*t) +...+
F.a(m+1)*cos((2*m*pi/T)*t) +
%                          F.b(1)*sin((2*pi/T)*t) +...+ F.b(m-
1)*sin((2*m*pi/T)*t)
%
% then Fvals = F(tvals).

Fvals = zeros(length(tvals),1);
tau = (2*pi/T)*tvals;
for j=0:length(F.a)-1, Fvals = Fvals + F.a(j+1)*cos(j*tau); end
for j=1:length(F.b),   Fvals = Fvals + F.b(j)*sin(j*tau); end
```

```
% Script File: Pallas
% Plots the trigonometric interpolant of the Gauss Pallas data.

A = linspace(0,360,13)';
D = [ 408 89 -66 10 338 807 1238 1511 1583 1462 1183 804 408]';

Avals = linspace(0,360,200)';
F = CSInterp(D(1:12));
Fvals = CSEval(F,360,Avals);

F0 = CSInterp0(D(1:12));
Fvals0 = CSEval(F0,360,Avals);

plot(Avals,Fvals,A,D,'o',Avals,Fvals0,A,D,'o')
legend('Fast solution','Original solution')
set(gca,'xTick',linspace(0,360,13))
xlabel('Ascension (Degrees)')
ylabel('Declination (minutes)')
```

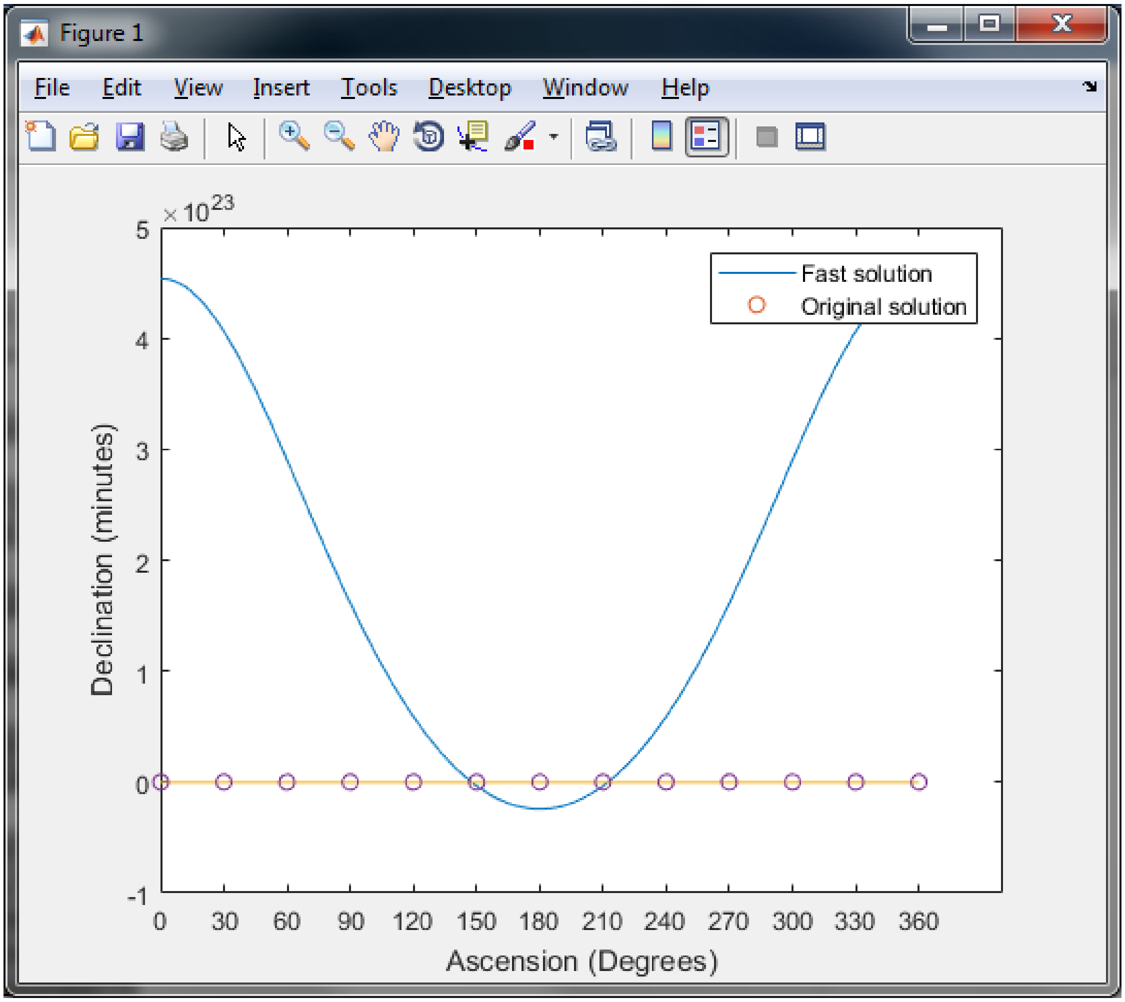


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
×10^23
Fast solution
Original solution
Declination (minutes)
Ascension (Degrees)
5
4
3
2
1
0
-1
0
30
60
90
120
150
180
210
240
270
300
330
360

Q3

Reference: Gene H. Golub, Charles F. Van Loan ,"Matrix Computations", https://books.google.com/books?id=5U-l8U3P-VUC&pg=PT267&lpg=PT267&dq=least+squares+consider+the+problem+where+p%3D1,2,+or+infinite.+Suppose+b1%3E%3Db2%3E%3Db3&source=bl&ots=7_CwPi_Qes&sig=AdHOn8-CGzckpQGRawzOrx9w7Jo&hl=en&sa=X&ved=0ahUKEwiNztHj29zXAhXCk-AKHZ9UCTgQ6AEILjAB#v=onepage&q&f=false, 5.3 The Full-Rank Least Squares Problem.

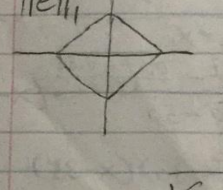
We say the system $Ax=b$ is overdetermine, An overdetermined system has no exact solution since "b" must be an element of $\mathrm{ran}(A)$, a proper subspace of $\mathbb{R}^m$. This suggests that we strive to minimize $\|Ax-b\|_p$ for some suitable choice of p.

- Minimization in 1-norm and infinity-norm is complicated by the fact that the function $f(x)=\|Ax-b\|_p$ is not differentiable for these value of p.

- The 2-norm is under orthogonal transformation. This means that we can seek an orthogonal Q such that the equivalent problem of minimizing $\|(Q^TA)x-(Q^Tb)\|_2$ is "easy" to solve.

Q3. $\min_{x \in \mathbb{R}} \left\| \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} x - \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right\|_p$,

$p = 1$, $\|e\|_1$



$\min \|Ax - b\|_1$ $\quad \min \sum_{i=1}^{n} y_i$

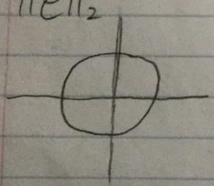
$\min_{x \in \mathbb{R}} \left\| \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} x - \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right\|_1$ $\quad$ subject to $-y \le Ax - b \le y$

$\Rightarrow \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} x = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$ $\quad \boxed{\text{We known that: } b_1 \ge b_2 \ge b_3}$

$x_1 = b_1 \ge x_2 = b_2 \ge x_3 = b_3$

$\boxed{\text{So, } x_{opt} = b_2}$

$p = 2$, $\|e\|_2$



$\min \|Ax - b\|_2^2$ $\quad \min \sum_{i=1}^{n} y_i^2$

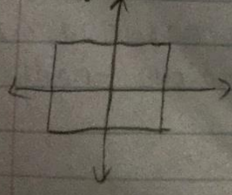
$A^T \cdot A \cdot x = b \cdot A^T$

$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} x = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$

$3x = b_1 + b_2 + b_3$

$$\boxed{\begin{aligned} 3x_{opt} &= (b_1 + b_2 + b_3) \\ &= \frac{(b_1 + b_2 + b_3)}{3} \end{aligned}}$$

$p = \infty$, $\|e\|_\infty$



$\min \|Ax - b\|_\infty$ $\quad \min\ y$ or $\max_k |y_k|$

$x_{opt} = \frac{b_1 + b_3}{2}$ $\quad \|e\|_1 \Leftrightarrow \|e\|_\infty$

$\|e\|_\infty \le \|e\|_2$ $\quad \frac{b_2}{1} \Longleftrightarrow \frac{b_1 + b_3}{2}$