hw5

August 6, 2018

1 Computing eigenvalues.

In [1]:
```
A = [ 0 -17 21;
      0 13 -15;
      0 10 -12
];
chi = poly(A)
lambdas = eig(A)
```

```
chi =

    1.0000   -1.0000   -6.0000         0


lambdas =

         0
    3.0000
   -2.0000
```

In [2]:
```
n = size(A,1);
N = 100;
q0 = rand(n,1)
epsilon = 1e-6;
q = q0./norm(q0);
lambda_curr = zeros(n,1);
for k = 1:N
    lambda_prev = lambda_curr;
    v = A*q;
    lambda_curr = q'*v;
    q = v./norm(v);
    if abs(lambda_curr - lambda_prev)<epsilon
        break;
    end
```

```
        end
        lambda_curr
```

```
q0 =

    0.8147
    0.9058
    0.1270


lambda_curr =

    3.0000
```

The algorithm is only able to solve one possible λ.

```
In [3]: for k = 1:N
            [L,U] = lu(A);
            A = U*L;
        end
        A
        diag(A)
```

```
A =

         0   -3.0000   21.0000
         0    3.0000  -15.0000
         0    0.0000   -2.0000


ans =

         0
    3.0000
   -2.0000
```

The diagonal elements of the output are the eigenvalues of matrix A.

4 Interpolation points

The observation points are equally spaced with $h = 0.5$, then

$$c_{j-1} + 4c_j + c_{j+1} = \frac{3}{h^2}(a_{j+1} - 2a_j + a_{j-1})$$

for $j = 1, \cdots, n-1$

```
In [4]: x = [0 0.5 1 1.5 2];
        y = [0.5 1.425639 2.640859 4.009155 5.305472];
        h = 0.5;
        k = 3/(h.^2);
        n = length(x) - 1
        %Solve a
        a = zeros(n+1,1);
        for i=1:(n+1)
            a(i) = y(i);
        end
        a(1:n);
```

```
n =

     4
```

4.1 a

```
In [5]: %Solve c
        cc = zeros(n-1,n-1);
        for i=2:(n-2)
            cc(i,i-1) = 1;
            cc(i,i) = 4;
            cc(i,i+1) = 1;
        end
        cc(1,1) = 6;
        cc(n-1,n-1) = 6;
        cc
        aa = zeros(n-1,1);
        for i = 2:n
            aa(i-1) = (a(i-1) - 2*a(i) + a(i+1))*k;
        end
        aa
        c = cc\aa;
        c0 = 2*c(1)-c(2)
        c = [c0;c]
        % Compute b and d
        d = zeros(n,1);
        b = zeros(n,1);
        for i = 1:n-1
            d(i) = (c(i+1) - c(i))/(3*h);
            b(i) = (a(i+1)-a(i))/h - (2*c(i)+c(i+1))*h/3;
        end
```

```
        d(n) = d(n-1)
        b(n) = b(n-1) + (c(n)+c(n-1))*h
        % Combine efficients
        coefs_nak = [d c b a(1:n)]
        obs = x';
        nak = mkpp(obs, coefs_nak);
        %Verification
        s = spline(x,y);
        standard_nak = s.coefs()
```

```
cc =

     6     0     0
     1     4     1
     0     0     6


aa =

    3.4750
    1.8369
   -0.8637


c0 =

    0.8079


c =

    0.8079
    0.5792
    0.3504
   -0.1440


d =

   -0.1525
   -0.1525
   -0.3296
   -0.3296


b =
```

```
    1.4855
    2.1790
    2.6438
    2.7470


coefs_nak =

   -0.1525    0.8079    1.4855    0.5000
   -0.1525    0.5792    2.1790    1.4256
   -0.3296    0.3504    2.6438    2.6409
   -0.3296   -0.1440    2.7470    4.0092


standard_nak =

   -0.1525    0.8079    1.4855    0.5000
   -0.1525    0.5792    2.1790    1.4256
   -0.3296    0.3504    2.6438    2.6409
   -0.3296   -0.1440    2.7470    4.0092
```

4.2 b

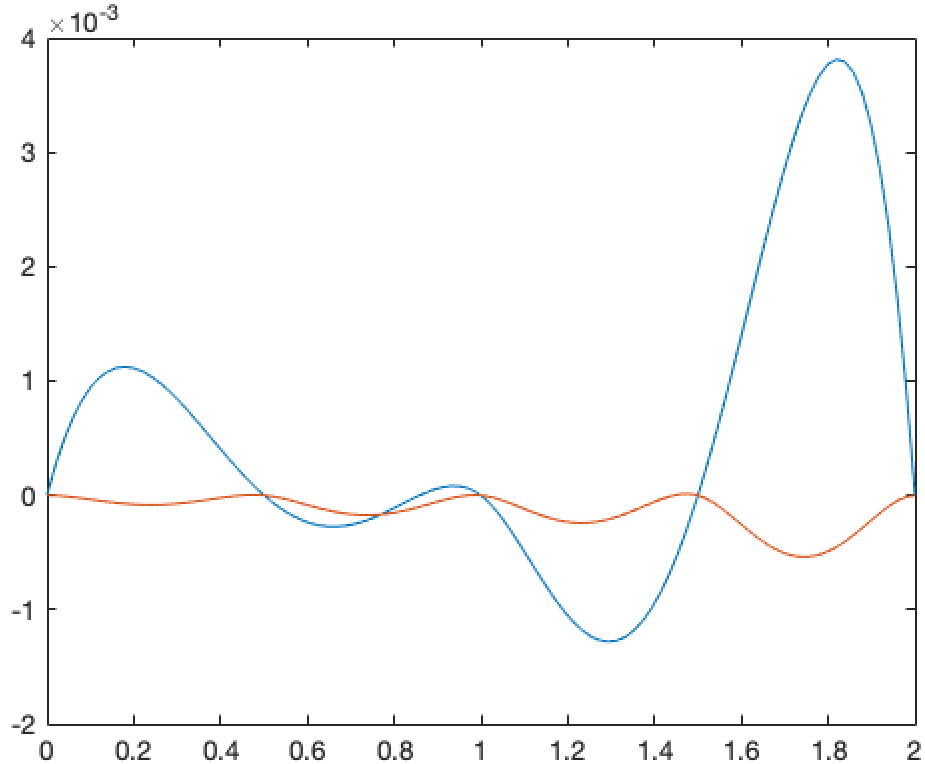
In [6]:
```
yp = [1.5 2.305472];
for i=1:n
    h(i) = x(i+1)-x(i);
end
cc = zeros(n+1,n+1);
for i = 2:n
    cc(i,i) = 2*(h(i)+h(i-1));
    cc(i,i-1) = h(i-1);
    cc(i-1,i) = h(i-1);
end
cc(1,1)=2*h(1);
cc(n+1,n+1) = 2*h(n);
cc(n+1,n) = h(n);
cc(n,n+1) = h(n);
aa = zeros(n+1,1);
for i=2:n
    aa(i) = 3*(a(i+1)-a(i))/h(i) - 3*(a(i)-a(i-1))/h(i-1);
end
aa(1) = 3*(a(2)-a(1))/h(1) - 3*yp(1);
aa(n+1) = 3*yp(2) - 3*(a(n+1)-a(n))/h(n);
c = cc\aa;
% Compute b and d
d = zeros(n,1);
```

```
b = zeros(n,1);
for i = 1:n
    d(i) = (c(i+1)-c(i))/(3*h(i));
    b(i) = (a(i+1)-a(i))/h(i)-h(i)*(c(i+1)+2*c(i))/3;
end

% Combine efficients
coefs_cs = [d c(1:4) b a(1:n)]
obs = x';
cs = mkpp(obs, coefs_cs);
%Verification
cs = spline(x,[1.5 y 2.305472]);
cs.coefs
```

```
coefs_cs =

   -0.1063    0.7557    1.5000    0.5000
   -0.1747    0.5963    2.1760    1.4256
   -0.2869    0.3342    2.6412    2.6409
   -0.4782   -0.0961    2.7602    4.0092


ans =

   -0.1063    0.7557    1.5000    0.5000
   -0.1747    0.5963    2.1760    1.4256
   -0.2869    0.3342    2.6412    2.6409
   -0.4782   -0.0961    2.7602    4.0092
```

4.3 c

```
In [7]: t = linspace (0,2);
        f = (1+t).^2 - 0.5*exp(t);
        plot(t, f - ppval(nak,t))
        hold on;
        plot(t, f - ppval(cs,t))
```



The clamped result is better.

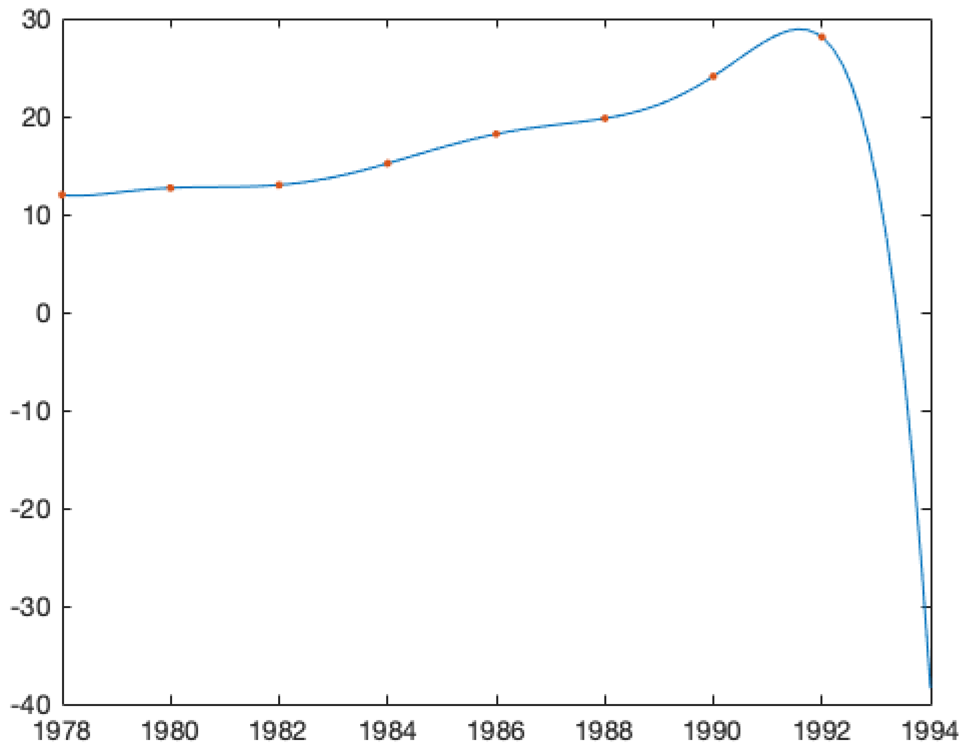
5 Lake Pollution

5.1 a

```
In [8]: x=[1978 1980 1982 1984 1986 1988 1990 1992];
        y=[12.0 12.7 13.0 15.2 18.2 19.8 24.1 28.1];
        syms t;
        n = length(x);
        L = 0;
        for(i = 1:n)
            e = 1;
            for j = 1:n
                if (i~=j)
                  e = e*((t-x(j))/(x(i)-x(j)));
                end
            end
            L = L + e*y(i);
        end
        t = linspace (1978,1994);
```

```
r = eval(L);
plot(t,r)
hold on;
plot(x,y,'.')
t = 1994;
eval(L)
```

```
ans =

  -38.4000
```

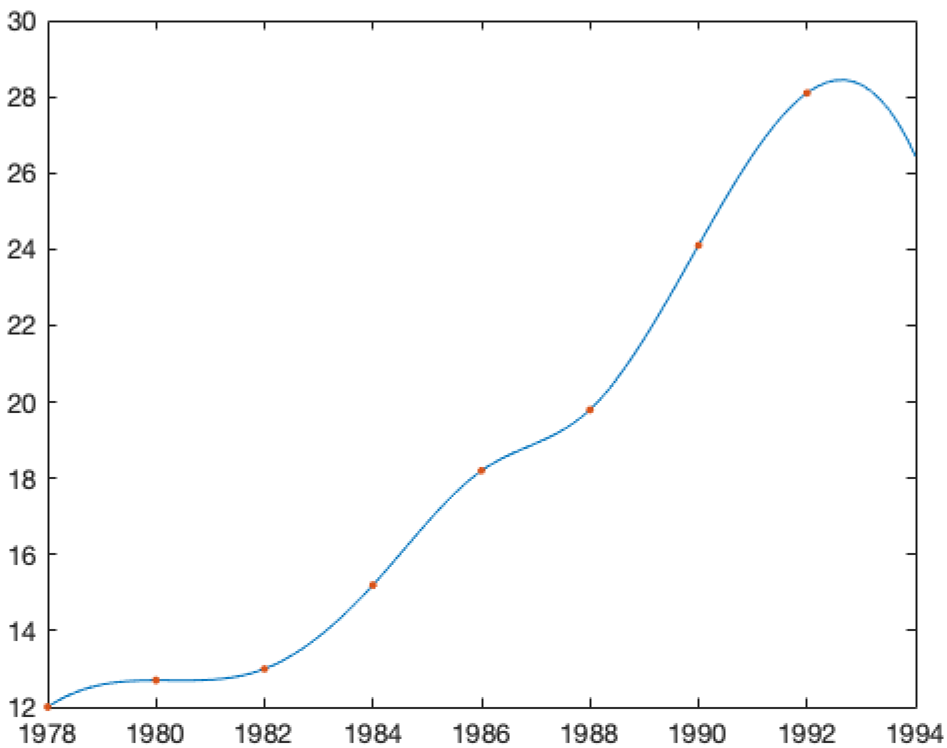


5.2 b

```
In [9]: x=[1978 1980 1982 1984 1986 1988 1990 1992];
        y=[12.0 12.7 13.0 15.2 18.2 19.8 24.1 28.1];
        t = linspace (1978,1994);
        r = spline(x,y,t);
        plot(t,r)
```

```
hold on;
plot(x,y,'.')
spline(x,y,1994)
```

```
ans =

   26.4407
```



5.3 c

The cubic spline result makes more sense because functions of high degree causes much bias.

2. (a) $Av = \lambda v$.

$\Leftrightarrow \chi(\lambda) = \det(\lambda I - M) = 0$.

Note that $\chi(\lambda) = 0$ is n degree equation of λ.

According to the fundamenental theorem of algebra,

① $\chi(\lambda)$ has at most n different roots.

Therefore, A can have at most n eigenvalues.

② if μ is a complex eigenvalue of A, then

μ is a root of $\chi(\lambda) = 0$.

$\bar{\mu}$ is also a root of $\chi(\lambda) = 0$.

$\therefore$ $\bar{\mu}$ is also an eigenvalue of A.

(b) ① Suppose v_0 is a λ-eigenvector of A. $\quad Av_0 = \lambda v_0$

$$P^{-1}AP(P^{-1}v_0) = P^{-1}A(PP^{-1})v_0 = P^{-1}Av_0 = P^{-1}\lambda v_0$$
$$= \lambda(P^{-1}v_0)$$

$\Rightarrow$ $P^{-1}v_0$ is a λ-eigenvector of $P^{-1}AP$

② Suppose v_1 is a λ-eigenvector of $P^{-1}AP$. $P^{-1}APv_1 = \lambda v_1$.

$$PP^{-1}APv_1 = P\lambda v_1$$
$$A(Pv_1) = \lambda \cdot (Pv_1)$$

$\Rightarrow$ Pv_1 is a λ-eigenvector of A.

From ① and ②, P and $P^{-1}AP$ have the same eigenvalues

(c) $\mathrm{diag}(D) = (d_1, d_2, \cdots, d_n)$

$$\chi(\lambda) = \det|\lambda I - D| = \prod_{i=1}^{n} (\lambda - d_i)$$

$\chi(\lambda) = 0$ has n roots: $d_1, d_2, d_3, \cdots d_n$.

(d) $\chi(\lambda) = \det(\lambda I - A)$

Constant term $= \chi(0) = \det(-A) = (-1)^n \det(A)$. ①

On the other hand,

$$\chi(\lambda) = \prod_{i=1}^{n} (\lambda - d_i)$$

constant term $= \prod_{i=1}^{n} (-d_i) = (-1)^n \prod_{i=1}^{n} d_i$ ②

From ①, ②.

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

3. (a)

$|(x-x_0)(x-x_1)| = |(x+1)(x-1)| = |x^2-1|$.

for $x\in[-1,1]$. $x^2-1\le 0$.

$|(x-x_0)(x-x_1)| = 1-x^2 \le 1$.

$|(x-x_0)(x-x_1)|_{max} = 1$ when $x=0$.

(b) $|(x-x_0)(x-x_1)| = |x^2-\frac{1}{2}|$. $x^2\in[0,1]$. $x^2-\frac{1}{2}\in[-\frac{1}{2},\frac{1}{2}]$.

$\therefore$ $|x^2-\frac{1}{2}|_{max} = \frac{1}{2}$. when $x=0$ or $x=1$ or $x=-1$

$\frac{1}{2}<1$. The maximum is less than in (a)

(c)

$x_0, x_1 \in [-1,1]$. Suppose $x_0 < x_1$.

$|(x-x_0)(x-x_1)|$ takes maximum at $x=-1$ or $x=1$ or $x=\frac{x_0+x_1}{2}$.

$|f(-1)| = |(-1-x_0)(-1-x_1)| = |(1+x_0)(1+x_1)| = (1+x_0)(1+x_1) = A$

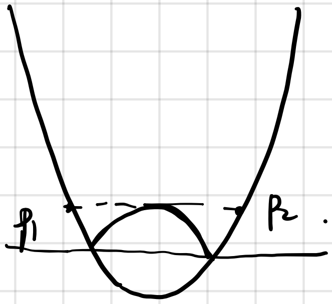
$|f(1)| = |(1-x_0)(1-x_1)| = (1-x_0)(1-x_1) = B$

$\left|f\left(\frac{x_0+x_1}{2}\right)\right| = \left|\frac{x_1-x_0}{2}\cdot\frac{x_0-x_1}{2}\right| = \left|-\frac{1}{4}(x_1-x_0)^2\right| = \frac{1}{4}(x_1-x_0)^2 = C$.

$A-B = (x_0+x_1)$

$x_0+x_1>0 \Rightarrow A>B$.

$x_0+x_1\le 0 \Rightarrow A\le B$.



Solve

$$\frac{1}{4}(x_1-x_0)^2 = (x-x_1)(x-x_0).$$

$$x^2-(x_0+x_1)x+x_0x_1=\frac{1}{4}(x_1-x_0)^2$$

$$x^2-(x_0+x_1)x-\left(\frac{1}{4}x_1^2-\frac{3}{2}x_0x_1+\frac{1}{4}x_2^2\right)=0$$

$$\Delta = (x_0+x_1)^2+4\left(\frac{1}{4}x_1^2-\frac{3}{2}x_0x_1+\frac{1}{4}x_0^2\right)$$

$$=(x_0^2+2x_0x_1+x_1^2)+(x_1^2-6x_0x_1+x_0^2)$$

$$=2x_0^2-4x_0x_1+2x_1^2$$

$$=2(x_1-x_0)^2.$$

$$x=\frac{x_0+x_1\pm\sqrt{2}(x_1-x_0)}{2}$$

if $1>p_2=\dfrac{(1-\sqrt{2})x_0+(1+\sqrt{2})x_1}{2}$. then $B>C$.

if $-1<p_1=\dfrac{(1+\sqrt{2})x_0+(1-\sqrt{2})x_1}{2}$. then $A>C$.

if $p_1\le -1<1\le p_2$. then $f_{max}=C$.

Therefore,

$$f_{max}=\begin{cases} C=\frac{1}{4}(x_1-x_0)^2 & \text{when } \begin{cases}(1-\sqrt{2})x_0+(1+\sqrt{2})x_1\ge 2\\ (1+\sqrt{2})x_0+(1-\sqrt{2})x_1\le -2\end{cases}\\[2ex] B=(1-x_0)(1-x_1) & \text{when } \begin{cases}(1-\sqrt{2})x_0+(1+\sqrt{2})x_1<2\\ x_0+x_1\le 0\end{cases}\\[2ex] A=(1+x_0)(1+x_1) & \text{when } \begin{cases}\dfrac{(1+\sqrt{2})x_0+(1-\sqrt{2})x_1}{2}>-2\\ x_0+x_1>0\end{cases}\end{cases}$$

if $x_0+x_1 \leq 0$.

$$f_{max} \geq \frac{\frac{(x_0-x_1)^2}{4}+(1-x_1)(1-x_0)}{2} = \frac{(x_0+x_1-2)^2}{8} \geq \frac{1}{2}.$$

if $x_0+x_1 > 0$.

$$f_{max} \geq \frac{\frac{(x_0-x_1)^2}{4}+(1+x_1)(1+x_0)}{2} = \frac{(x_0+x_1+2)^2}{8} > \frac{1}{2}.$$

Therefore, $f_{max} \geq \frac{1}{2}$ is true for all $x_0, x_1 \in [-1,1]$.

Note that

$A = 1+x_0+x_1+x_0x_1 < 4$. $(x_0 \neq x_1)$.

$B = 1-(x_0+x_1)+x_0x_1 < 4$

$C = \frac{1}{4}(x_0-x_1)^2 < 1$.

$\therefore f_{max} < 4$.

$\Rightarrow f_{max} \in [\frac{1}{2}, 4)$.

6. (a)

Interpolation $\Rightarrow a_j = y_j$ ①

Continuity $\Rightarrow a_j + b_j(x_{j+1} - x_j) + c_j(x_{j+1} - x_j)^2 = a_{j+1}$ ②

Smoothness $\Rightarrow$ $b_j + 2c_j x_{j+1} - 2c_j x_j = b_{j+1}$ ③

From ①, ②, ③.

$$b_{j+1} = b_j + 2(x_{j+1} - x_j) \cdot \frac{(y_{j+1} - y_j) - b_j(x_{j+1} - x_j)}{(x_{j+1} - x_j)^2}$$

$$b_{j+1} = -b_j + \frac{2(y_{j+1} - y_j)}{x_{j+1} - x_j}.$$

(b) $3n - (3n-1) = 1$.

One more condition needs to be provided,

e.g. ① not-a-knot: $S_0''(x_1) = S_1''(x_1)$

or $S_{n-2}''(x_{n-1}) = S_{n-1}''(x_{n-1})$

② clamped spline: $S_0'(x_0) = \lambda_1$ (constant)

or $S_{n-1}'(x_n) = \lambda_2$ (constant).

Propose one condition:

$S_0'(x_0) = b_0 + 2c_0 x_0 - 2c_0 x_0 = b_0 = \lambda_1$ (given constant).