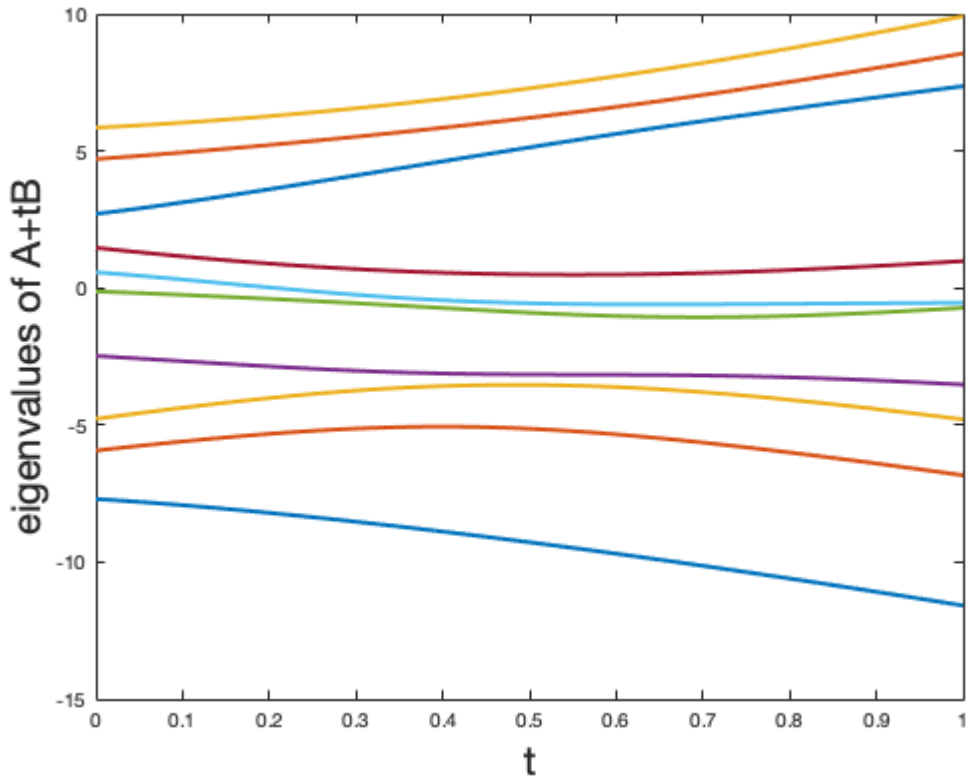
Contents

Problem 1

```
A=randn(10); A=A+A';
B=randn(10); B=B+B';

T=0:.01:1;
e=[];
for t = 0:.01:1;
    e=[e eig(A+t*B)];
end

figure(13)
clf
plot(T,e,'linewidth',2)
xlabel('t','fontsize',20)
ylabel('eigenvalues of A+tB','fontsize',20)
```

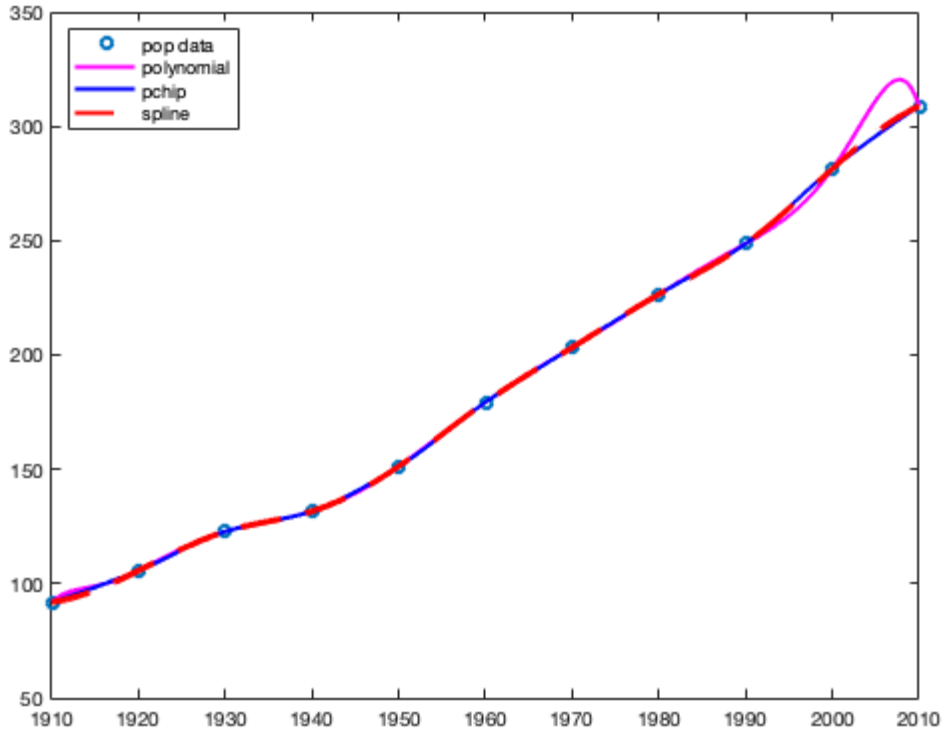


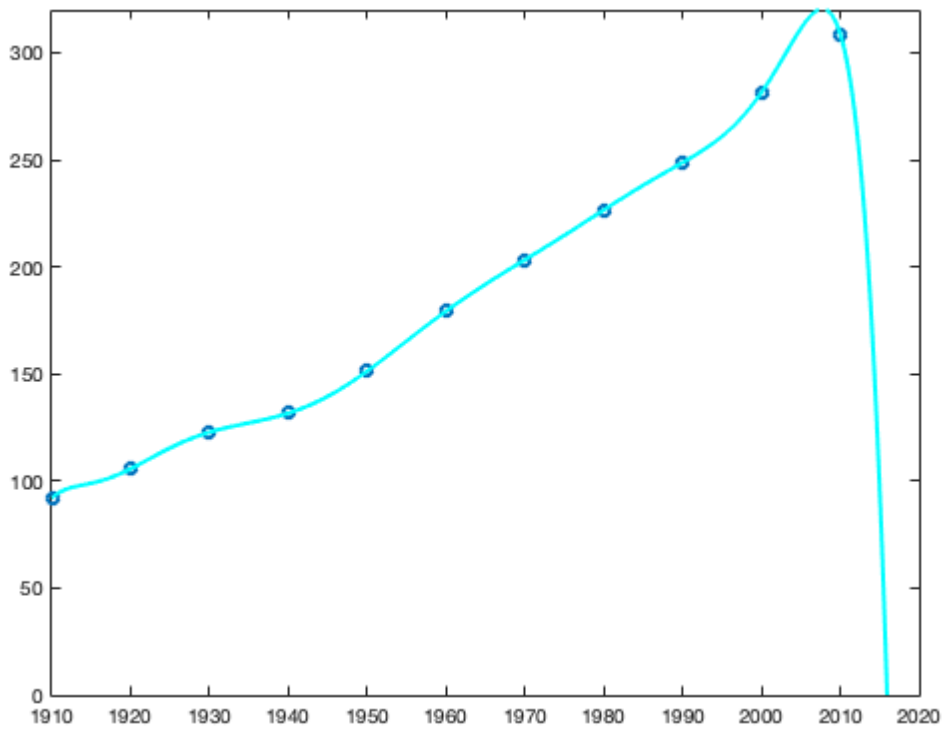
Problem 2

```
year = [1910  1920  1930  1940  1950  1960  1970  1980  1990 2000 2010];
pop = [91.97  105.71  122.78  131.67  151.33  179.32  203.30  226.54  248.71 281.42 308.75];

u = 1910:.1:2010;
```

```
p = polyinterp(year,pop,u);
pc = pchip(year,pop,u);
ps = spline(year,pop,u);

figure(11)
plot(year,pop,'o',u,p,'m',u,pc,'b',u,ps,'r--','linewidth',2)
legend('pop data','polynomial','pchip','spline','location','northwest')

% pchip and spline both seem to work well between the points. The
% polynomial has bad behavior between the last two points due to polynomial
% wiggle.

u2 = 1910:.1:2020;
p2 = polyinterp(year,pop,u2);
figure(12)
plot(year,pop,'o',u2,p2,'c','linewidth',2)
axis([1910 2020 0 320])

% The polynomial interpolant crosses zero at year 2016.

pchip(year,pop,2005)

% Using pchip gives us a value of 295,727,370 at 2015.
```

```
ans =

  295.7274
```





Problem 3

```
f   = @(x)(2-exp(x));
fp  = @(x)(-exp(x));
fpp = @(x)(-exp(x));
```

```
% Starting guess:
x = 3;  % For Newton's method
y = x;  % For Halley's method

fprintf('\n Step  Newton       (error)     Halley       (error) \n')
for j=1:10
    x = x - f(x)/fp(x);
    y = y - 2*f(y)*fp(y)/(2*fp(y)^2-f(y)*fpp(y));
    fprintf(' %i    %.8f   %.3e   %.8f   %.3e\n',j,x,x-log(2),y,y-log(2))
end

% Starting at x=3, Halley's method converged to within roundoff in
% 4 steps. It took Newton's method 7 steps.

% Halley's method is order 3.
```

```
 Step  Newton       (error)     Halley       (error)
 1    2.09957414   1.406e+00   1.36222801   6.691e-01
 2    1.34459131   6.514e-01   0.71703877   2.389e-02
 3    0.86588373   1.727e-01   0.69314832   1.136e-06
 4    0.70724297   1.410e-02   0.69314718   1.110e-16
 5    0.69324606   9.888e-05   0.69314718   1.110e-16
 6    0.69314719   4.889e-09   0.69314718   1.110e-16
 7    0.69314718   1.110e-16   0.69314718   1.110e-16
 8    0.69314718   1.110e-16   0.69314718   1.110e-16
 9    0.69314718   1.110e-16   0.69314718   1.110e-16
 10    0.69314718   1.110e-16   0.69314718   1.110e-16
```

Published with MATLAB® R2020a