Answers to Problem Set 4
Group name: Ferienspass

Sebastian Kühnl: 5642348
Alexander Dück (as: reebyte): 5504077
Patrick Blank (as: paddyblank): 6729110
Christian Wierschem: 6729288

1 Question 1

Chebyshev approximation using equidistant nodes and Chebyshev nodes. However, there is a difference between lecture slides 7 and the notes from te lecture, as you will see in the code:

```
function [yequi,ychebsli,ycheblec]=cheb(fct,x,m,xmin,xmax)

%define function space with fundefn
fspace=fundefn('cheb',m-1,xmin,xmax);
distance=(xmax-xmin)/(m-1);
nodesequi=zeros(m,1);
ynodesequi=zeros(m,1);
nodeschebslides=zeros(m,1);
ynodeschebsli=zeros(m,1);
ynodescheblec=zeros(m,1);
nodescheblecture=zeros(m,1);
%create nodes
%also, calculate function values at x
for j=1:m
  nodesequi(j)=xmin+(j-1)*distance;              % equidistant nodes
  ynodesequi(j)=fct(nodesequi(j));               % function values
  nodeschebslides(j)=-cos((2*j-1)*pi/(2*m));     % Chebyshev nodes according to slide set 7
  nodescheblecture(j)=-cos((2*j-1)*pi/(m));     % Chebyshev nodes according to lecture notes
  ynodeschebsli(j)=fct(nodeschebslides(j));
  ynodescheblec(j)=fct(nodescheblecture(j));
end

%calculate the matrix of basis functions
Bequi=funbas(fspace,nodesequi);  %equidistant
Bchebsli=funbas(fspace,nodeschebslides);  %Chebyshev
```

```
Bcheblec=funbas(fspace,nodescheblecture);  %Chebyshev


%get polynomial coefficients
cequi=Bequi\ynodesequi;  %equidistant
cchebsli=Bchebsli\ynodeschebsli;  %chebychev
ccheblec=Bcheblec\ynodescheblec;

%approximate the function
yequi=funeval(cequi,fspace,x);
ychebsli=funeval(cchebsli,fspace,x);
ycheblec=funeval(ccheblec,fspace,x);

end
```

Linear and cubic splines, also using the Miranda-Fackler toolbox:

```
function [yspllin,ysplcub]=spl(fct,x,m,xmin,xmax)
  fspacespllin=fundefn('spli',m-1,xmin,xmax,1);  %linear splines
  fspacesplcub=fundefn('spli',m-1,xmin,xmax,3);  %cubic splines
  distance=(xmax-xmin)/(m-1);
  nodesspl=zeros(m,1);
  ynodes=zeros(m,1);
  %nodes
  for i=1:m
    nodesspl(i)=xmin+(i-1)*distance;   %eqidistant nodes
    ynodes(i)=fct(nodesspl(i));        %fct values at nodes
  end

  %calculate the matrix of basis functions
  Bspllin=funbas(fspacespllin,nodesspl);
  Bsplcub=funbas(fspacesplcub,nodesspl);

  %get polynomial coefficients
  cspllin=Bspllin\ynodes;
  csplcub=Bsplcub\ynodes;

  %approximate the function
  yspllin=funeval(cspllin,fspacespllin,x);
  ysplcub=funeval(csplcub,fspacesplcub,x);

end
```

The function, which was given in the task, defined for potential vector input:

```
function y=simplef(x)
  y=1./(1+25.*x.^2);
end
```

Main code for PS4P1:

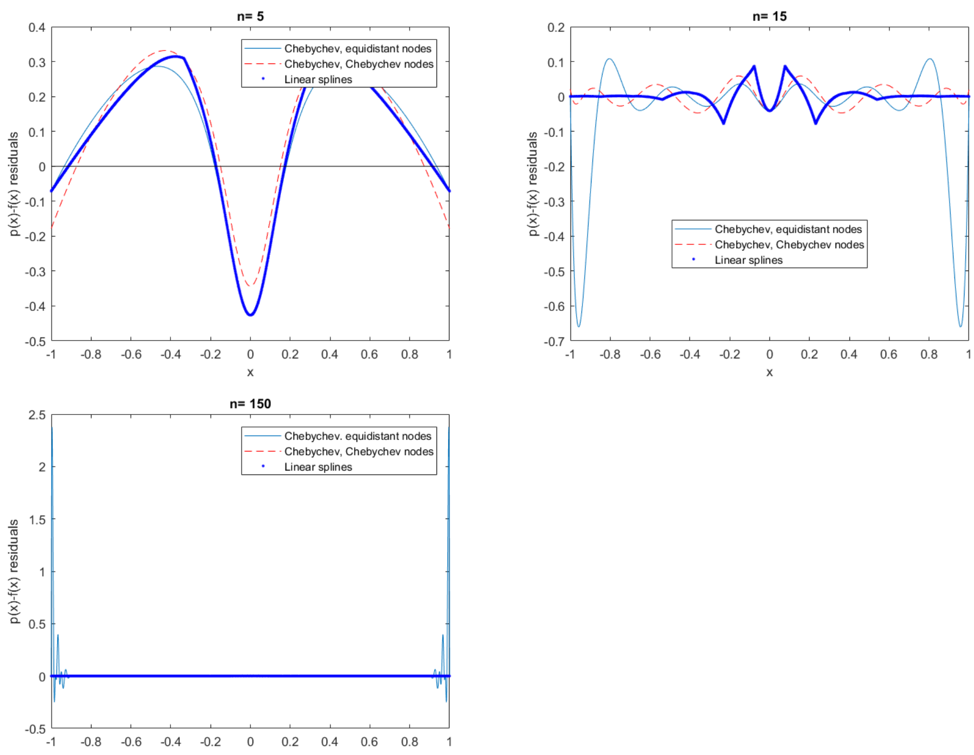
```
%PS4P1
clear;
close all;
clc;

%Chebychev

%variable declaration
n1=5;  %number of nodes
n2=15;
n3=150;
%f(x) is simplef.m
f=@simplef;
xmin=-1;
xmax=1;
b=linspace(xmin,xmax,2000); %x-space
b=b';

[yapequi,yapchebsli,yapcheblec]=cheb(f,b,n1,xmin,xmax)
    ;
[yapequi2,yapchebsli2,yapcheblec2]=cheb(f,b,n2,xmin,
    xmax);
[yapequi3,yapchebsli3,yapcheblec3]=cheb(f,b,n3,xmin,
    xmax);


%SPLINES equidistant nodes
[yapspllin,yapsplcub]=spl(f,b,n1,xmin,xmax);
[yapspllin2,yapsplcub2]=spl(f,b,n2,xmin,xmax);
[yapspllin3,yapsplcub3]=spl(f,b,n3,xmin,xmax);

%actual function
yact=simplef(b);


%plots compare with same n
figure
plot(b,yapequi-yact,b,yapchebsli-yact,'--r',b,
    yapspllin-yact,'.b')
```

```
line([-1, 1],[0, 0],'color','black')
xlabel('x')
ylabel('p(x)-f(x) residuals')
title('n= 5')
legend('Chebychev, equidistant nodes','Chebychev, Chebychev nodes','Linear splines')

figure
plot(b,yapequi2-yact,b,yapchebsli2-yact,'--r',b,yapspllin2-yact,'.b')
xlabel('x')
ylabel('p(x)-f(x) residuals')
title('n= 15')
legend('Chebychev, equidistant nodes','Chebychev, Chebychev nodes','Linear splines')

figure
plot(b,yapequi3-yact,b,yapchebsli3-yact,'--r',b,yapspllin3-yact,'.b')
xlabel('x')
ylabel('p(x)-f(x) residuals')
title('n= 150')
legend('Chebychev. equidistant nodes','Chebychev, Chebychev nodes','Linear splines')


%plots comparison same node method (no cheb lecture and no cubic splines)

figure
plot(b,yapequi-yact,'.b',b,yapequi2-yact,'--r',b,yapequi3-yact)
xlabel('x')
ylabel('p(x)-f(x)')
title('Chebychev, equidistant node approximations')
legend('n=5','n=15','n=150')

figure
plot(b,yapchebsli-yact,'.b',b,yapchebsli2-yact,'--r',b,yapchebsli3-yact)
xlabel('x')
ylabel('p(x)-f(x)')
title('Chebychev, Chebychev node approximations')
legend('n=5','n=15','n=150')

figure
```

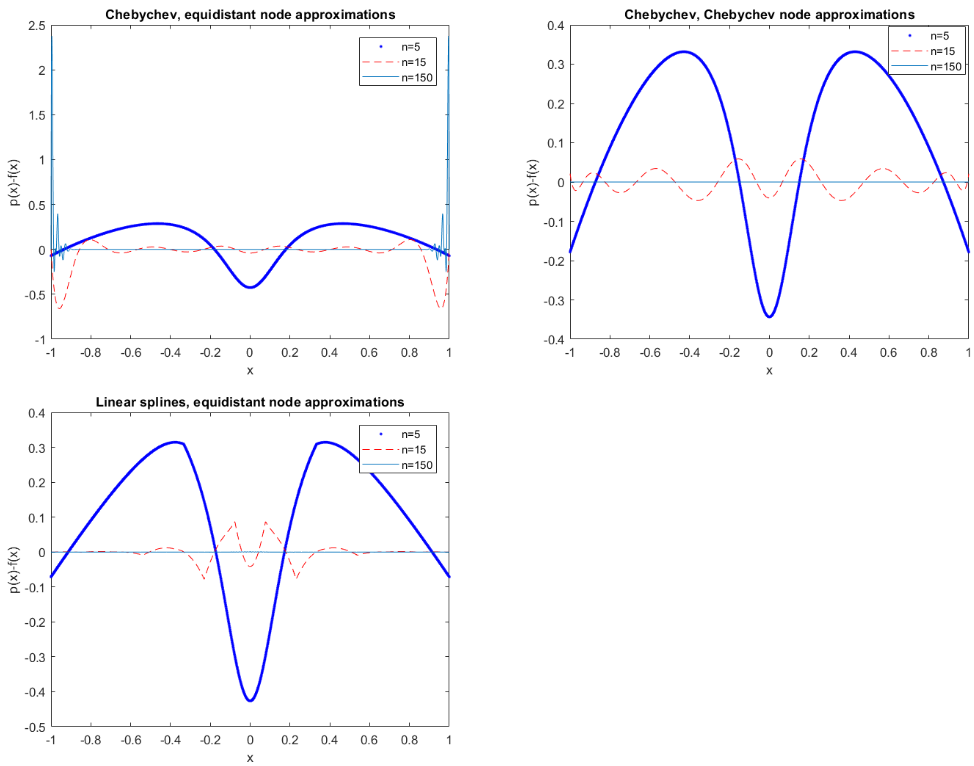
```
plot(b,yapspllin-yact,'.b',b,yapspllin2-yact,'--r',b,yapspllin3-yact)
xlabel('x')
ylabel('p(x)-f(x)')
title('Linear splines, equidistant node approximations')
legend('n=5','n=15','n=150')


%compare linear splines and cubic splines

%plots compare with same n

figure
plot(b,yapspllin-yact,b,yapsplcub-yact,'--r')
line([-1, 1],[0, 0],'color','black')
xlabel('x')
ylabel('p(x)-f(x) residuals')
title('Splines, n= 5')
legend('linear','cubic')

figure
plot(b,yapspllin2-yact,b,yapsplcub2-yact)
xlabel('x')
ylabel('p(x)-f(x) residuals')
title('Splines, n= 15')
legend('linear','cubic')

figure
plot(b,yapspllin3-yact,b,yapsplcub3-yact,'--r')
xlabel('x')
ylabel('p(x)-f(x) residuals')
title('Splines, n= 150')
legend('linear','cubic')

%compare slides and lecture

%plots compare with same n
figure
plot(b,yapcheblec-yact,b,yapchebsli-yact,'--r')
line([-1, 1],[0, 0],'color','black')
xlabel('x')
ylabel('p(x)-f(x) residuals')
title('Chebychev, Chebychev nodes, n= 5')
legend('Lecture','Slides')

```

```
figure
plot(b,yapcheblec2-yact,b,yapchebsli2-yact,'--r')
xlabel('x')
ylabel('p(x)-f(x) residuals')
title('Chebychev, Chebychev nodes, n= 15')
legend('Lecture','Slides')

figure
plot(b,yapcheblec3-yact,b,yapchebsli3-yact,'--r')
xlabel('x')
ylabel('p(x)-f(x) residuals')
title('Chebychev, Chebychev nodes, n= 150')
legend('Lecture','Slides')
```

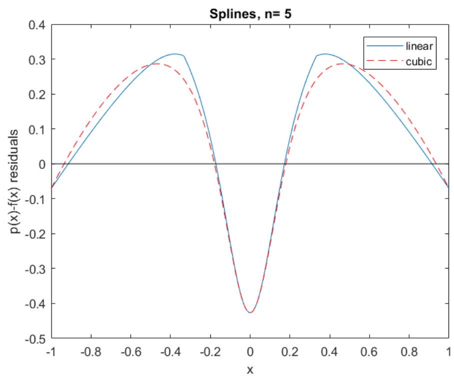
Plots are ordered in chronological order! (For comparison, the residuals to actual function are plotted.)

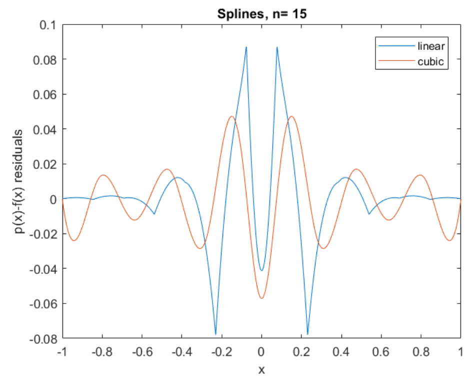


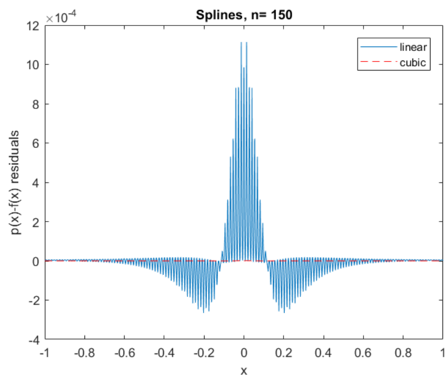
For n=5, equidistant nodes and Chebyshev nodes as well as linear splines are very similar. For n=15, linear splines become more edgy. It performs well at the edges and average else. Both Chebyshev approximations are very similar in [-0.75;0.75], while equidistant nodes fall off at the edges (as expected). For n=150 this effect is even stronger, but it moves closer to the corner. The others are not comparable due to the residual scale. The effect does not occur when using Chebyshev nodes because there are more nodes at the corner to prevent these large fluctuations.



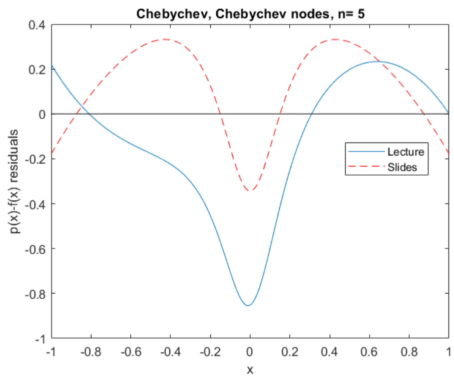
In this figure again the effect of equidistant nodes when using Chebyshev can be seen as large fluctuations at the corner. Besides this, as the number of nodes increases, the approximation gets closer to the real function.

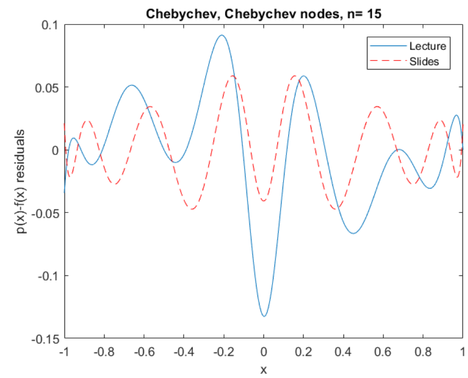


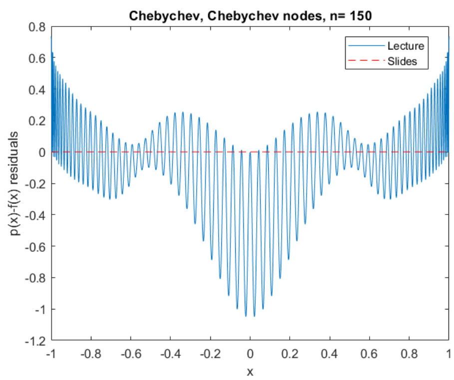




Linear and cubic splines are very similar when n=5. For n=15 one can observe that cubic splines are smoother than linear splines and perform better in the center (around 0), whereas linear splines perform better at the corner (it becomes smooth and then becomes nearly a straight line). For n=150, at first sight, linear splines perform badly, but it is only relative to cubic splines (look at the scale). As n increases, the approximation gets better when using splines.







The slides formula seems to be the right one. The function is symmetric and so is the approximation. The lecture formula leads to very odd (i.e. asymmetric) approximations.

In general, it seems to be very odd that the residuals at 0 are not 0, because there should be a node and thus the residual should be zero. Maybe it is due to the toolbox calculations. Other possibilities have been thought of and precluded.

2 Question 2