Numerical Analysis (089180) A.Y. 2022-2023
prof. Simona Perotto Luca Liverotti
simona.perotto@polimi.it luca.liverotti@polimi.it

Lab 8 – Homework Solutions

December 2, 2022

A Homework

Homework 8.1. Consider the function $f(x) = e^x$ in the interval $[-1, 1]$ and the associated piecewise linear interpolating polynomial on a set of nodes $\{x_i\}$.

a. Determine how many equally spaced nodes x_i are needed in order to ensure an interpolation error with infinity norm smaller than 10^{-3}.

b. Construct the interpolating polynomial based on such a number of nodes, compute the associated error and check if the request on accuracy is satisfied.

Solution Homework 8.1.

hw_8_1.m

```
clc
clear all
close all

f = @(x) exp(x);

% a)
% Using Theorem 10.2 and imposing that the right hand side is less or equal
% than 1e-3

h = sqrt( 1e-3 * 8 / exp(1) )

n = ceil(2 / h) + 1 % = 38


% b)

x = linspace(-1, 1, n);
y = exp(x);

x_plot = linspace(-1, 1, 1000);
y_plot = interp1(x,y,x_plot);

figure
plot(x_plot, f(x_plot), 'k-', x_plot, y_plot, 'r-', x, y, 'rx', 'LineWidth', 2, 'MarkerSize',
8)
axis([-1.1 1.1 0 3])
set(gca,'FontSize', 16)
set(gca,'LineWidth', 1.5)

err = max(abs(f(x_plot) - y_plot))

% The requirement is satisfied.
```

Homework 8.2. Consider the function

$$f(x) = \frac{1}{(x-0.3)^2+0.01} + \frac{1}{x^2+0.04} - 6.$$

a. Sample $f(x)$ at 10 equally spaced nodes in the interval $[-1,3]$ and construct the minimum degree interpolating polynomial.

b. Numerically compute the error. Provide a method to improve the result.

Solution Homework 8.2.

hw_8_2.m

```
clc
clear all
close all

f = @(x) 1./((x - 0.3).^2 + 0.01) + 1./(x.^2 + 0.04) - 6;

% a)

n = 10;
x = linspace(-1, 3, n)';
fx = f(x);

coeff_polyfit = polyfit(x, fx, n-1);

x_plot = linspace(-1, 3, 1000);

p_equally = polyval(coeff_polyfit, x_plot);

figure
subplot(2,1,1)
plot(x_plot, f(x_plot), '-b', 'LineWidth', 2)
hold on, box on
plot(x_plot, p_equally, '-r', 'LineWidth',2)
plot(x, f(x), 'rx','LineWidth',2, 'MarkerSize',10)
set(gca,'FontSize',16)
set(gca,'LineWidth',1.5)
xlabel('x','FontSize',16)
ylabel('f(x)','FontSize',16)
legend('f', 'p_{equally}')

% b)

err_equally = abs(f(x_plot) - p_equally);

subplot(2,1,2)
hold on, box on
plot(x_plot, err_equally, '-r', 'LineWidth',2)
set(gca,'FontSize',16)
set(gca,'LineWidth',1.5)
xlabel('x','FontSize',16)
ylabel('|e(x)|','FontSize',16)
legend('p_{equally}')
```

Homework 8.3. Let $f(x) = x^5 + 1$ and $x_i = 0.2 \cdot i$, with $i = 0, 1, 2, 3$.

a. Let $\Pi_n f(x)$ be the interpolating polynomial of $f(x)$ at the nodes x_i. Compute $\Pi_n f(x)$ and provide an upper bound for $|E_n(x)| = |\Pi_n f(x) - f(x)|$.

b. Let $\Pi_1^H f(x)$ be the piecewise linear interpolation of $f(x)$ at the nodes x_i. Compute an upper bound for $\left|E_1^H(x)\right| = \left|\Pi_1^H f(x) - f(x)\right|$.

Solution Homework 8.3.

hw_8_3.m

```
clc
clear all
close all

f = @(x) x.^5 + 1;

x = [0 0.2 0.4 0.6];
p3 = polyfit(x, f(x), 3)

x_plot = linspace(0, 0.6, 1000);
p3_plot = polyval(p3, x_plot);

figure
plot(x, f(x), 'rx','LineWidth', 2, 'MarkerSize', 10)
hold on, box on
plot(x_plot, f(x_plot), 'b--', 'LineWidth', 2)
plot(x_plot, p3_plot, 'r-', 'LineWidth', 2)
axis([-0.05 0.65 0.99 1.09])
set(gca, 'FontSize', 16)
set(gca, 'LineWidth', 1.5)
xlabel('x','FontSize',16)
ylabel('f(x)','FontSize',16)
leg = legend('nodes', 'f', 'p3', 'Location', 'nw');

err = max(abs(f(x_plot) - p3_plot))
err_estimate = 72*(0.2)^4/(4*4)

% This is consistent with the estimate.

% b)

s3_plot = interp1(x,f(x),x_plot);
plot(x_plot, s3_plot, 'g-','LineWidth', 2)
leg = legend('nodes', 'f', 'p3', 's3', 'Location', 'nw');

err = max(abs(f(x_plot) - s3_plot))
err_estimate = 1/8*0.2^2*20*(0.6)^3

% This is consistent with the estimate.
```

Homework 8.4. Consider the following function

$$f(x) = \begin{cases} 0 & x \in [0, 1) \\ 1 & x \in [1, 2]. \end{cases}$$

First interpolate the function using the Lagrange polynomial with an increasing number of nodes. Then repeat the procedure using the piecewise linear and a least-squares approximation. What do you observe? Justify the answer.

Solution Homework 8.4.

hw_8_4.m

```
clc
clear all
close all

x_plot = linspace(0, 2, 1000);
f = @(x) 1*(x >= 1);

figure
plot(x_plot, f(x_plot), 'b-', 'LineWidth', 2)
hold on, box on
axis([-0.1 2.1 -1 2])
set(gca, 'FontSize', 16)
```

```
set(gca, 'LineWidth', 1.5)
xlabel('x','FontSize',16)
ylabel('f(x)','FontSize',16)

% Equispaced nodes

for n = [2, 4, 8, 16, 32]
  x = linspace(0, 2, n);

  coeff = polyfit(x, f(x), n-1);

  pn = polyval(coeff, x_plot);

  plot(x_plot, pn, 'r-', 'LineWidth',2)
  pause
end

% The result is very poor, for many reasons.
% The function to be interpolated is discontinuous,
% the interpolating polynomial has too many oscillations and
% the computation of the polynomial becomes ill-conditioned when
% the number of nodes increases.

% Piecewise linear polynomials

figure
plot(x_plot, f(x_plot), 'b-', 'LineWidth', 2)
hold on, box on
axis([-0.1 2.1 -1 2])
set(gca, 'FontSize', 16)
set(gca, 'LineWidth', 1.5)
xlabel('x','FontSize',16)
ylabel('f(x)','FontSize',16)

for n = [2, 4, 8, 16, 32]
  x = linspace(0, 2, n);

  s_plot = interp1(x,f(x),x_plot);
  plot(x_plot, s_plot, 'g-','LineWidth', 2)

  pause
end

% With linear splines the interpolation approximates the function
% with increasing precision, since for big n the slope at
% the discontinuity gets higher.

figure
plot(x_plot, f(x_plot), 'b-', 'LineWidth', 2)
hold on, box on
axis([-0.1 2.1 -1 2])
set(gca, 'FontSize', 16)
set(gca, 'LineWidth', 1.5)
xlabel('x','FontSize',16)
ylabel('f(x)','FontSize',16)

x = linspace(0, 2, 15);

for n = [2, 4, 8, 16, 32]

  v = polyfit(x,f(x), n);
  s_plot = polyval(v, x_plot);
  plot(x_plot, s_plot, 'g-','LineWidth', 2)

  pause
end

% Once again the result is very poor.
% The function to be interpolated is discontinuous,
% the interpolating polynomial has too many oscillations and
```

```
% the computation of the polynomial becomes ill-conditioned when
% the number of nodes increases because the interpolant tends to be smooth.
```