MAT 423: Solution homework 1 (08/27)

Ex 1. [4pts] Let $f(x) = x\ln(x+2)$ and $x = 0$.

a) ∘ $P_2(x) = f(0) + f'(0)x + f''(0)\frac{x^2}{2} = 0 + x\ln 2 + \frac{1}{2}x^2$.
∘ Thus, $P_2(\frac{1}{2}) = \frac{\ln 2}{2} + \frac{1}{8} \approx .472$. .5+.5 pt
An upper-bound for the error is given by:

$$|f(1/2) - P_2(1/2)| \leq \max_{\xi\in[0,\frac{1}{2}]} |f^{(3)}(\xi)| \frac{(\frac{1}{2})^3}{3!}$$ .5 pt

Since $f^{(3)}(\xi) = \frac{-x-6}{(x+2)^3}$, we deduce an upper-bound for $f^{(3)}(\xi)$ on $[0, \frac{1}{2}]$:

$$\max_{\xi\in[0,\frac{1}{2}]} |f^{(3)}(\xi)| \leq \frac{\max_{\xi\in[0,\frac{1}{2}]} |\xi+6|}{\min_{\xi\in[0,\frac{1}{2}]} |\xi+2|^3} = \frac{6.5}{8}.$$ .5 pt

Thus, the error is bounded by:

$$|f(1/2) - P_2(1/2)| \leq \frac{6.5}{8} \cdot \frac{(\frac{1}{2})^3}{3!} \approx .0169.$$

Remark. The exact error is actually: $|f(\frac{1}{2}) - P_2(\frac{1}{2})| \approx .0134$.
We can also find a better upper-bound for $|f^{(3)}(\xi)|$ on $[0, 1]$ by showing that the maximum is reached at $\xi = 0$ (thus $\max_{\xi\in[0,\frac{1}{2}]} |f^{(3)}(\xi)| \leq \frac{6}{8}$).

b) ∘ Using similar computations as in *a)*, we find that for $x \in [-\frac{1}{2}, \frac{1}{2}]$:

$$|f(x) - P_2(x)| \leq \max_{\xi\in[-\frac{1}{2},\frac{1}{2}]} |f^{(3)}(\xi)| \frac{|x|^3}{3!} \leq \frac{6.5}{8}\frac{|x|^3}{3!} = \mathbf{c}\,|x|^3,$$ .5+.5 pt

with $\mathbf{c} = \frac{6.5}{48}$.
∘ Using the approximation P_2, we have:

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} f(x)\,\mathrm{d}x \approx \int_{-\frac{1}{2}}^{\frac{1}{2}} P_2(x)\,\mathrm{d}x = \left[\ln 2\,\frac{x^2}{2} + \frac{x^3}{6}\right]_{-\frac{1}{2}}^{\frac{1}{2}}$$ .5 pt

$$= 0 + 2\frac{(\frac{1}{2})^3}{6} = \frac{1}{24} \approx 4.17 \cdot 10^{-2}.$$

An upper-bound for the error is given by:

$$\begin{aligned}\left|\int_{-\frac{1}{2}}^{\frac{1}{2}} f(x)\,\mathrm{d}x - \int_{-\frac{1}{2}}^{\frac{1}{2}} P_2(x)\,\mathrm{d}x\right| &\leq \int_{-\frac{1}{2}}^{\frac{1}{2}} |f(x)-P_2(x)|\,\mathrm{d}x \leq \int_{-\frac{1}{2}}^{\frac{1}{2}} \mathbf{c}|x|^3\,\mathrm{d}x \\ &= 2\mathbf{c}\int_0^{\frac{1}{2}} x^3\,\mathrm{d}x = 2\mathbf{c}\left.\frac{x^4}{4}\right|_0^{\frac{1}{2}} = \frac{2\mathbf{c}}{4\cdot 2^4} \approx 4.23\cdot 10^{-3}\end{aligned}$$

.5 pt

Ex 2. [3pts] Let $f(x) = \mathrm{e}^{-x}$, $x_0 = 0$ and $h = .1$.
The Taylor approximation of f at $x_0 + h$ of order 4 gives:

$$P_4(h) = 1 - h + \frac{h^2}{2} - \frac{h^3}{3!} + \frac{h^4}{4!} \approx .9048.$$

1.5 pt

An upper-bound for the error is given by:

$$|f(h) - P_4(h)| \leq \max_{[0,.1]} |f^{(5)}(\xi)|\,\frac{|h|^5}{5!} = \max_{[0,.1]} |\mathrm{e}^{-\xi}|\,\frac{h^5}{5!} \leq 1\cdot\frac{h^5}{5!} \approx 8.33\cdot 10^{-8}.$$

1.5 pt

Remark. We could also use a 5^{th} order Taylor approximation, but 4^{th} order is enough.

Ex 3. [Lipschitz condition]

a) ∘ We give 2 methods to prove the statement: f *Lipschitz implies* f *continuous.*

- [with sequences] Take x_* in $[a,b]$ and $x_n \overset{n\to\infty}{\longrightarrow} x_*$. We have:

$$|f(x_n) - f(x_*)| \leq \mathbf{L}|x_n - x_*| \overset{n\to\infty}{\longrightarrow} 0.$$

  Thus, f continuous at x_*. Since this is true for any x_* in $[a,b]$, f is continuous on $[a,b]$.

- [with (δ,ε)] Take x_* in $[a,b]$ and $\varepsilon > 0$. Consider $\delta = \varepsilon/\mathbf{L}$. We have for any $|x - x_*| \leq \delta$:

$$|f(x) - f(x_*)| \leq \mathbf{L}|x - x_*| \leq \mathbf{L}\delta \leq \varepsilon.$$

  Thus, f continuous at x_*. We deduce that f continuous on $[a,b]$.

∘ The function $f(x) = \sqrt{x}$ is continuous on $[0,1]$ but not Lipschitz (because f' *'explodes'* at $x = 0$).

b) We suppose that f is a C^1 function on $[a,b]$.
∘ To show that f is Lipschitz, we use the mean-value theorem: for any x, y in $[a,b]$ there exists $c \in [x,y]$ such that:

$$|f(x) - f(y)| = |f'(c)||x - y|.$$

Now take $\mathbf{L} = \max_{c\in[a,b]} |f'(c)|$ ($\mathbf{L}$ exists since f' is continuous on a bounded and close interval). We deduce from the previous equation that:

$$|f(x) - f(y)| \leq \mathbf{L}|x - y|.$$

Since this inequality is true for any x, y in $[a, b]$, we deduce that f is $\mathbf{L}$-Lipschitz on $[a, b]$.
∘ Conversely, if f $\mathbf{L}$-Lipschitz, then:

$$|f'(x)| = \left|\lim_{h\to 0} \frac{f(x+h) - f(x)}{h}\right| \leq \lim_{h\to 0} \frac{|f(x+h) - f(x)|}{|h|} \leq \lim_{h\to 0} \frac{\mathbf{L}\cdot|h|}{|h|} = \mathbf{L}.$$

Thus, $|f'(x)| \leq \mathbf{L}$.

Ex 4. [3pts]

We consider $f(x) = x^2 - 3$, $a = 1$ and $b = 2$. Thus, $x_* = \sqrt{3}$ is in between a and b. Since $f(a) = -2 < 0$ and $f(b) = 1 > 0$, we can apply the bisection method. .5+.5 pt

Moreover, the sequence $\{x_n\}_n$ satisfies:

$$|x_n - x_*| \leq \left(\frac{1}{2}\right)^{n+1} |b - a| = \left(\frac{1}{2}\right)^{n+1}.$$ 1 pt

To have an accuracy of 10^{-4}, a sufficient condition is:

$$\left(\frac{1}{2}\right)^{n+1} \leq 10^{-4} \Rightarrow -(n+1)\ln 2 \leq -4\ln 10 \Rightarrow n \geq \frac{4\ln 10}{\ln 2} - 1 \approx 12.28.$$ .5 pt

Thus, we need 13 iterations to obtain an accuracy of at least 10^{-4}.
Using the script of **Ex. 5**, we find: $x_{13} = 1.7320$. .5 pt

Remark. We can confirm the calculation by estimating the exact error:

$$|\sqrt{3} - x_{13}| = 5.6179 \times 10^{-05} \leq 10^{-4}.$$

Ex 5. See the scripts below.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%     Bisection Method  (MATLAB/OCTAVE)     %%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% function f and interval [a,b]
f = @(x) x^2-3;
a = 1;
b = 2;
%% parameter simulation (number of iterations)
N = 13;
%% Saving
saveX = zeros(1,N+1);
%% Test:
%%   f(a) and f(b) must have different sign
if ( (f(a)*f(b)) > 0 )
  fprintf(['-----------------------------------------\n',...
           '  Exit program\n',...
           '  f(a) and f(b) must have different signs.\n',...
           '-----------------------------------------\n'])
  return;
end
%%----------------------------------------------------%%
%%                        Loop                        %%
%%----------------------------------------------------%%
%% initialization
a0 = a;
b0 = b;
x  = (a0+b0)/2;
saveX(1) = x;
%% loop
for i=1:N
  if (f(a0)*f(x)<0)
    b0 = x;
  else
    a0 = x;
  end
  %% save
  x = (a0+b0)/2;
  saveX(i+1) = x;
end
%% Estimation error: x* = last value of x_n
xS = saveX(end);
errorBisection = sqrt( (saveX-xS).^2);
fprintf(['---------------------------------------\n',...
         ' Bisection method after %i iterations:\n',...
```

```
        '   x = %f\n',...
        '  Max. error:\n',...
        '   error < %e\n',...
        '---------------------------------------\n',...
        ],N,xS,abs(b-a)*(1/2)^(N+1))
%%----------------------------------------------------%%
%%----------------------------------------------------%%

%%------------ plot
figure(1);
plot(0:N,saveX,'-o')
xlabel('number of iterations (n)')
ylabel('x')
title('Bisection method: convergence of x_n')
figure(2);
semilogy(0:(N-1),errorBisection(1:N),'-or','linewidth',3)
axis([0 N 10^(-17) 1])
xlabel('number of iterations (n)')
ylabel('error: |x_n-x_*|')
title('Error of the bisection method')
```

```
######################################################
#####         Bisection Method (PYTHON)          #####
######################################################
import numpy as np
import matplotlib.pyplot as plt
import sys

## function f and interval [a,b]
def f(x):
    return x**2-3
a = 1.0
b = 2.0
## parameter simulation (number of iterations)
N = 13
## Saving
saveX = np.zeros(N+1)
## Test:
##   f(a) and f(b) must have different sign
if ( (f(a)*f(b)) > 0 ):
    print ('----------------------------------------- \n' +
           '  Exit program\n' +
           '  f(a) and f(b) must have different signs.\n' +
           '-----------------------------------------')
    sys.exit()
##----------------------------------------------------##
##                        Loop                        ##
##----------------------------------------------------##
## initialization
a0 = a
b0 = b
x  = (a0+b0)/2
saveX[0] = x
## loop
for i in range(1,N+1):
    if (f(a0)*f(x)<0):
        b0 = x
    else:
        a0 = x
    ## save
    x = (a0+b0)/2
    saveX[i] = x

## Estimation error: x* = last value of x_n
xS = saveX[-1]
```

```python
errorBisection = np.sqrt( (saveX-xS)**2);
print ('---------------------------------------\n' +
       ' Bisection method after {:,} iterations:\n'.format(N) + # or {:05d}
       '   x =  {:,}\n'.format(xS) +                          # or {0:5,.9f}
       '  Max. error:\n' +
       '   error < {:,}\n'.format(np.abs(b-a)*(1/2)**(N+1)) +
       '---------------------------------------\n')
##----------------------------------------------------##
##----------------------------------------------------##

##------------ plot
# convergence x_n
plt.plot(range(N+1), saveX, '--o', lw=2, color='blue')
plt.xlabel('number of iterations (n)')
plt.ylabel('x')
plt.title('Bisection method: convergence of x_n')
# convergence error
plt.figure()                        # new figure
plt.plot(range(N), errorBisection[0:N], '-o', lw=2, color='red')
plt.yscale('log')
plt.xlabel('number of iterations (n)')
plt.ylabel('error: |x_n-x_*|')
plt.title('Error of the bisection method')
plt.show()                          # or plt.show(block=False)
```

```julia
####################################################
#####        Bisection Method (JULIA)         #####
####################################################
## function f and interval [a,b]
f(x) = x^2-3
a = 1
b = 2
## parameter simulation (number of iterations)
N = 13
## Saving
saveX = zeros(N+1)
## Test:
##   f(a) and f(b) must have different sign
if ( (f(a)*f(b)) > 0 )
    print("-----------------------------------------\n",
          "  Exit program\n",
          "  f(a) and f(b) must have different signs.\n",
          "-----------------------------------------\n")
    return
end

##----------------------------------------------------##
##                        Loop                        ##
##----------------------------------------------------##
## initialization
a0 = a
b0 = b
x  = (a0+b0)/2
saveX[1] = x
## loop
for i=1:N
  if (f(a0)*f(x)<0)
    b0 = x
  else
    a0 = x
  end
  ## save
  x = (a0+b0)/2
  saveX[i+1] = x
end
## Estimation error: x* = last value of x_n
xS = saveX[end]
errorBisection = sqrt.( (saveX .- xS).^2)
theError = abs(b-a)*(1/2)^(N+1)
```

```
print("---------------------------------------\n",
      " Bisection method after $N iterations:\n",
      "   x = $xS\n",
      "  Max. error:\n",
      "   error < $theError\n",
      "---------------------------------------\n")
##----------------------------------------------------##
##----------------------------------------------------##

##------------ plot
using PyPlot
figure(1)
plot(0:N,saveX,linestyle="-.")
xlabel("number of iterations (n)")
ylabel("x")
title("Bisection method: convergence of x_n")
figure(2)
semilogy(0:(N-1),errorBisection[1:N],linestyle="-.",color="r",linewidth=2)
axis((0,N,10.0^(-17),1))
xlabel("number of iterations (n)")
ylabel("error: |x_n-x_*|")
title("Error of the bisection method")
```