MTH5110 **Introduction to Numerical Computing**

Final exam (marking scheme)

10:00am Monday 18th May, 2015
Duration: 2 hours

Name:
Student ID:

Save the worksheet when you have entered your name and ID (and save the worksheet at regular intervals, say every 5 minutes, to avoid data loss).

When you have finished the exam save the file and send the file from your college email account by email as an attachment to:

w.just@qmul.ac.uk

with subject field "<student ID> mth5110 exam".

This is an OPEN BOOK exam, all questions count.

permitted:
- any printed material, e.g. books
- any handwritten notes
- photocopies of any kind
- use of a computer (Maple, google, wikipedia, ...)

prohibited:
- using electronic communication devices (e.g. mobile phones, email - except to submit the exam - , twitter,...)
- sharing material with other students

YOU ARE NOT PERMITTED TO START READING THE SUBSEQUENT PARTS OF THIS QUESTION PAPER UNTIL INSTRUCTED TO DO SO BY AN INVIGILATOR.

STOP SCROLLING !!!

YOU ARE NOT PERMITTED TO START READING THE SUBSEQUENT PARTS OF THIS QUESTION PAPER UNTIL INSTRUCTED TO DO SO BY AN INVIGILATOR.

Problem 1

a) **[8 marks]**

Evaluate the expression 114/9 using arithmetic with 2 digits in the mantissa. Explain briefly the outcome.

```
> evalf[2](evalf[2](114)/evalf[2](9));
```

$$12. \tag{1}$$

each evalf (2+2+2 marks) (one not needed)
result mathematically incorrect as 114 has 3 significant digits (2 marks)

b) **[8 marks]**

When computing $\sqrt{\sqrt{2}\,\sqrt{2}-2}$ in Maple we seem to obtain a complex number

```
> sqrt(sqrt(2.0)*sqrt(2.0)-2.0);
```

$$0.00003162277660\, I \tag{2}$$

Explain this finding.

Floating point arithmetics involves truncation errors (4 marks).
sqrt(2.0) givesa result which is slightly too small (4 marks)

c) **[12 marks]**

Trace the following piece of Maple code

```
> for k from 2 to 4 do
     k:=(k-5)*(k-2):
  end do:
```

2 marks for each output, 4 marks for the last one (2+2+2+2+4)

```
> for k from 2to 4 do
     print(k):
     k:=(k-5)*(k-2):
     print(k):
  end do:
  print(k):
```

$$2$$

0
1
4
5 (3)

Problem 2
It can be shown that the integrals

$$I_n = \int_1^{\infty} x^{-3n} e^{-x^3} \, dx$$

obey the recurrence relation

$$I_n = \frac{1}{3e} - \left(n + \frac{2}{3}\right) I_{n+1}$$

a) **[8 marks]**

State with a reason whether the forward iteration is stable or unstable,
and state with a reason whether the backward iteration is stable or unstable.

Forward iteration is stable (2 marks) since 1/(n-2/3)<1 for large n (2 marks).
Backward iteration is unstable (2 marks) since n-2/3 >1 for large n (2 marks).

b) **[8 marks]**
Using a stable iteration scheme compute the value of I_{10} with about three significant digits.

Some seed (2 marks)
Correct iteration formula (2 marks)
Some loop with some limits(2 marks)
Consistent output (2 marks)

```
> In:=1.0;
  for n from 0 to 9 do
    In:=(exp(-1.0)/3.0-In)/(n+2.0/3.0);
  end do;
```

$$In := 1.0$$
$$In := -1.316060279$$
$$In := 0.8632120552$$
$$In := -0.2777195905$$
$$In := 0.1091852921$$
$$In := 0.002880254636$$
$$In := 0.02113168690$$
$$In := 0.01522421902$$

$$In := 0.01400899062$$
$$In := 0.01253278728$$
$$In := 0.01138900273 \quad (4)$$

c) **[8 marks]**

Using error propagation estimate the absolute error of the result you have computed in part b).

>

Estimate of seed error (2 marks)
Error propagation for each step (4 marks)
Final estimate (2 marks)

$$\prod_{n=0}^{9} \frac{1}{\left(n+\frac{2}{3}\right)} < \frac{1}{9!} = \frac{1}{362880}$$

Problem 3

a) **[6 marks]**

State a Newton Raphson map for solving the equation $e^x = x$.

>

General NR map (2 marks)
Rearrange equation for rhs 0 (2 marks)
Correct map (2 marks)

```
> x-(exp(x)-x)/(diff(exp(x)-x,x));
```

$$x - \frac{e^x - x}{e^x - 1} \quad (5)$$

b) **[10 marks]**

The inverse function of the function $f(x) = x\,e^x$ is called the Lambert W-function $W(y)$. Write a procedure which computes the value of $W(y)$ using a Newton Raphson scheme. Your procedure should have a single input, the value of y, and it should return the value of $W(y)$ with at least 3 significant digits.

>

NR code from notes (2 marks)

```
> my_lambert:=proc(y)
    # adjust input                (2marks)
   local xold,xnew,phi,f,tol;
    # tolerance in code           (2 marks)
   tol:=10.0^(-3);
    # adjust seed                 (2 marks)
   xnew:=y;
    # fictitious old value for x to start the loop
   xold:=xnew-tol-1;
```

```
 # NR map for f(x)-y=0                    (2 marks)
f:=x->x*exp(x)-y;
phi:=x->x-f(x)/D(f)(x);
 # loop with termination condition
while abs(xnew-xold)>tol do
    # transcribe x value
  xold:=xnew;
    # NR iteration
  xnew:=phi(xold);
end do;
 # output
return xnew;
end proc;
```

$my\_lambert := \textbf{proc}(y)$ **(6)**

$\quad \textbf{local}\ xold, xnew, \phi, f, tol;$

$\quad tol := 10.0\hat{}(-3);$

$\quad xnew := y;$

$\quad xold := xnew - tol - 1;$

$\quad f := x \to x * \exp(x) - y;$

$\quad \phi := x \to x - f(x) / \mathrm{D}(f)(x);$

$\quad \textbf{while}\ tol < \mathrm{abs}(xnew - xold)\ \textbf{do}\ xold := xnew;\ xnew := \phi(xold)\ \textbf{end do};$

$\quad \textbf{return}\ xnew$

$\textbf{end proc}$

c) **[6 marks]**

Use your procedure to compute the value of $W(2)\,\mathrm{e}^{W(2)}$. What is the absolute error of your result?

```
>
```

evaluation (2 marks)
exact result 2 (2 marks)
absolute error (2 marks)

```
> res:=my_lambert(2.0)*exp(my_lambert(2.0));
```

$$res := 2.000000529 \qquad \textbf{(7)}$$

```
> res-2.0;
```

$$5.29\ 10^{-7} \qquad \textbf{(8)}$$

Problem 4

Consider the definite integral

$$I = \int_0^{\frac{1}{\pi}} \sqrt{x}\cos\left(\frac{1}{x}\right)\,\mathrm{d}x \quad .$$

a) **[10 marks]**

Write a Maple procedure which computes a numerical approximation of the integral using the trapezodial rule. Your procedure should have a single input, the number of subintervals n. Your procedure should return the numerical approximation of I and an estimate of the absolute error of the result (you may use any Maple command to derive the error estimate).

```
>
```

Algorithm from notes (2 marks)
Kernel in code (2 marks)
Correct limits in code (2 marks)
Correct evaluation at endpoint 0 (2 marks)
Absolute error in output (2 marks)

```
> my_int:=proc(n)
      local h,s,a,b,f;
      f:=x->sqrt(x)*cos(1.0/x);
      a:=0.0;
      b:=1.0/evalf(Pi);
       # stepsize
      h:=(b-a)/n;
       # boundary points
      s:=0.0-sqrt(1.0/evalf(Pi));
       # sum over nodes
      s:=s+2*add(f(a+k*h),k=1..n-1);
       # output  and absolute error
      return s*h/2,abs(evalf(int(sqrt(x)*cos(1/x),x=0..1/Pi))-s*h/2)
  ;
  end proc;
```

$my\_int := \mathbf{proc}(n)$ **(9)**

$\quad \mathbf{local}\ h, s, a, b, f;$

$\quad f := x \rightarrow \mathrm{sqrt}(x) * \cos(1.0 / x);$

$\quad a := 0.;$

$\quad b := 1.0 / evalf(\pi);$

$\quad h := (b - a) / n;$

$\quad s := 0. - \mathrm{sqrt}(1.0 / evalf(\pi));$

$\quad s := s + 2 * add(f(a + k * h), k = 1 .. n - 1);$

$\quad \mathbf{return}\ 1/2 * s * h, \mathrm{abs}(evalf(int(\mathrm{sqrt}(x) * \cos(1/x), x = 0 .. 1/\pi)) - 1/2 * s * h)$

$\mathbf{end\ proc}$

b) **[8 marks]**

Produce a plot where you show the absolute error vs. n for $n = 2, 3, 4, ..., 200$.
How does the absolute error depend on n?

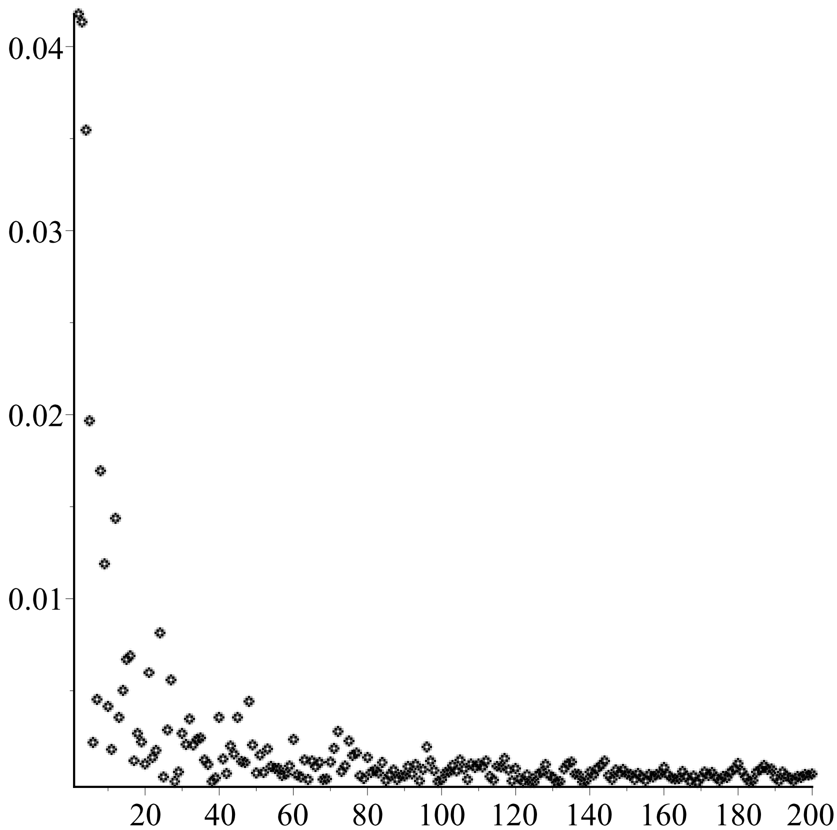
```
>
```

Create list/sequence, e.g. in a loop which contains correct data format, n and error (2 marks)

```
> lst:=NULL:
  for k from 2 to 200 do
     lst:=lst,[k,my_int(k)[2]];
  end do:
```

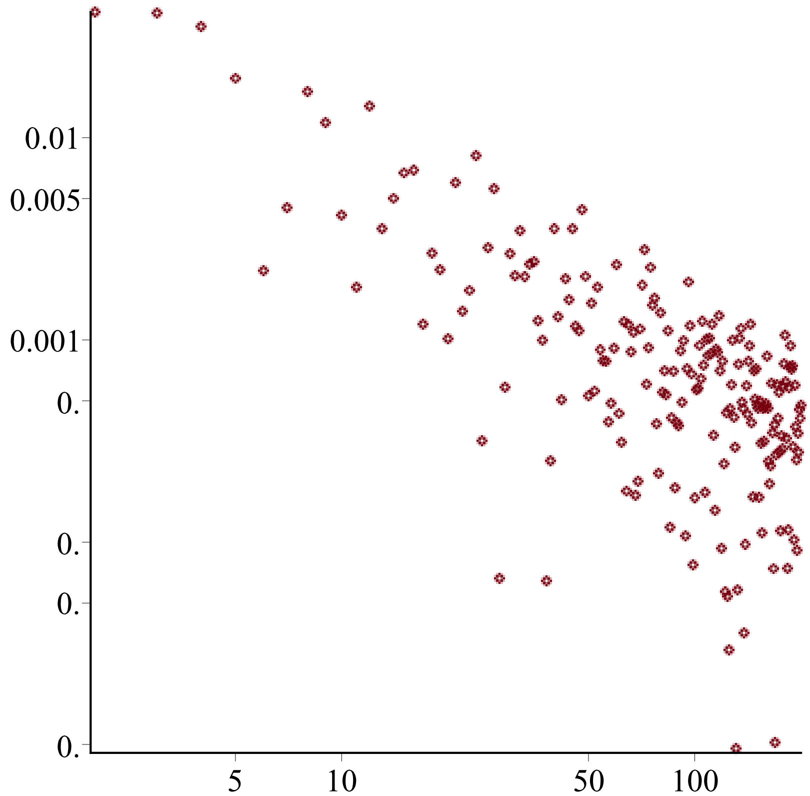
Plot using e.g. listplot (2 marks)

```
> plots[listplot]([lst],style=point);
```



Decays like n^{-1} (4 marks), e.g. logarithmic plot (kernel not smooth and strongly oscillating - full marks for this statement as well).

```
> plots[loglogplot]([lst],style=point);
```



c) **[8 marks]**
Use your procedure to determine the value of I with at least 6 significant digits.

>

At about 200000 nodes are needed, correct input (4 marks)
Run the programm (4 marks)

```
> my_int(200000);
```

$$-0.02316479862,\ 7.0459\ 10^{-7} \quad (10)$$