MTH5110 **Introduction to Numerical Computing**

Final exam

??:??pm ????day ??th May, 2014
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 attachement 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) **[10 marks]**

We want to evaluate 1.25/7 using Maple. Explain briefly the outcome of each of the following Maple commands. Why do they differ ?
Which command gives the result when using arithmetic with 2 digits in the mantissa ?

```
> 1.25/7.0;
```

$$0.1785714286 \quad (1)$$

```
> evalf[2](1.25/7.0);
```

$$0.18 \quad (2)$$

```
> evalf[2](1.25)/evalf[2](7.0);
```

$$0.1714285714 \quad (3)$$

```
> evalf[2](evalf[2](1.25)/evalf[2](7.0));
```

$$0.17 \quad (4)$$

(bookwork)

(i) standard 10 digit arithmetic (2 marks)
(ii) 2 digit display for 10 digit arithmetic (2 marks)
(iii) 10 digit arithmetic for 2 digit floats (2 marks)
(iv) 2 digit display for arithmetic with 2 digt floats, 2 digits mantissa (2+2 marks)

b) **[6 marks]**

Compute the value of 1009-9 when using arithmetic with 2 digit precision.

(bookwork)

```
> evalf[2](evalf[2](1009)-evalf[2](9));
```

$$990. \quad (5)$$

(2+2+2 marks, for each evalf, or similar)

c) **[12 marks]**

The sequence of Pell numbers is defined by the recursion $P_0 = 0$, $P_1 = 1$, and $P_{n+2} = 2\,P_{n+1} + P_n$ for $n \geq 0$. Write a Maple procedure which for a given input n returns the value of the nth Pell number P_n.

Your procedure should display an error message if an inconsistent value for n has been used. Use your algorithm to compute P_{23}.

(unseen)

Fibbonacci algorithm from problem sheet, or similar (consistent loop and update) (2 marks)
Correct output (no list) (2 marks)
Cases n=0 and n=1 (2 marks)
Adjustment of recursion relation (2 marks)
Error message (noninteger 1 mark, negative 1 mark, alternatively n::nonnegint)

```
> my_pell:=proc(n::integer)
    local a,b,c,k;
    if (n<0) then
      error "n<0"
    end if;
    a:=0;
    b:=1;
    if n=0 then
       c:=0;
    elif n=1 then
       c:=1;
    end if;
    for k from 2 to n do
      c:=a+2*b;
       a:=b;
       b:=c;
    end do;
    return c;
  end proc;
```

$my\_pell := \mathbf{proc}(n{::}integer)$ **(6)**

local a, b, c, k;

if $n < 0$ **then error** "n<0" **end if**;

$a := 0$;

$b := 1$;

if $n = 0$ **then** $c := 0$ **elif** $n = 1$ **then** $c := 1$ **end if**;

for k **from** 2 **to** n **do** $c := a + 2 * b$; $a := b$; $b := c$ **end do**;

return c

end proc

```
> my_pell(23);
```

225058681 **(7)**

Run the procedure (2 marks)

Problem 2

a) **[8 marks]**

A bisection algorithm for which the initial interval has been specified as $[-2.5, 1]$ returns the value -0.3125. How many bisection steps have been performed by the algorithm and what can you say about the size of the absolute error of the result ?

(bookwork)

Midpoints via Maple

```
> (1-2.5)/2;
```

-0.7500000000 (8)

```
> (1-0.75)/2;
```

0.1250000000 (9)

```
> (0.125-0.75)/2;
```

-0.3125000000 (10)

2(3) bisections (4 marks)

```
> (0.125+0.75)/2;
```

0.4375000000 (11)

absolute error less than 0.4375 (4 marks) (2 marks for statement: half interval width)

b) **[10 marks]**

Write a Maple procedure which computes the zero of a function using bisection. Your procedure should have as input the function, the endpoints of the initial interval, a tolerance, and a maximal number of iteration steps. The procedure should terminate if the length of the interval and the absolute value of the function evaluated at the endpoints is smaller than the tolerance. Your procedure should return the estimate of the zero and the number of iteration steps. An error message should be displayed if the number of steps exceeds the maximum specified in the input.

(unseen)

Bisection algorithm from notes (2 marks)
Adjust input (2 marks)
Add counter (2 marks)
Change loop control statement/output (2 marks)
Error message (2 marks)

```
> bisection:=proc(f,a,b,tol,imax)

    local low,up,mid,k;

    low:=a;
    up:=b;

    if f(low)*f(up)>0 then
       error "f(a) f(b) >0";
    end if;

    for k from 1 to imax do

      mid:=(low+up)/2;

      if f(low)*f(mid)>0 then
         low:=mid;
      else
         up:=mid;
```

```
        end if;

       if up-low<tol and abs(f(up))<tol and abs(f(low))<tol then
          return mid,k;
        end if;

     end do;

     error "no convergence";

   end proc;
```

bisection := **proc**(f, a, b, tol, imax) **(12)**

```
    local low, up, mid, k;
    low := a;
    up := b;
    if 0 < f(low) * f(up) then error "f(a) f(b) >0" end if;
    for k to imax do
        mid := 1 / 2 * low + 1 / 2 * up;
        if 0 < f(low) * f(mid) then low := mid else up := mid end if;
        if up - low < tol and abs(f(up)) < tol and abs(f(low)) < tol then
            return mid, k
        end if
    end do;
    error "no convergence"
end proc
```

c) **[8 marks]**

Use your procedure to compute a solution to the equation $\tan(x) = x$ in the interval $[1, 2]$. Explain your findings.

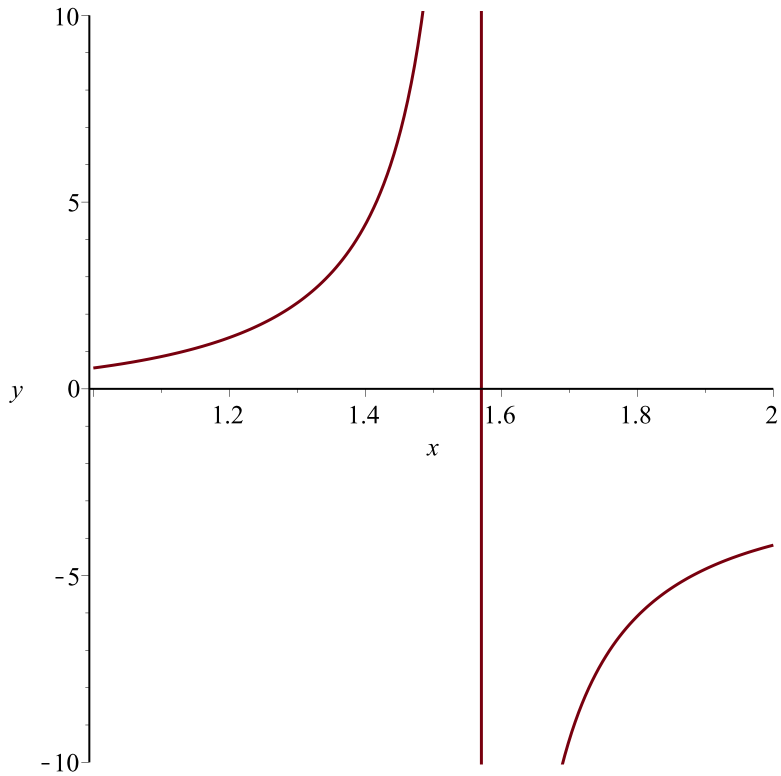
(unseen)

Correct input function (2 marks)
Sensible values for tol and imax (2 marks) (say tol<0.1 and imax>10)

```
> bisection(x->tan(x)-x,1.0,2.0,0.0001,100);
Error, (in bisection) no convergence
```

Function has a pole (4 marks), e.g. (2 marks just for plot only)

```
> plot(tan(x)-x,x=1.0..2.0,y=-10..10);
```



Problem 3

a) **[10 marks]**

State a Newton Raphson map for solving the equation $|x| = 1 - \cos(x)$.

(unseen)

General NR map (2 marks)
Square equation (remove kink) (4 marks)
Rearrange equation for rhs 0 (2 marks)
Correct map (2 marks)

```
> x-(x^2-(1-cos(x))^2)/(diff(x^2-(1-cos(x))^2,x));
```

$$x - \frac{x^2 - (1 - \cos(x))^2}{2x - 2(1 - \cos(x))\sin(x)} \quad (13)$$

b) **[12 marks]**

Given a real number ε, $0 \le \varepsilon < 1$, we seek to write a procedure which computes the inverse function $f^{-1}(y)$ of the function $f(x) = x - \varepsilon \sin(x)$ (the so called Kepler's equation) evaluated at y, using a Newton Raphson scheme. Your procedure should have two inputs, the values of ε and y. The procedure should return the numerical value of $f^{-1}(y)$ with 5 significant digits. Use your procedure to compute $f^{-1}(-3.8)$ for $\varepsilon = 0.7$.

(unseen)

NR code from notes (2 marks)

```
> my_nr:=proc(eps,y)
    # adjust input                          (2marks)
    local xold,xnew,phi,f,tol;
    # tolerance in code                     (2 marks)
    tol:=10.0^(-5);
    # 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-eps*sin(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_nr := **proc**(*eps*, *y*) **(14)**

local *xold*, *xnew*, phi, *f*, *tol*;

$tol := 10.0\hat{}(-5)$;

$xnew := y$;

$xold := xnew - tol - 1$;

$f := x \to x - eps * \sin(x) - y$;

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

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

return *xnew*

end proc

Run the procedure (2 marks)

```
> my_nr(0.7,-3.8);
```

$$-3.532974018 \qquad (15)$$

Problem 4

Consider the definite integral

$$\frac{\pi}{2} = \int_{-1}^{1} \sqrt{1 - x^2}\, dx \quad .$$

a) **[8 marks]**

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

(unseen)

Algorithm from notes (2 marks)
Kernel in code (factor 2) (2 marks)
Limits in code (2 marks)
Absolute error in output (2 marks)

```
> my_pi:=proc(n)
    local h,s,a,b,f;
    f:=x->2.0*sqrt(1-x^2);
    a:=-1.0;
    b:=1.0;
    # stepsize
    h:=(b-a)/n;
    # boundary points
    s:=f(a)+f(b);
    # 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(Pi)-s*h/2);
  end proc;
```

```
my_pi := proc(n)                                                    (16)
    local h, s, a, b, f;
    f := x→2.0 * sqrt(1 − x^2);
    a := −1.0;
    b := 1.0;
    h := (b − a) / n;
    s := f(a) + f(b);
    s := s + 2 * add(f(a + k * h), k = 1 ..n − 1);
    return 1 / 2 * s * h, abs(evalf(Pi) − 1 / 2 * s * h)
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?

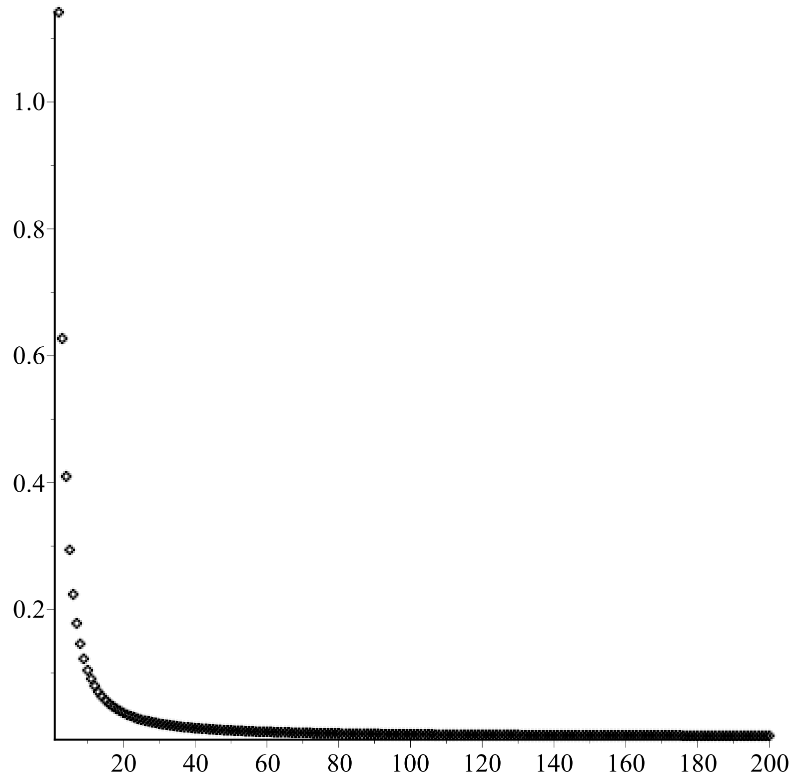
(unseen)

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_pi(k)[2]];
  end do:
```

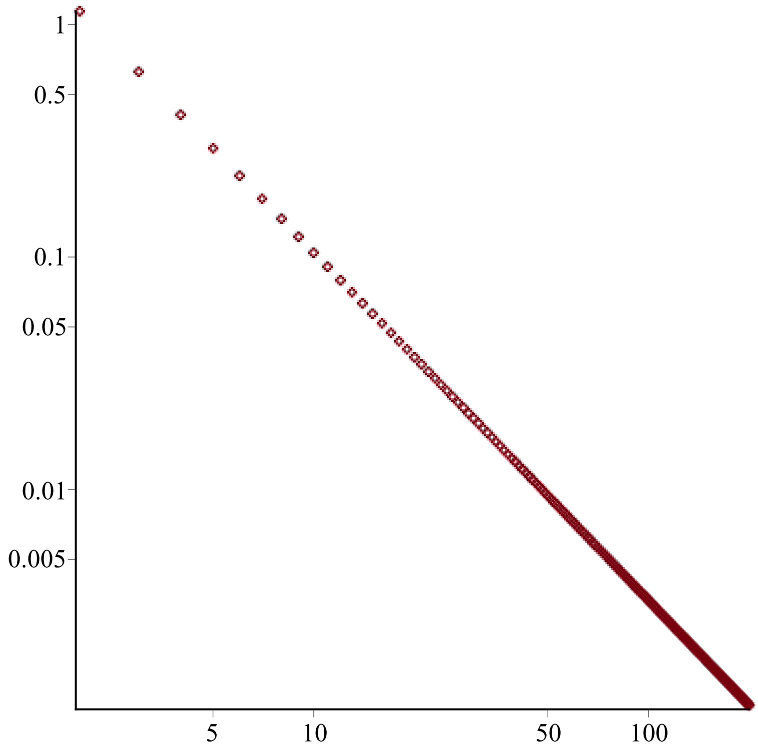
Plot using e.g. listplot (2 marks)

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



Decays like $n^{-1.5}$ (4 marks), e.g. logarithmic plot (kernel not smooth - full marks for this statement as well).

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



c) **[8 marks]**

Use your procedure to determine the value of π with at least 6 significant digits.

(bookwork)

At about 10000 nodes are fine, correct input (4 marks)
Run the programm (4 marks)

```
> my_pi(10000);
```

3.141589328, 0.000003326 **(17)**