

Name: \_\_\_\_\_

Student number: \_\_\_\_\_

Introduction to Scientific Computing  
PSCB57  
Final Exam

Professor Hanno Rein  
University of Toronto, Scarborough

Friday, 19th December 2014, SW128, 9-11am

	Points	Max Points
Question 1		4
Question 2		5
Question 3		3
Question 4		3
Question 5		3
Question 6		4
Question 7		3
Question 8		4
Question 9		8
Question 10		4
Question 11		4
Bonus Question		(4 Bonus)
Total		45

- No aid sheets, books or other notes are allowed.
- All electronic devices must be stored together with your belongings at the back of the room.
- Write your answers on the question sheet. If you need more paper raise your hand.
- The length of the white space for each question gives you an idea of the expected length and complexity of the answer.
- The University of Toronto's Code of Behaviour on Academic Matters applies to all University of Toronto Scarborough students. The Code prohibits all forms of academic dishonesty including, but not limited to, cheating, plagiarism, and the use of unauthorized aids. Students violating the Code may be subject to penalties up to and including suspension or expulsion from the University.



## Question 1

4 Points

What is ...

git? \_\_\_\_\_

linux? \_\_\_\_\_

python? \_\_\_\_\_

vi? \_\_\_\_\_

## Question 2

5 Points

We work in IEEE754 double floating point precision. Circle all numbers that can be represented exactly.

1.1          3.0          0.0          -1e-1          -1e+1

0.125          0.1024          1e+1024          0.5          10.25

## Question 3

3 Points

We still work in IEEE754 double floating point precision. Calculate the following expressions.

$2 + 4 =$  \_\_\_\_\_

$1e+308 + 1e+308 =$  \_\_\_\_\_

$((1e+18 + 1.125) - 1e18) + 1e-18 =$  \_\_\_\_\_

## Question 4

3 Points

---

```
SET 1 r8
PRINT r8
SET 1 r5
PRINT r5
ADD r8 r5 r2
COPY r5 r8
PRINT r2
COPY r2 r5
SET -5 r3
JUMP r3
```

---

What is the above assembler program doing? (The syntax of assembler commands can be found on the last page.)

---

What is the last number that is being printed before the program terminates?

---

## Question 5

3 Points

Next to every YouTube video is a counter that shows how many people watched the video. On December 1st 2014, the counter for the video *Psy - Gangnam Style* stopped working after 2 147 483 647 people watched it. Explain what happened:

---

---

---

Which number was shown to the next person watching it?

---

---

What could Google do to prevent this bug in the future?

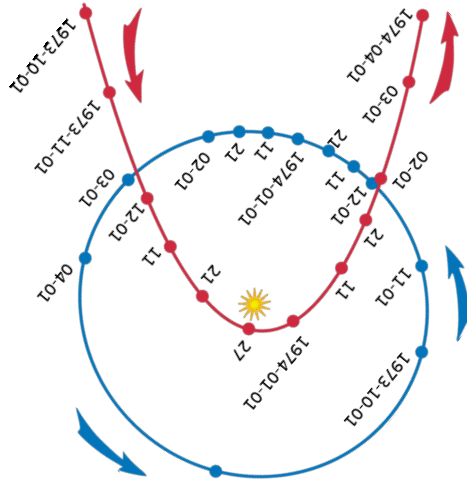
---

---

---

### Question 6

4 Points



The image shows the paths of the comet Kohoutek and the Earth. The comet is on a nearly parabolic orbit. If the coordinate system is chosen appropriately (you don't have to worry about this), its path can be described as a quadratic function:

$$f(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2.$$

Astronomers have measured the position of the comet 4 times. Their measurements are the following pairs:  $(-2, 3)$ ,  $(-1, 4)$ ,  $(0.1, 1)$  and  $(2, 2)$ . We want to fit the function  $f(x)$  and find the coefficients  $a_0$ ,  $a_1$  and  $a_2$  using a linear least square fit. Fill in the missing numbers in the following system of equations:

$$\begin{pmatrix} \_ \\ \_ \\ \_ \\ \_ \end{pmatrix} = \underbrace{\begin{pmatrix} \_ & \_ & \_ \\ \_ & \_ & \_ \\ \_ & \_ & \_ \\ \_ & \_ & \_ \end{pmatrix}}_{\equiv C} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} + \begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{pmatrix}$$

The  $e_i$  values in the above equation are the residuals (errors). In the lecture, we minimized the  $e_i$ s to arrive at the following linear system of equations

$$b = A \cdot a$$

where we introduced the matrix  $A = C^T \cdot C$  and the vector  $b = C^T \cdot y$ . Describe, qualitatively in equations or words, how to solve the system of linear equations  $b = A \cdot a$  using the LU decomposition.

---



---



---



---



---

## Question 7

3 Points

Find the Lagrange polynomial that interpolates the following points (no need to simplify it).

(0, 3)  
( $\sqrt{2}$ , 4)  
(6, 7)  
(8,  $\pi$ )

What is the order of the polynomial? \_\_\_\_\_

## Question 8

4 Points

```
def f(x):  
    return x*x-2.  
def df(x):  
    return 2.*x  
x = 3.  
for i in xrange(10):  
    x -= f(x)/df(x)  
print x
```

What is the intended purpose of the above python code? State the mathematical problem.

---

---

---

What is the name of the method? \_\_\_\_\_

Name at least one other method we discussed in the lectures that does the same. \_\_\_\_\_

## Question 9

8 Points

Rewrite the following second order differential equation as a first order differential equation:

$$f''(x) = -(2\pi)^2 f(x).$$

Use the initial conditions  $f(0) = 0$ ,  $f'(0) = 1$  and a timestep of  $dx = 0.25$  to integrate the system for two timesteps using the Euler method.

What is your result for  $f(0.5)$ ? \_\_\_\_\_

What is the analytic result for  $f(0.5)$ ? \_\_\_\_\_

Describe two different ways how you could improve the numerical result:

1) \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

2) \_\_\_\_\_

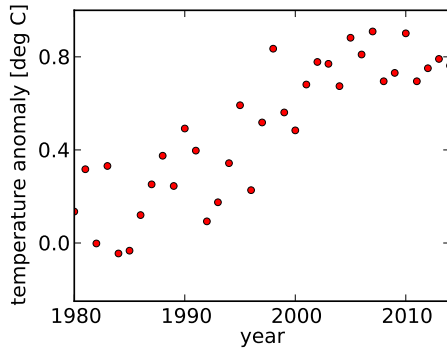
\_\_\_\_\_

\_\_\_\_\_

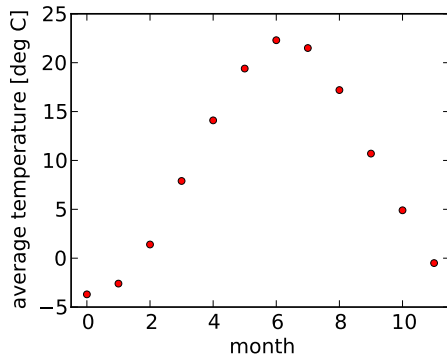
## Question 10

4 Points

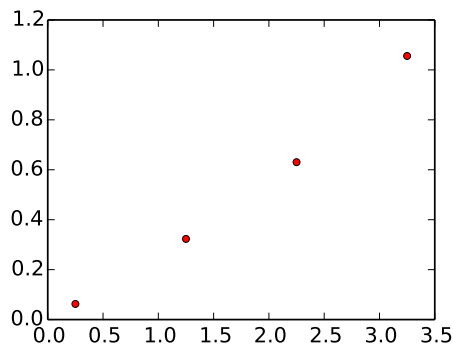
Mark the most reasonable method to fit each dataset.



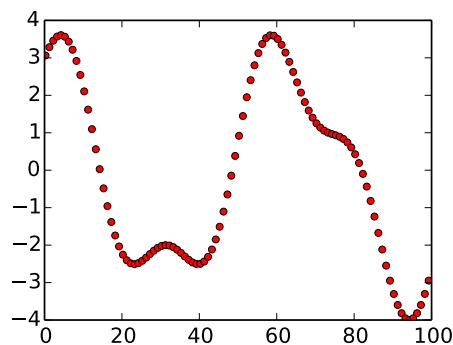
- Linear interpolation
- Least square fit of  $f(x) = a_0 + a_1 \cdot x$
- Cubic spline
- Lagrange polynomial



- Least square fit of  $f(x) = a_0 + a_1 \cdot x$
- Cubic spline
- Markov Chain Monte Carlo
- None of the above



- Piece-wise constant interpolation
- Least square fit of  $f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4$
- Lagrange polynomial
- Markov Chain Monte Carlo



- Piece-wise constant interpolation
- Least square fit of  $f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4$
- Lagrange polynomial
- Least square fit of  $f(x) = a_0 \cdot \sin(x) + a_1 \cdot \cos(x)$



## Question 11

4 Points

---

```
# Main loop
for i in xrange(5000):

    theta_proposed = [theta+0.1*numpy.random.randn()
                      for theta in theta_current]
    P_proposed = P(Dx,Dy,theta_proposed)

    # Calculate likelihood ratio (uses log likelihood to
    # avoid floating point issues)
    ratio = math.exp(P_proposed - P_current)

    # Accept the new theta values if likelihood is larger
    if ratio>1.:
        theta_current = theta_proposed
        P_current = P_proposed

    # Save current theta value in chain
    chain.append(theta_current[0])

# End main loop
```

---

The above python code shows the main loop of an implementation of the Metropolis-Hastings algorithm. The code is missing an important part of the algorithm. Propose a fix (you can write it directly into the code or below). Explain how the algorithm could get stuck in a local minima without your fix.

---

---

---

---

---

---

---

---

## Bonus Question

4 Bonus Points

The integrator IAS15 is a new numerical integrator to solve ODEs. It was published recently in a paper by Rein & Spiegel (2014). The integrator is 15th order and thus extremely accurate. In fact, it is so accurate that after one timestep, the relative error is of the order of machine precision (double floating point precision). Estimate how large the relative error is after  $10^{10}$  timesteps. First, assume that the errors of each timestep add up linearly. This scenario might occur if the error made in a each individual timestep is not completely independent from the previous error made.

---

---

---

---

Second, assume that the errors are uncorrelated. In that case the sum of the errors behaves like a random walk and only grows like the square root of the number of timesteps (this is known as Brouwers law).

---

---

---

---

## Appendix A Assembler commands

- **ADD r1 r2 r3**  
This command adds the value stored in register r1 to the value stored in r2 and stores the result in r3.
- **IF r1 r2**  
This command only executes the next command if the value in register r1 is bigger than the value in register r2.
- **COPY r1 r2**  
This command copies the value in register r1 to register r2.
- **SET s1 r1**  
This command sets the value in register r1 to the number s1 (s1 is just a number, not the address of a register).
- **JUMP r1**  
This command moves the instruction pointer by the value of r1.
- **PRINT r1**  
This command prints the value in register r1 on the screen.