PH502 Assignment 1

ICHEC

Deadline: 16 February 2018 at 17:30

1. Create a new git repo for this assignment. At the end of the assignment please capture the output of the 'git status' and 'git log' commands (e.g. `$ git log > gitlog_output` will write the output to a file called `gitlog_output`).

   For each subsequent question create a separate directory in which the code and doc are kept. The git repo should cover the all these sub-directories. What we are looking for is stepwise development by committing code (plus comments) on a regular basis.

2. Create a program that applies this integer function repeatedly;

$$f(n) = \left.\begin{array}{ll} n/2 & \text{if n even} \\ & \\ 3n+1 & \text{if n odd} \end{array}\right\}$$

   (See `http://en.wikipedia.org/wiki/Collatz_conjecture`)

   (a) The user enters a positive integer from the command line

   (b) Applies the function above

   (c) Terminate the program when the last three numbers in the sequence are 4,2,1.

   (d) Print out the sequence of numbers, in the minimum number of lines, but ensuring that the printout does not extend more than 40 characters for each line.

```
! This format does not add a newline.
write(6,fmt='(i0,a)',advance='no') i,', '
```

```
/* Omit "\n" to continue printing to same line */
printf("%d, ",i);
```

   (e) For example when starting from 50 the result is;

```
50, 25, 76, 38, 19, 58, 29, 88, 44, 22,
11, 34, 17, 52, 26, 13, 40, 20, 10, 5,
16, 8, 4, 2, 1,
```

3. Write a program that can carry out the following steps:

   (a) Write a function that calculates the determinant (`http://en.wikipedia.org/wiki/Determinant`) of any 3x3 matrix.

$$\begin{aligned} \det\left(\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}\right) &= a\det\left(\begin{bmatrix} e & f \\ h & i \end{bmatrix}\right) - b\det\left(\begin{bmatrix} d & f \\ g & i \end{bmatrix}\right) + c\det\left(\begin{bmatrix} d & e \\ g & h \end{bmatrix}\right) \\ &= a(ei-fh) - b(di-fg) + c(dh-eg) \\ &= aei + bfg + cdh - afh - bdi - ceg \end{aligned}$$

   (b) Use a set of loops to generate the following matrix (Hilbert matrix):

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{pmatrix}$$

(c) Use Cramer's rule (`http://en.wikipedia.org/wiki/Cramer's_rule`) to calculate the determinant of the 4x4 matrix above.

$$\det\left(\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix}\right) = a\det\left(\begin{bmatrix} f & g & h \\ j & k & l \\ n & o & p \end{bmatrix}\right) - b\det\left(\begin{bmatrix} e & g & h \\ i & k & l \\ m & o & p \end{bmatrix}\right) + c\det\left(\begin{bmatrix} e & f & h \\ i & j & l \\ m & n & p \end{bmatrix}\right) - d\det\left(\begin{bmatrix} e & f & g \\ i & j & k \\ m & n & o \end{bmatrix}\right)$$

(d) Write the matrix and the determinant to the standard output using a formatted print statement.

4. Similar to the Trapezoidal Rule exercise in the first practical the Simpson's Rule is another method for numerical integration (`http://en.wikipedia.org/wiki/Simpson's_rule`). If the function is sampled at $n+1$ even intervals $\{f_o = f(a), f_1 = f(a+h), f_2 = f(a+2h), \cdots, f_n = f(b)\}$ then,

$$\int_a^b f(x)dx \sim \frac{h}{3}\left[f(a) + 4(f_1 + f_3 + \cdots) + 2(f_2 + f_4 + \cdots) + f(b)\right]$$

(a) Create a program that calculates the area under the curve $y = tan(x)$ from $0 \to \pi/3$.

(b) The user enters the number of points that will be used in calculating the area, which should be odd. Check if it is odd or even.

(c) Empirically find the number of points needed to get the area within 1.0×10^{-5} of the actual result ($\ln(2)$). You should not neeed more than 1000 points to get this.

5. Create a program, numericalIntegration.c/f90, that calculates the area under the curve $f(x) = sin(x)$ from $0 \to \pi$ using the Trapezoidal and Simpson's Rule and Gaussian Integration. Compare with the actual result.

Trapezoidal rule

$$\int_a^b f(x)dx \sim \frac{b-a}{2N}(f(x_0) + 2f(x_1) + 2f(x_2) + \cdots + 2f(x_{N-1}) + f(x_N))$$

Where $x_0 = a$ and $x_N = b$ with $N-1$ equidistance points in between.

Simpson's rule

If the function is sampled at $n+1$ equal intervals (n must be even) $\{f_o = f(a), f_1 = f(a+h), f_2 = f(x+2h), \cdots, f_n = f(b)\}$ then,

$$\int_a^b f(x)dx \sim \frac{h}{3}\left[f(a) + 4(f_1 + f_3 + \cdots) + 2(f_2 + f_4 + \cdots) + f(b)\right]$$

Gauss Quadrature

Using the two point Guass quadrature rule, the integral of sin(x) can be approximated by

$$\int_a^b f(x)dx \sim h\left[f\left(\left(k-\frac{h}{\sqrt{3}}\right)\right)+f\left(\left(k+\frac{h}{\sqrt{3}}\right)\right)\right],$$

where $h=\frac{b-a}{2}$ and $k=a+h$

(a) Make each integration method a separate function each in a separate file.

(b) Create a Makefile that compiles and links the numericalIntegration program. Also add a 'clean' target that removes all object files.

(c) Allow the user to pick the integration method and the number of intervals from the command line.

(d) Print the calculated value and the actual value to the screen.

(e) Tabulate the results of all methods for the following interval values: $n=\{2,8,16,64\}$.

(f) Comment on the results obtained.

Please tar or zip all the files and directories for all of the questions and email to Adam (`adam.ralph@ichec.ie`) by the required time.