Hannah Machado

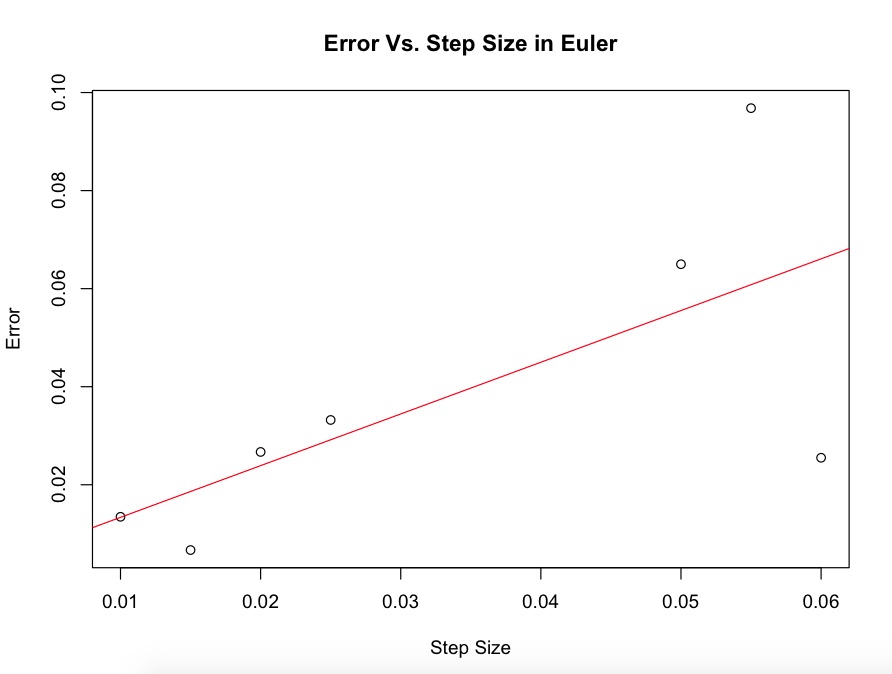
CSC 210

Apr. 12th, 2017

Homework 10

1. We showed in class that the global error bound in Euler's algorithm is proportional to the step size. Now, in PHASER solve the initial-value problem x' = x, x(0) = 1 to compute x(1) = e = 2.7182818284590452354, using Euler's algorithm with seven different step sizes. Using R, plot the errors against the step sizes. Next, use the lm function of R to determine the "best-fit line" to the data points. What is the proportionality constant of errors vs. step size?

|  |  |
| --- | --- |
| Step Size | Error |
| 0.01 | 2.7182818284590452354 - 2.704813829421526 = 0.01346799903 |
| 0.015 | 2.7182818284590452354 -2.711595039150290 = 0.0066867893 |
| 0.02 | 2.7182818284590452354 -2.691588029073605= 0.02669379938 |
| 0.025 | 2.7182818284590452354 - 2.685063838389973 = 0.03321799006 |
| 0.05 | 2.7182818284590452354- 2.653297705144421 = 0.06498412331 |
| 0.055 | 2.7182818284590452354 -2.621466265900668=  0.09681556255 |
| 0.06 | 2.7182818284590452354 -2.692772785766811 = 0.02550904269 |

The error is proportional to the step size in Euler

2.**Beetle model:** Download the R file beetle\_parameters.R from the Beetle episode and run (source) it. Read out the parameter values of a and b for the logistic differential equation y' = ay + by^2. Now, use these parameter values in the Cubic1D ODE differential equation in Phaser, with the appropriate initial condition, and print out the Xi values for 14 days.

Now write an R code that computes the averages of the data points and Phaser points and prints out the difference between the two averages.

Phaser points:

0.00000E+000 1.000000000000000E+000

1.00000E+000 1.950066560664090E+000

2.00000E+000 3.805703072430779E+000

3.00000E+000 7.430066887613052E+000

4.00000E+000 1.450904651729568E+001

5.00000E+000 2.833545996713467E+001

6.00000E+000 5.534072328075156E+001

7.00000E+000 1.080864515413754E+002

8.00000E+000 2.111075484824234E+002

9.00000E+000 4.123247231226571E+002

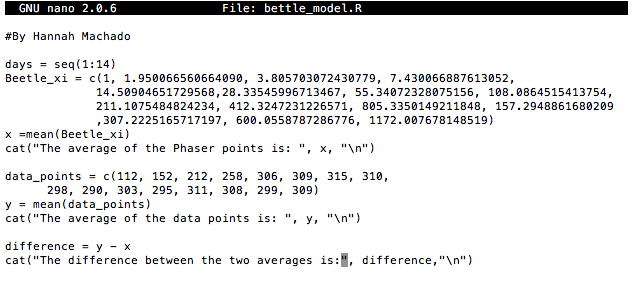
1.00000E+001 8.053350149211848E+002

1.10000E+001 1.572948861680209E+003

1.20000E+001 3.072225165717197E+003

1.30000E+001 6.000558787286776E+003

1.40000E+001 1.172007678148519E+004





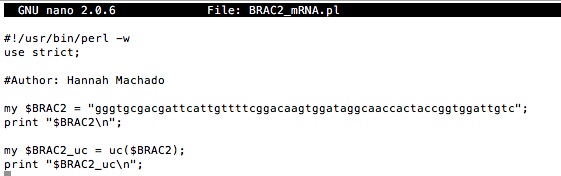
**3. BRAC2 mRNA: Visit the Web page**

<http://www.ncbi.nlm.nih.gov/nuccore/224004157?report=genbank>

containing a partial coding mRNA sequence of the breast cancer gene BRAC2. You can read more about this gene at<https://ghr.nlm.nih.gov/gene/BRCA2>

Observe that the first 60 nucleotides of this gene are:

**gggtgcgacg attcattgtt ttcggacaag tggataggca accactaccg gtggattgtc**

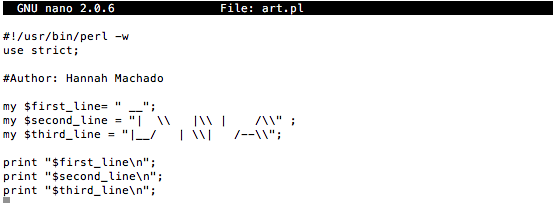
**Create a Perl program that stores this sequence without the blank spaces in a string and prints out the string in lower case and also in upper case letters.**

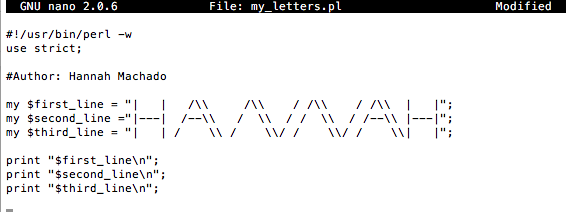
****

**4. ASCII art:** Write a Perl program that prints out the image below.

\_\_   
| \ |\ | /\   
|\_\_/ | \| /--\

*Note:* You can copy the image and paste it into your editor. To print the escape character \ you need to use \\ .

*Personal challenge:* Design your own letters to print out your initials.

Personal challenge:

**5. Finding bugs:** Find and exterminate the bugs in the Perl code below

#!/usr/bin/Perl -w  
use strict  
  
my $firt 10\_bp = "gggtgcgacg";  
my $second\_10\_bp = "attcattgtt";  
 $gene = "BRAC2";  
  
print 'The first 20 bp of $gene are";  
print "$first\_10\_bp$second\_10\_bp\n";

so that it prints out:

The first 20 bp of BRAC2 gene are gggtgcgacgattcattgtt

