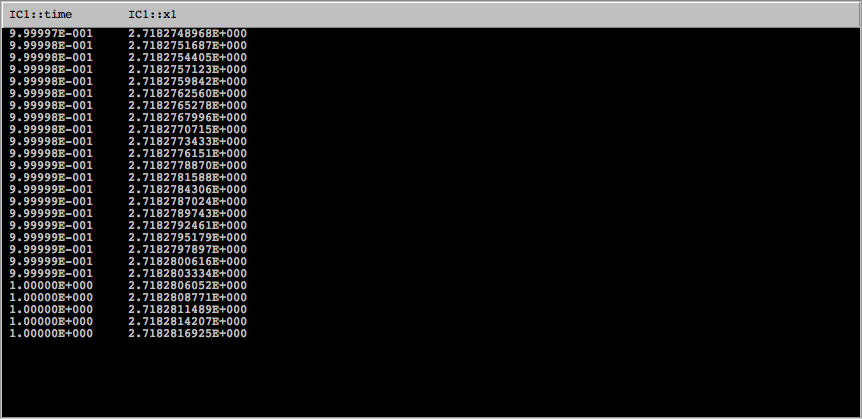
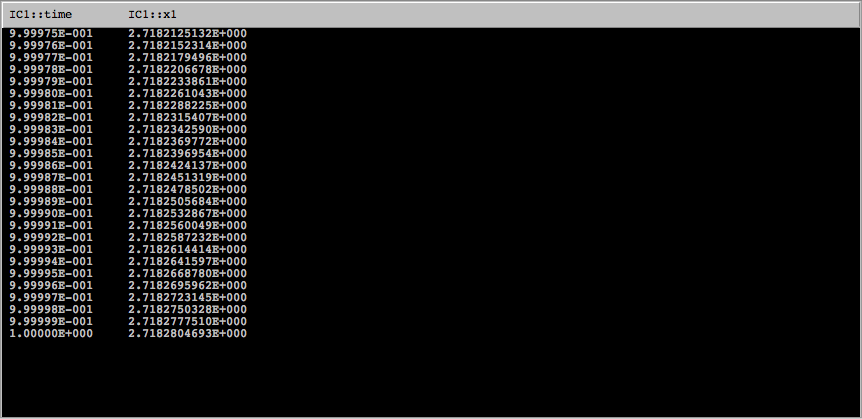
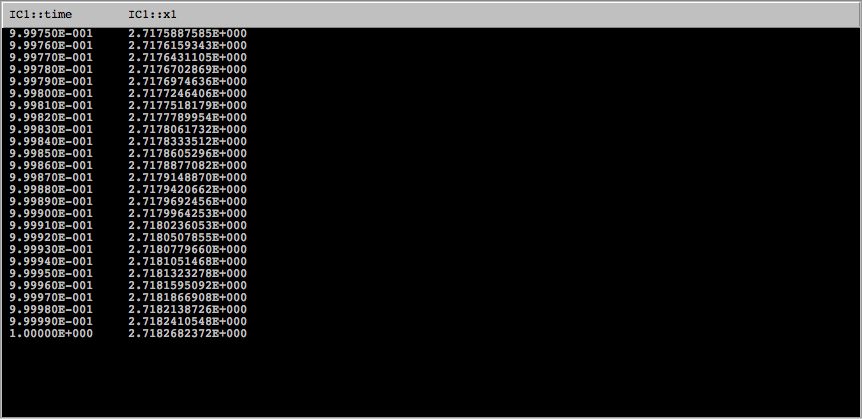
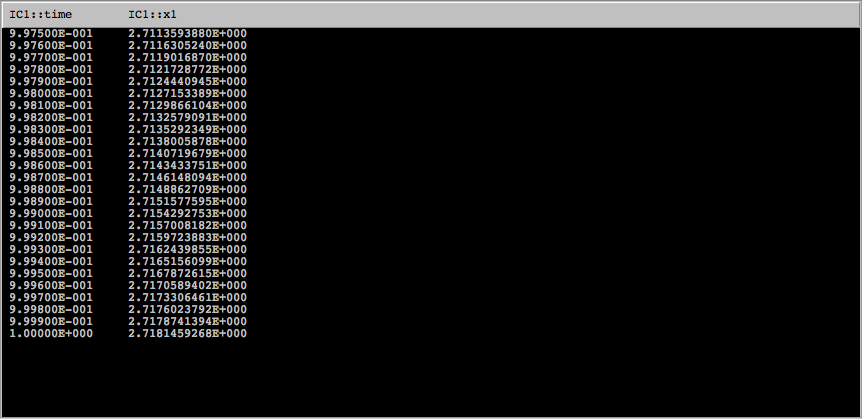
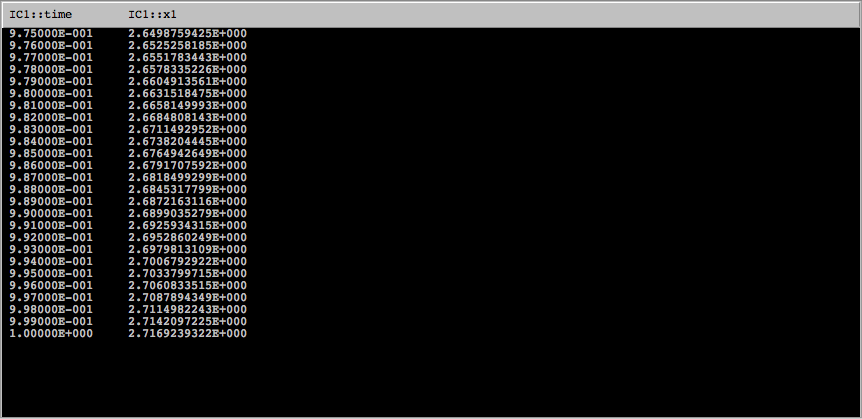
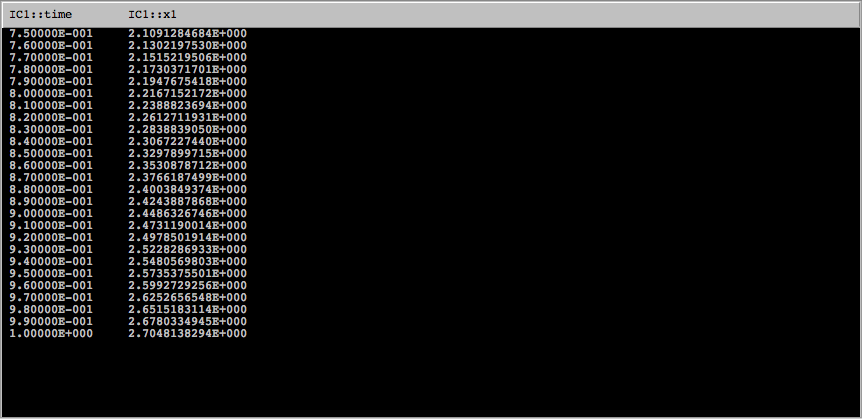
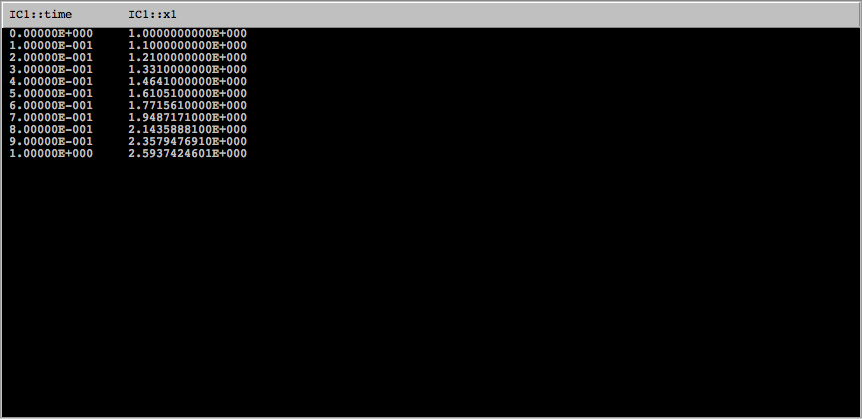
Isabel C Wyss

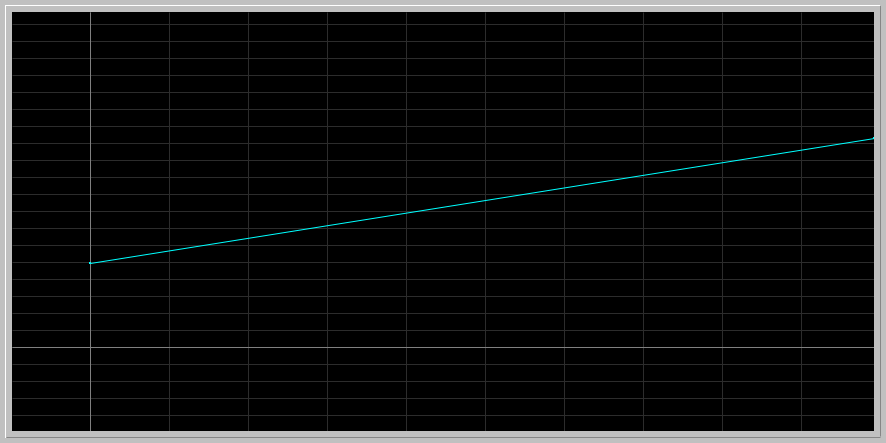
CSC 210

**Homework: Week 6**

1. **Error vs. step size in Euler:** 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 six different step sizes. Using your favorite plotting program, plot the errors against the step sizes. Do you get a linear relationship? Be careful of the scales on your graph.

* The following iterations and graph are for the step sizes: h=0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001, 0.0000001





* When the errors are plotted against the step sizes, a linear relationship is formed, as we can see from the positive linear slope in the graph
* This means that as the step size increases, the errors increase too, and vice versa as they decrease.

1. **An explosion problem:** Here we consider the "explosion" problem x'= x^2, x(0)=1. This simple equation shows up in chemical reactions where two atoms get together to form a molecule.
   * Verify that the solution of this initial-value problem is x(t) = 1/(1-t).

x’(t) = (1 - t)^-1

= -(1-t)^-2

= (1-t)^-2 → This is the equivalent to x’ = x^2, meaning the solution is verifie

* + Using Euler's Algorithm with steps h = 0.01 and h = 0.005, compute x(0.99). What is the error in your computations? Hint: What is the exact value of x(0.99) = ?
* The exact value of x(.99) = 100
* With a step size of 0.01 → value = 2.6780334945, error = 97.321966506
* With a step size of 0.005 → value = 2.6846039681, error = 97.315396032
  + Now in Phaser, compute x(1) with Euler using steps h = 0.01 and h = 0.005. What is the error in your calculations?
* Exact value of x(1) → infinity
* Step size of 0.01 → value on phaser: 2.708138294
* Step size of 0.005 → value on phaser: 2.7115171229
* The error for both step sizes of 0.01 and 0.005 is infinite, this is because there is an asymptote at x = 1

1. **One step of Runge-Kutta (4):** Consider our canonical example of x' = x, x(0) = 1. By hand, compute one step of Runge-Kutta (4) algorithm with step size of h = 1. Compare your answer to the one you obtain from Phaser. Note: The formulas for Runge-Kutta (4) are available in PhaserHelp, or the link above.

COMPUTATIONS BY HAND:

* yn+1 = yn + ( K1 + 2K2 + 2K3 + K4) / 6 ,

Where:

* K1 = h f( tn, yn ),
* K2 = h f( tn + (1/2) h, yn + (1/2) K1 ),
* K3 = h f( tn + (1/2) h, yn + (1/2) K2 ),
* K4 = h f( tn + h, yn + K3 ).
* K1= 1\*f(xn)= 1\*1=1
* K2= 1\*f(xn +K1)= 1\*2=2
* K3= 1\*f(xn +K2)= 1\*3=3
* K4= 1\*f(xn +K2)= 1\*4=3
* yn+1= 1 + (1+2+3+4)/6= 2.66666667

COMPUTATIONS BY PHASER:

* 0.00000E+000 1.0000000000E+000
* 1.00000E+000 2.7083333333E+000

1. **Euler for systems of ODEs:** Euler's algorithm can be generalized for systems of ODEs. For example, for the pair of differential equations

dx/dt = f (x, y)

dy/dt = g (x, y)

with initial conditions x(0) = x\_0 and y(0) = y\_0, Euler's algorithm with step size h becomes

x\_(n+1) = x\_n + h\*f(x\_n, y\_n)

y\_(n+1) = y\_n + h\*g(x\_n, y\_n).

Now consider the Epidemics problem from last week. Take a = 0.6, r = 0.003, S\_0 = 200, I\_0 = 20, and step size h = 0.2. Compute two steps of Euler by hand. Compare your numbers with those from Phaser.

BY HAND:

ds/ dt = -rSI, dI/dt = rSI - aI

S1 = 200 + 0.2 ( -0.003 \* 200 \* 20) = 197.6

I1 = 20 + 0.2 ( 0.003 \* 200 \* 20 - 0.6 \* 20) = 20

S2 = 197.6 + 0.2 (-0.003 \* 197.6 \* 20) = 195.228

I2 = 20 + 0.2 (0.003 \* 197.6 \* 20 - 0.6 \* 20) = 19.9712

IN PHASER:

S2 = **1**.**952288** → I2 = **1.9971200**

1. **Reading Euler:** Try to read the original paper of Euler listed above. Is his algorithm the same as the one we derived in class? What does he have to say about errors? This problem is for your own edification; you do not have to turn it in.

* The algorithm written in Euler’s original paper is slightly different from the one that was derived in class
* The difference between the two equations is as follows: in the paper, the algorithm is derived as: b’ = b + A (a’-a), in class, the algorithm was derived as Xn+1 = xn + hf(x).
* Although the equations are different in structure/ format, the derived meaning is essentially the same
* In the original paper of Euler, it explains that when smaller intervals are taken, more accurate values will be obtained, and the errors accumulate due to multitude
* Hence, the more steps that are taken, the more there will be accumulated error