Review Practical Programming

Review of Common Issues

Can you find the problem in this code?

4.

While loop:

```
> x=0.3
> i=seq(55)
> h = 0

> while(x > h)
+    {h <- 1+(sum(h+x^i))}
> h #display sum of function
[1] 1.428571


> x=6.6
> i=seq(8)
> h = 0

> while(x > h)
+    {h <- sum(h+x^i)}
> h #display sum of function
[1] 4243335
```

It generates the correct answer, for the wrong reason

Mixing a loop and a vectorized solution

Function vs. Script

```
> x=seq(1:100)
> domeig=c(mean(x),var(x))
> domeig
[1]  50.5000 841.6667
```

```
> domeig <- function(x){
+   m <- mean(x)
+   v <- var(x)
+   y=list(mean=m,variance=v)
+   return(y)
+ }
```

Given this function, we can now use it.

```
> x <- rnorm(8)
> domeig(x)

$mean
[1] -0.1757752

$variance
[1] 1.104
```

Lab 3 Exponential Growth

Exponential Population Growth

Discrete Time – Update Rule

$$X(t+\Delta t) = X(t) + X\Delta t - Y\Delta t$$

$$X(t+\Delta t) - X(t) = (X - Y)\Delta t$$

$$\frac{X(t+\Delta t) - X(t)}{\Delta t} = X - Y$$

Can use this equation to predict future population size

Continuous Time – Differential Equation

$$\frac{dN}{dt} = rN$$

Rate of Change

How do we use this to predict future population growth?

Two Approaches

Analytical Solution – Exact

$$\frac{dN}{dt} = rN \xrightarrow{\text{Integrate}} N_t = N_0 e^{rt}$$

Problem: many functions cannot be integrated.

Solution
Numerical Approximation Algorithms

- Euler
- Revised Euler
- Runge-Kutta 2, 4
- lsoda

Reading from Shiflet and Shiflet

Numerical Approximation: Euler Algorithm

Finite difference method

```
growth_rate = 0.10
population(0) = 100
growth(t) = growth_rate * population(t - Δt)
population(t) = population(t - Δt) + growth(t) * Δt
```

Projections

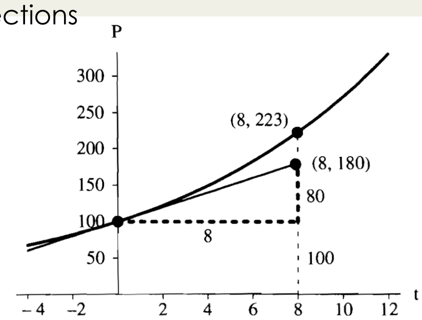


Figure 5.2.1 Actual point, (8, 223), and point obtained by Euler's Method, (8, 180)

Explicit time step

Generates Error
Dependent on time step and rate of change

Estimating Error

If we know the exact solution (Na), we can compare the numerical approximation (Np) to it.

$$RMSEP = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Np_i - Na_i)^2}$$

Numerical Approximation: lsoda

- Industrial strength solver
- Uses a look forward and look backward approach
- Variable time step

Tasks

3.1 Discrete time model projections

3.2 Forrester type diagram of continuous time model

3.3 Continuous time population projections
- use exact solution

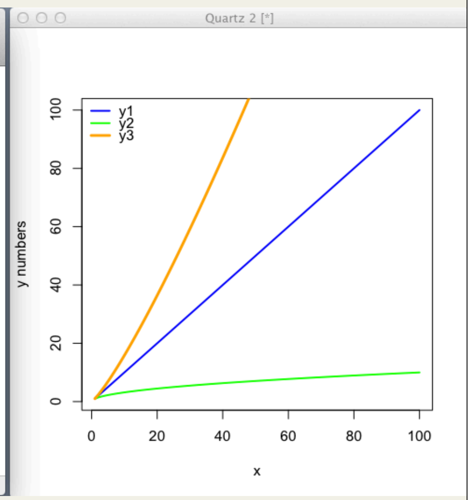
3.4 Numerical approximation: compare accuracy of two techniques
- Euler
- lsoda (deSolve) package

Programming Tip

Multiple Lines on a Graph

Example_Plot_multiple_lines.R

```
x=1:100
y1 = x
y2 = x^(1/2)
y3 = x^(1.2)

# first plot - builds plot window
plot(x,y1,col="blue",type="l",lwd=2,
   ylab="y numbers")

# add another line
points(x,y2,col="green",type="l",lwd=2)

# add a thid line
points(x,y3,col="orange",type="l",lwd=3)

# create a legend
legend("topleft",
  legend=c("y1","y2","y3"),
  col=c("blue","green","orange"),
  lwd=c(2,2,3),
  bty="n")

# note that the first plot call sets the x & y axis limits.
# These may need to be adjusted.
```



Laboratory Report Format

- 1 paragraph introduction restating the laboratory objectives.
- For each task
  - Describe the task
  - Describe the action you take (methods)
  - Results: present evidence of task completion (usually plots)
  - Describe an ecological explanation or interpretation of your results. Answer any questions.
- Place a copy of R code in an appendix.
  - Please practice neat coding – formatting & comments

Tasks

3.1 Discrete time model projections

3.2 Forrester type diagram of continuous time model

3.3 Continuous time population projections

- use exact solution

3.4 Numerical approximation: compare accuracy of two techniques

- Euler
- lsoda (deSolve) package