J in the Mathematics Classroom

Cliff Reiter, Department of Mathematics, Lafayette College, Easton, PA 18042 U.S.A.

In the past several years I have used computer laboratory sessions as a regular part of my college level mathematics teaching. In the fifteen most recent classes, I have used *Mathematica* as the primary software six times (in the calculus sequence) and I have used J nine times. I have used J in Number Theory, Discrete Structures, Linear Algebra with Applications, and in a special topics course: Mathematical Visualization, Fractals and J. This note summarizes and illustrates my computer enhancement of those courses in the most recently offered version.

Lab Exercises in Math 328 Number Theory

L01 Computations, Theorems, Conjectures: Exploring Prime Numbers
The Sieve of Eratosthenese, Counting Numbers of Primes, The Prime Number Theorem, The Twin Prime Conjecture.

L02 Euclidean Algorithm and Linear Diophantine Equations
Multivariable extended gcd's and solving linear Diophantine equations

L03 Solving Equations and Congruences by Brute Force
Warm Up: Solving Some Congruences, Exploring Quadratic Congruences, Exploring Cubic Congruences, Using Congruences to Analyze Integer Equations. The students are then expected to be able to answer the following using a combination of mathematical and computer techniques:

> ***Find all the powers of ten that are two more than twice the square of a positive integer and explain how you arrived at your conclusion. Your answer should be a short paragraph explaining your conclusion along with the solution(s) you determined. Your explanation will likely include a brief J expression and result that verify your claim regarding solutions to a certain congruence.***

L04 Some Experiments with Factoring
Trial Division, Pollard's p − 1 Factorization Method, The Pollard Rho Method, J's q:

L05 Patterns with Quadratic Residues
Quadratic Residues by Brute Force, Primes mod 4, discovering properties of the Legendre symbol

L06 Some Experiments with Primality Testing
Direct Factorization, Pepin's Test, Probabilistic Primality Testing, Applied to Repunits

L07 Diophantine Equations: Pythagorean Triples
Pythagorean Triples via Brute Force, Some Density Experiments, Parametrizing Pythagorean Triples (led to circle proof), Parametrizing a Pellian Equation

L08 Continued Fractions and Pell's Equation
Convergents and Rationals, Convergents and Quadratics, Pell's Equation and Fundamental Units, How fast do units grow?

An illustration of how a few of the patterns with the Legendre symbol (Lab 05) might be discovered follows. The Legendre function is defined below for an odd prime *p* so that `a leg p` is 1 if *a* is a square modulo *p*, _1 if it is not a square, and 0 if gcd(*a*, *p*) = *p*. Notice that primes that are 1 versus 3 modulo 4 have distinct behaviors regarding quadratic residues.

```
   sq=: ] | *:@i.

   sq 11
0 1 4 9 5 3 3 5 9 4 1

   mod=: |~

   13 mod 11
2

   char=: mod = sq@]

   3 char 11
0 0 0 0 0 1 1 0 0 0 0

   leg=: <:@(+/)@char"0

   3 14 11 2 leg 11
1 1 0 _1

   ]pr1=:(#~ 1: = 4&|) p:>:i.50
5 13 17 29 37 41 53 61 73 89 97 101 109 113 137 149 157 173
181 193 197 229 233

   ]pr3=:(#~ 3: = 4&|) p:>:i.50
3 7 11 19 23 31 43 47 59 67 71 79 83 103 107 127 131 139 151
163 167 179 191 199 211 223 227

   _1 leg pr1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

   _1 leg pr3
_1 _1 _1 _1 _1 _1 _1 _1 _1 _1 _1 _1 _1 _1 _1 _1 _1 _1 _1 _1
_1 _1 _1 _1 _1 _1 _1

   pr3 leg 5
_1 _1 1 1 _1 1 _1 _1 1 _1 1 1 _1 _1 _1 _1 1 1 1 _1 _1 1 1 1 1
_1 _1

   5 leg pr3
```

```
_1 _1 1 1 _1 1 _1 _1 1 _1 1 1 _1 _1 _1 _1 1 1 1 _1 _1 1 1 1 1
_1 _1

   pr3 leg 3
0 1 _1 1 _1 1 1 _1 _1 1 _1 1 _1 1 _1 1 _1 1 1 1 _1 _1 _1 1 1
1 _1

   3 leg pr3
0 _1 1 _1 1 _1 _1 1 1 _1 1 _1 1 _1 1 _1 1 _1 _1 _1 1 1 1 _1
_1 _1 1
```

Lab Exercises in Math 182: Discrete Structures

L01 Logic and Propositional Equivalence
Warm up (J), Truth Tables with Two Propositions, Testing Several Propositions, Truth Tables with Three Propositions, Testing Several Propositions

Boolean Expression	J expression
$\neg p$	`-. P`
$p \vee q$	`p +. Q`
$p \wedge q$	`p *. Q`
$p \oplus q$	`p ~: q`
$p \rightarrow q$	`p <: q`
$p \leftrightarrow q$	`p = q`

Table of Boolean and J Expressions.

L02 Algorithms, Complexity and a Glimpse at Sorting
Informal Discussion of Time Complexity, Bubble Sort, Merge Sort, J Sort

L03 Extended Euclidean Algorithm and Random Sequences
Extended Euclidean Algorithm, A Classic "Random" Number Generator, Inverses Modulo m, Reversing a Random Sequence.

L04 Representing Relations with Matrices and Digraphs
Matrices and Digraphs Representing Relations, Combining Relations and Closures, Transitive Closure I, Transitive Closure II: Warshall's Algorithm

Experiments that exercise the tautology that $p \rightarrow q \equiv \neg p \vee q$ follow. Of course, students investigated much more complex tautologies in Lab 01.

```
   'p q'=:|: #: i.2^2

      p,.q,.(p<:q)
0 0 1
0 1 1
1 0 0
1 1 1
      p,.q,.(-.p)+.q
0 0 1
0 1 1
```

1 0 0 1 1 1

Labs Exercises in Math 272: Linear Algebra with Applications

L01 Curve Fitting by Solving Linear Systems
Warm Up: J Arithmetic, Warm Up: Vectors, Matrices and Solving Systems, Determining a Quadratic Model for Board feet versus Diameter, Polynomials from Power Tables (Vandermonde Matrices), Higher degree fits, Hermite Polynomials (A Fit Using Derivatives)

L02 Traffic Flow
Getting Row Reduced Matrices using linear.ijs, set up a traffic flow system of 8 equations in 12 unknowns and get and interpret the general solution.

L03 Properties of Determinants
Exploring Determinants and matrix algebra properties, row reduction facts and applications to volumes of parallelepipeds.

L04 Transition Matrices
Multi-step Transitions, Populations, lots of interpretation

L05 Geometric Transformations of the Plane
Plotting and Scaling a Polygon, Rotations, Homogeneous Coordinates and Translations, Creating an Animation

L06 Geometric Transformations in Three Dimensions
Plotting a Polygon in 3D, Geometric Transformations in 3D, Creating Animations in 3D

L07 Frame's Method, Eigenvalues and Spotted Owls
Frame's Method, Eigenvalues, Dynamics, Spotted Owls, How much more land is required for stability?

L08 Projection and Least Squares Fitting
Least Squares, Projections, Fitting Least Squares to Wine Data (linear long term trend, annual cyclic pattern).

Lab 03 allows students to empirically discover many properties of determinants which seems to make a more complete discussion of them easier. Lab 07 is one of their favorites since it leads to computations that suggest a certain amount of additional forest is required to allow spotted owls population to become stable. A favorite of mine is Lab 08 where a somewhat periodic behavior appears on top of a linear trend (Figure 1) can be modeled with a linear plus fourier model. Getting the least squares fit to 142 equations in 22 unknowns results in an impressive fit. See Figure 2.

```
load 'plot'

load 'd:\t\2014\jimc\wine.ijs'

mp=:+/ . *
```

```
   plot x;y
```

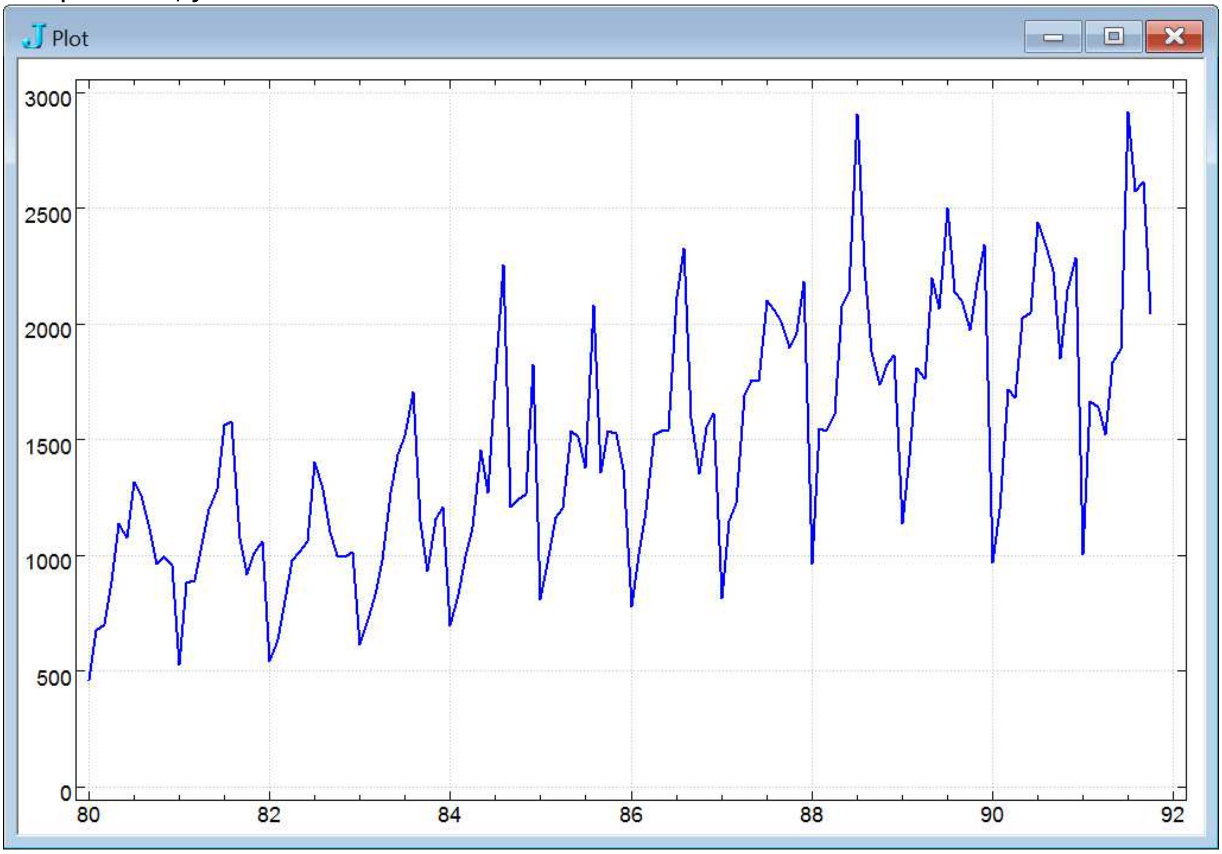


Figure 1. Periodic behavior with a linear trend.

```
   linfour=: (],[:}.@, 1 2 o./ i.@>:@[ * o.@] )"0

   3 linfour 1.11
1.11 _0.338738 0.637424 _0.860742 1 _0.940881 0.770513 _0.509041

   $10 linfour x
142 22

   $coef=: y %. 10 linfour x
22

   $z=:(10 linfour x) mp coef
142

   'pensize 2' plot x;y,:z
```

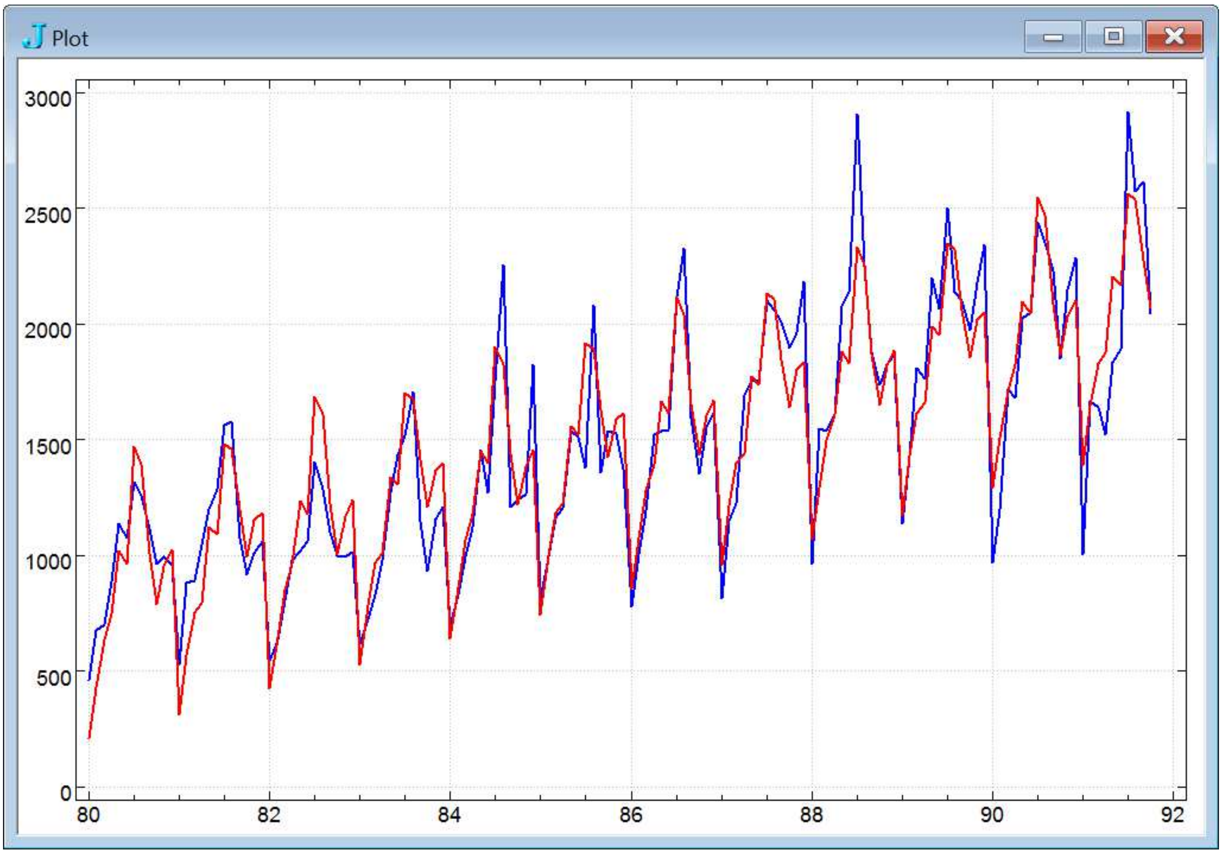


Figure 2. Data and a linear-fourier fit.

Math 379 Special Topic: Mathematical Visualization, Fractals and J

Course Description: Mathematical Visualization, Fractals and Chaos

This course explores the visualization of mathematical objects and algorithms using computer graphics and the programming language J which will be introduced as needed. The topics include fractals, chaos, fractal dimension, iterated function systems, finite automata, fuzzy logic, image processing, complex dynamics, frieze, crystalline and hyperbolic symmetry, and chaotic attractors. Three-dimensional representations will be projected to two dimensions, ray-traced and manipulated in real time.

Prerequisite of Math 272 (Linear Algebra with Applications) or Corequisite of Math 300 (Vector Spaces) or permission of the instructor

This course covered a significant portion of the author's Fractals, Visualization and J text. There were two one hundred point in class exams, approximately 30 homework assignments totaling

around 300 points and a final project worth 150 points. During the spring of 2014 there were 16 students in the class. The following Table of contents is highlighted in yellow where a section was primarily discussed in demonstration mode. The green is where students primarily explored the topics by themselves.

Table of Contents

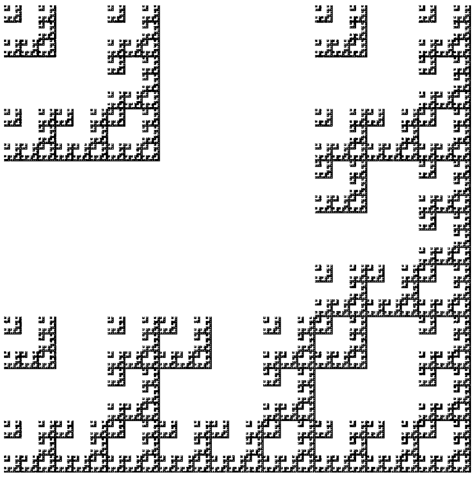
Exam 2

The in-class exams were 75 minutes long. The students had J and open book resources (including their homework scripts); however, they were not allowed to use a browser to visit external sites. Each question was worth 12 points. There were 12 questions on the exam (6 in the Translation and Theory section, 6 in the Practicum section). They could answer as many as they liked with 100 points being considered perfect. Partial credit (and a curve) were given. Student evaluations of the course suggest that students found the homework and exams much more rewarding than the demonstration mode. The raw first exam scores (same format) ranged from 23 to 78. The raw second exam scores ranged from 54 to 111. The questions from the second exam follow.

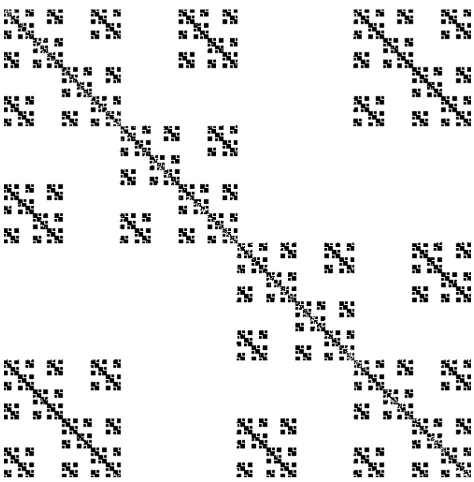
Translation and Theory

Answers for problems (1)-(6) were on the paper exam.

(1) By creating a table as in Section 4.8 compute the fractal dimension of the figure below



(2) It is true that the fractal dimension of a non-overlapping fractal where each component arises from a rescaling (the same for the *x* and *y* direction) and a translation satisfies $\sum_i s_i^d = 1$ where the s_i are the rescaling factors.Write down an equation whose solution would give the fractal dimension of the figure below. You do not need to solve for *d*.



(3) (a) Write an adverb that computes $f(y) = \sin^m(\pi y) + \cos^m(\pi y)$

(b) Write a conjunction that computes $g(y) = \sqrt[m]{y^n + 1}$

(4) Express in common math notation:

(a) `f=: 2 : 'm * (-.y)*y^n'`

(b) `g=: 2 : '3&o.@((m%n)&* '`

(5) Suppose that local life used in Section 6.3 was replaced by

```
   ]L=: 1,1 20 1,:1
1  1 1
1 20 1
1  1 1

   llife=: +/@:,@(L&*) e.  2 3 21 23"_
```

Describe in words when a dead cell will become alive and an alive cell will stay alive.

(6) Suppose that `b` is a 600 by 800 by 3 array representing an image. In words describe how the following images are related to the original.

(a) `|. b`

(b) `|.”2 b`

(c) `|.”1 b`

Practicum

Problems (7)-(12) were answered in a word document including the J computations and results (movies were sent as separate files).

(7) Create a high resolution logistic type image of $f(x) = \lambda \sin(3x)$ using `cile254` to display the frequency information.

(8) Create a chaotic attractor with D_9 symmetry.

(9) Run a Boolean automaton using windows of width 5 and applying `rule 1e9` where
`rule=: (32#2)&#:`
You should use periodic boundary conditions and a random initial configuration with 128 cells. Exhibit the same number of time steps with super pixels of size 6.

(10) Create a movie of the time evolution of the majority rule automata on 5 by 5 neighborhoods.

(11) Construct the fractal shown in Problem 1 of this exam using inverse iterated function systems.

(12) Create a grayscale version of the *keys.jpg* (see Section 8.1) image by averaging the red and blue components.