Keeping it R.E.A.L.

Anthony Tongen
Associate Professor
NSF-UBM

• Multiple team-taught courses per year
  – Mathematical Models in Biology
  – Biometry
  – Biomechanics/Biophysics

• Two semester ‘Calculus with Functions I,II’ that is equivalent to Calculus I (whatever equivalent means)

• Interdisciplinary research with biologists
James Madison University’s Computational Courses

- Math 248 – Computers and Numerical Algorithms
  - Programming in a high-level language
  - Introduction to algorithms involving rootfinding, solving systems of linear equations, integrating, differentiating, and interpolating.
James Madison University’s Computational Courses

- **Math 448 – Numerical Analysis**
  - Further study and analysis of algorithms used to solve nonlinear equations, systems of linear and nonlinear equations. Includes iterative methods.

- **Math 449 – Numerical Analysis for Differential Equations**
  - Further study and analysis of numerical techniques to solve ordinary and partial differential equations.
Keeping it R.E.A.L.: Research Experiences for All Learners
Research and Classroom Projects in Computational Mathematics

Carla D. Martin and Anthony Tongen
James Madison University

DRAFT version: May 2010
MOSAIC emphases

• Algorithmic development
  – Converting words to a written algorithm (mathematical expression)
  – Converting the written algorithm basic code to perform the ‘experiment’

• Computers aren’t perfect!!!!!!
  – Examples that show that computers don’t actually know everything and we are just trying to pry something out of them
Project #1

• Consider an iterated function that takes as input a number between zero and one and outputs either

  i) a number that is twice as big if the original result is less than one

  OR

  ii) a number that is twice as big minus one if the original result is greater than one.

The output value becomes the input value for the next iteration.
Bernoulli Map

- Also known as the dyadic transformation or 2n mod 1 problem:

\[
x_{n+1} = \begin{cases} 
2x_n, & \text{if } 2x_n < 1 \\
2x_n - 1, & \text{if } 2x_n \geq 1 
\end{cases}
\]

- Treat as a fixed point iteration with initial condition \( x_0 \).
Bernoulli Map

• Find the initial value(s) that never changes when plugged into the Bernoulli map.
• Find period two and period four orbits. What determines the period of the orbit?
• Given $n$, can you always find a period $n$ orbit? If so, what is it?
• Introduce cobweb diagram and terminology of fixed point iteration
Fixed Point Iterations

\[ x_{n+1} = \begin{cases} 
2x_n, & \text{if } 2x_n < 1 \\
2x_n - 1, & \text{if } 2x_n \geq 1
\end{cases} \]

- Find fixed point(s)
- Fixed point iteration
- Cobweb diagram
- Periodic orbits
Computational Bernoulli Map

$$\begin{cases} 2x_n, & \text{if } 2x_n < 1 \\
\end{cases}$$

- Using type double, the computer will be correct for the first 52 iterations.
- It doesn't matter which periodic orbit they are trying to compute.
- What is going on? Why 52 iterations?
Computational Bernoulli Map

\[
\begin{cases}
2x_n, & \text{if } 2x_n < 1 \\
\end{cases}
\]

- Write function to handle rational numbers
- Display patterns
- Does their code work for irrational numbers?
Conway’s Prime Producing Machine*

17 78 19 23 29 77 95 77 1 11 13 15 15 55
91 85 51 38 33 29 23 19 17 13 11 14 2 1
A B D H E F I R P S T L M N

• The input is 2.
• A step involves multiplying the current number by the leftmost member of the above table which gives a whole number answer.
• Output happens whenever a pure power of 2 occurs.

Conway’s Prime Producing Machine

<table>
<thead>
<tr>
<th>17</th>
<th>78</th>
<th>19</th>
<th>23</th>
<th>29</th>
<th>77</th>
<th>95</th>
<th>77</th>
<th>1</th>
<th>11</th>
<th>13</th>
<th>15</th>
<th>15</th>
<th>55</th>
</tr>
</thead>
<tbody>
<tr>
<td>91</td>
<td>85</td>
<td>51</td>
<td>38</td>
<td>33</td>
<td>29</td>
<td>23</td>
<td>19</td>
<td>17</td>
<td>13</td>
<td>11</td>
<td>14</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>D</td>
<td>H</td>
<td>E</td>
<td>F</td>
<td>I</td>
<td>R</td>
<td>P</td>
<td>S</td>
<td>T</td>
<td>L</td>
<td>M</td>
<td>N</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>825</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>725</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>1925</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>2275</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A</td>
<td>425</td>
<td>B</td>
<td>156</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>B</td>
<td>390</td>
<td>S</td>
<td>132</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>S</td>
<td>330</td>
<td>E</td>
<td>116</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>E</td>
<td>290</td>
<td>F</td>
<td>308</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F</td>
<td>770</td>
<td>T</td>
<td>364</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>T</td>
<td>910</td>
<td>A</td>
<td>68</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A</td>
<td>170</td>
<td>P</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conway’s Prime Producing Machine

• Horribly inefficient
• Importance of using types
  – Using type double without any modifications, the students cannot produce the first prime number
  – Using int isn’t too much better
• Impact of roundoff error
A Rather Normal Project

• A normal number is a number whose digits are equally distributed.
• What is an example of a normal number?
Normal number facts

• The set of normal numbers is dense in the real numbers.
• Easier to find non-normal numbers; focus on values between zero and one.
• Every rational number is eventually periodic and therefore not normal.
• Two irrational numbers have been proven to be normal: Champernowne’s number and the Copeland-Erdos constant.
Last example

- Go to town!

\[ x_{n+1} = \begin{cases} 
  x_n / 2, & \text{if } x_n \text{ is even} \\
  3x_n + 1, & \text{if } x_n \text{ is odd}
\end{cases} \]
Other possibilities

• Chaotic Rabbits
• Fractions of fractions
• Diffusion by chance
• Follow the rules!!
• Getting to the root of the problem
Conclusions

• The previous examples allow the students to go from concept to algorithm to code
• Conway’s Prime Producing Machine and the Bernoulli Map are two examples that help students better understand how computers store real values and effectively demonstrate roundoff error.
• Fun – depending on how you define fun!
Thanks

• Dr. Carla Martin
• Numerous Math 248 students and future Math 235 students