

**AC 2007-1490: "KANGAROO" THINKING: MATHEMATICS, MODELING, AND
ENGINEERING IN INTRODUCTORY COMPUTER PROGRAMMING FOR
ENGINEERS**

Eric Maase, Oklahoma State University

"KANGAROO" THINKING: MATHEMATICS, MODELING, AND ENGINEERING IN INTRODUCTORY COMPUTER PROGRAMMING FOR ENGINEERS

Abstract

Engineering education is concerned with how to transition freshman to successful students whom ultimately become successful engineers in the professional world. With limitations on curricula hours and compulsory courses, many universities are revamping their introductory freshman courses with an eye to give students an impression of what it means to be an engineer. By impressing on students the potential of the engineering field, it is hoped that students will be more motivated to stay in engineering majors. At the same time incorporating math and physics into introductory coursework will help prepare for students for the rigors of advanced computational strategies necessary for successful engineering applications.

The Introduction to Computer Programming for Engineers has been redesigned incorporating a series of assignments which provide the opportunity to teach the aspects of structured programming languages (Visual Basic for Applications, VBA, is the language of instruction) while also allowing the integration of many features of engineering into the course. The approach has proven invaluable both in attracting and maintaining student interest in the course materials while simultaneously improving student understanding and perspectives on mathematics, engineering and modeling. Most important are the implications of how redesigning curriculum can have an impact on raising students' awareness of what it means to be an engineer.

Introduction & Background

Freshmen majoring in engineering at the College of Engineering, Architecture and Technology at Oklahoma State University are required to take an introductory course in computer programming where they are taught the basics of a structured programming language. The computer language in the course is Visual Basic for Applications (VBA), an application-based language provided by Microsoft as a programming interface, interacting with Microsoft Office programs including Microsoft Excel.

The link to Excel becomes an integral part of the course, allowing graphing and object animation as the students learn a programming language syntax and programming structures. Other benefits of VBA include the broad availability of the software to the student, as Microsoft Office Suite is the only required program, the relative ease of moving data to and from Excel worksheets allows students to learn basic programming structures quickly, and a real-time compiler allows faster program development and debugging. VBA also provides the flexibility of building programs that interact with other Windows applications. The VBA language is an introduction to the more formal higher-level Visual Basic and Visual Basic .NET (VB, VB.NET) languages should a student further pursue or need to develop programs that are more complex, stand-alone, or independent.

The instructional methods for this class, learning objectives, and the link to Microsoft Excel have enabled the development of a series of assignments taking advantage of the numerous additional possibilities provided by the VBA environment. Three assignments through the course center on

a particular animal, examples in previous semesters have included mice, kangaroos, hamsters and rabbits – animals to which students can easily identify and relate.

An initial assignment asks students to develop a computer program that calculates the jumping distance of animals. This assignment not only teaches basic programming structures and programming practices, but also provides a method for understanding the limits of mathematical models, the importance of units in model equations, and the need for both a numerical and logical validation of the calculations of any model.

The second assignment asks students to develop a program calculating the necessary take-off angle to jump to a given target. The assignment includes discussions of model limitations as the students become aware of just how many solutions exist mathematically for the problem (an infinite number) and further discusses the need for engineering validation of their work. This assignment exposes students to basic mathematical methods needed for root determination methods and the inherent limitations of such methods.

The third assignment poses as an engineering project proposal to control and maintain populations of animals - the problem itself is given as a project to the students and incorporates ethical complications. The students' goal is the development of a predator-prey population simulation. Students extend and enhance a basic population model introduced in class by incorporating additional effects and influences of their choosing. Simulation results after the model changes require both numerical and engineering validation. While the basic population simulation is not presented as a mathematically complicated system, after working through the more enjoyable and understandable aspects, students become aware that the basic aspects of the problem they have solved - population simulation - are basic mathematical issues in any systems modeling and the methods used in their work are solutions of systems of first order ODEs (initial value problems) being solved by Euler's method. Such systems of equations are common in many engineering disciplines.

'Kangaroo' Thinking

The basic goals of the animal problem series is the gradual improvement of students programming skills, understanding of use of numerical methods, understanding and importance of validation and testing, and the concept of simulation. In each assignment, other aspects of engineering are introduced including units, model limitations, engineering ethics, engineering model uncertainty and legal liability.

Modeling 'Kangaroo' Jumping

The initial assignment asks students to develop a computer program that calculates the distance a kangaroo jumps.

Function Programs and Introductory Programming Principles

Problem: Animal Jumping to specific target distances

Equations: Development of a Model, Unit consistency

Modeling: The development of a Model using Science / Math / Information

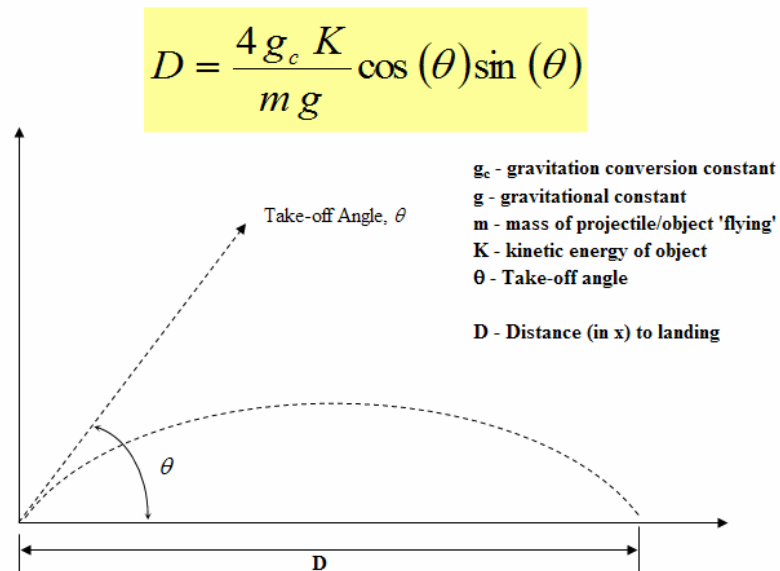
Practical Solutions: Knowns and unknowns, Simple Model Solution

Programming: Function Programs, Input/Output, Variables/Storage

Engineering: Model Limitations, Experimental data, Validation of Results

To begin the basic projectile motion and kinetic energy equations are combined developing a simple 'jump model' of an animal. Model development, limitations, and implementation are discussed in the class lecture. The final model describes jump distance based on an animals' mass, energy, gravity, and jump take-off angle.

Figure 1. A 'Jump' (Projectile) Model



Numerous parameters (values) of the model are unknown and require the students do some basic research to determine how actual Kangaroos behave in order to validate program results.

This initial assignment, in addition to introducing basic programming structures and practices, provides better understanding of the limits of mathematical models, the importance of units in model / equations, and the need for both a numerical and logical validation of calculations for any model. As an aspect of instruction – numerical validation – its importance and meaning are a primary focus.

The Professional / Target Jumping 'Kangaroo'

In the second assignment students develop a program to calculate the specific take-off angle needed to jump to a given target distance.



Problem Solving with Computer Programs

Problem: Animal Jumping to specific target distances

Equation Solving: Root Methods, Units and Research

Modeling: Polynomial Equation unknown solutions

Practical Solutions: Equation Solving, or Root Methods

Numerical Method Background: Newton's / Secant / Bisection Root Methods

Programming: Programming Loops, Tolerance, Division by zero

Engineering: Application, Validation of Results

This assignment extends the previous course materials and introduces the concepts of modular programming along with basic root finding methods. The basics of this assignment require students to develop a program for root finding, which might include any of the following methods: Bisection, Newton's method, and/or the Secant method. To aid in the validation of their own work an Excel workbook both describing Newton's method and providing an animated, interactive solution is made available to the students (Figure 2).

Figure 2. Newton's Method with a VBA program having multiple options

Use the 'spinner' below to select an example function (five available).

$f(x) = 3x^3 + 3x^2 - x - 1$

The graph limits in X are at your discretion

Graph Limits

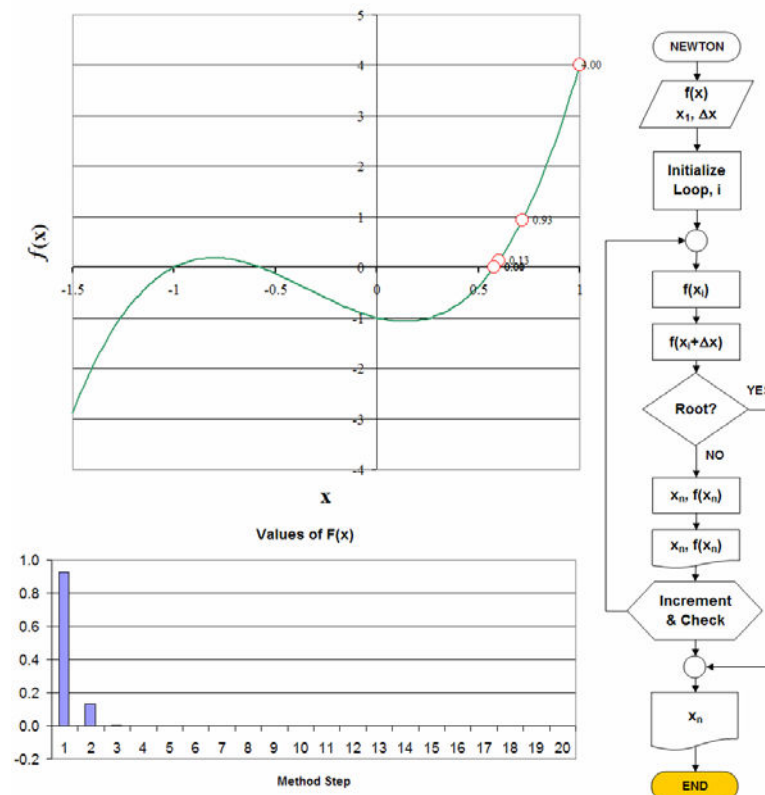
X_{LOW} to X_{HIGH}
 -1.5 1.0

Set X and ΔX . Choose 'Run' to begin a root search.

Run Newtons Solution

X_{INIT} 1.000
 ΔX 0.010

#	X_0	$F(X_0)$	X_n	$F(X_n)$
0	1.0000	4.00000		



This workbook provides notes intended to outline and teach the numerical method involved in root determination, a flowchart of the process from which the student can develop their own programmed solution, and finally also provides one source for validating the students work.

While the assignment includes further discussion of model limitations as the students become aware of just how many solutions exist mathematically for the problem (an infinite number) and the further importance of engineering validation of results.

Figure 3. Animation programs as a teaching tool and to encourage creativity

These Sub programs are found in the Module 'basTEST_ANIM'



ANIMATE #1

ANIMATE #2

IncrementLeft	-50
IncrementTop	25

ANIMATE #3

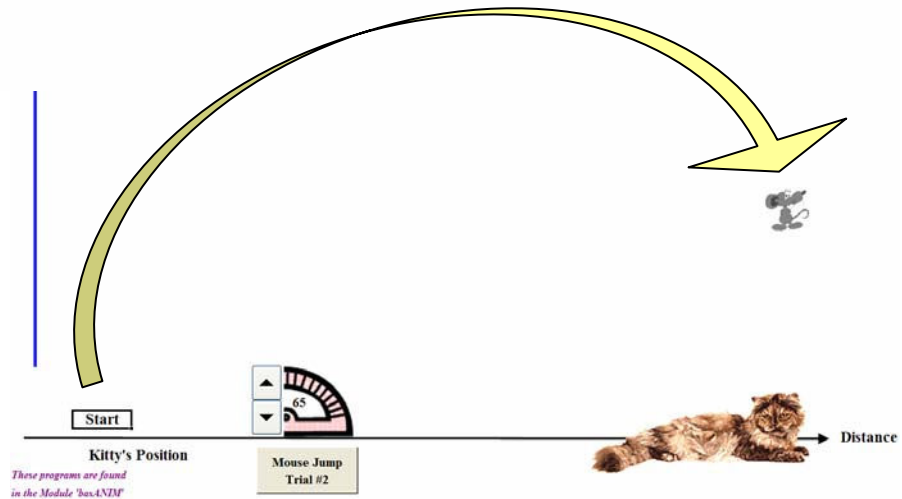
IncrementLeft	1
IncrementTop	1
IncrementRotation	1

ANIMATE #4

In order to involve the students and foster their creative nature the basic possibility of animation is introduced. The concepts required in VBA to animate any object are straightforward and easily outlined – and include the aspect of building from small steps into more impressive effects. The programming principles needed including loops (FOR-NEXT) and decision structures (IF statements) in order to create more complex effects.

Animation also serves as a tool in instruction of the root methods as well by animating graphical solution steps in root determination - an Excel workbook the students are able to use for validation of their work and as notes from which to further learn the method.

The programs above provide the basic aspects of VBA animation of objects and are used in lab instruction.



The jumping mouse above is a more complex example of an animation where an object moves in accordance with the mathematical principles from Assignment#1, the basic ‘jumping’ (Newtonian projectile equations) models. This animation serves in lecture to both reiterate the nature of math, models, and outcomes. The ‘risk’ is the target jumping mouse being consumed by the cat of course. While the lesson may be amusing to the students, and it usually is, the goal is to reinforce that the engineering profession requires testing, validation, and that there are risks.

'Kangaroo' Population Behavior and Control

The third assignment is posed as an engineering problem of controlling and maintaining a population of animals.

System Modeling and Solving Systems of Equations

Problem: Animal Population balances

Modeling: Developing a system of first order differential equations (ODE)

Practical Solutions: Finite-difference equations

Numerical Method Background: First-order Euler's method

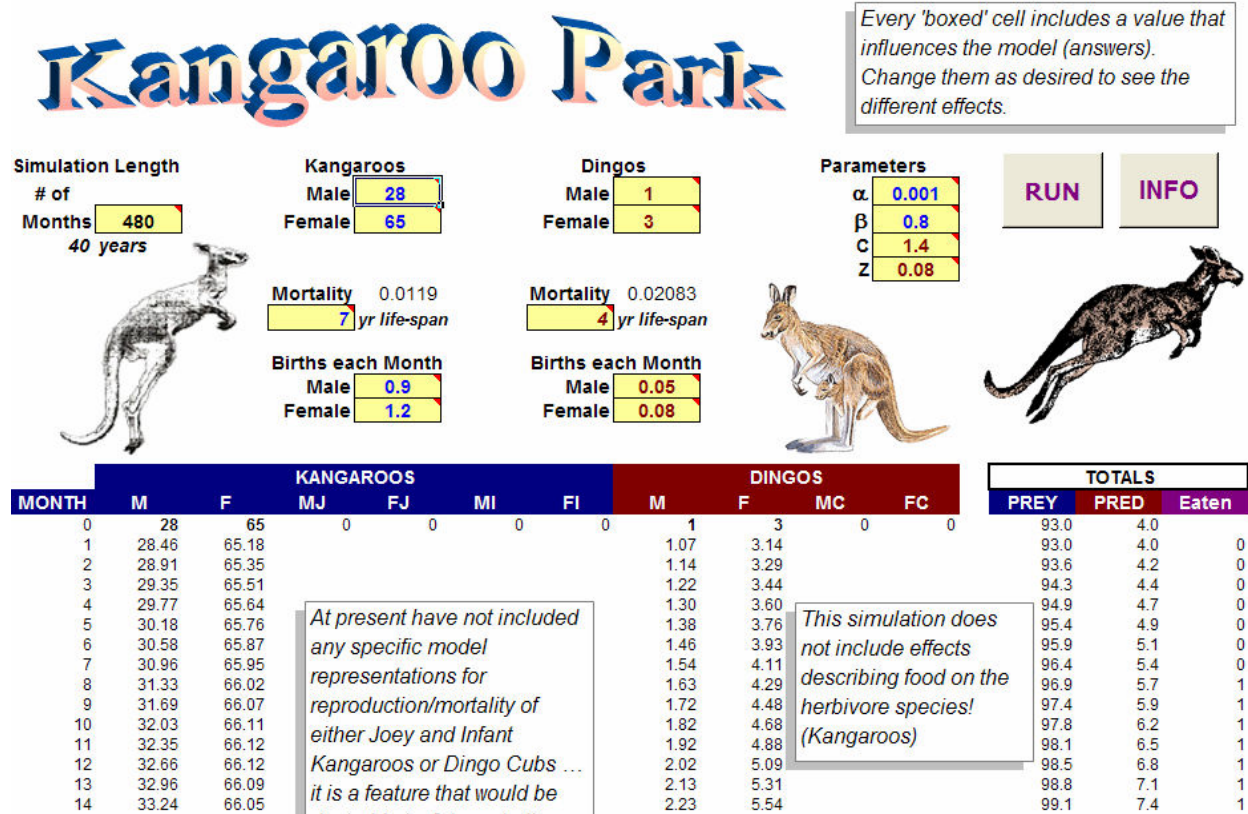
Programming: Arrays, System Description and Simulation, Random values

Engineering: Simulation, Model Changes, Validation of Results, Ethics

The students are provided an introduction to control volumes and material balances (input - output + generation = accumulation) as might apply to animals eating, reproducing, and dying. Now introduce a predator, in this case dingoes, into the system. The programming goal is to develop a simulation of a predator-prey population over time.

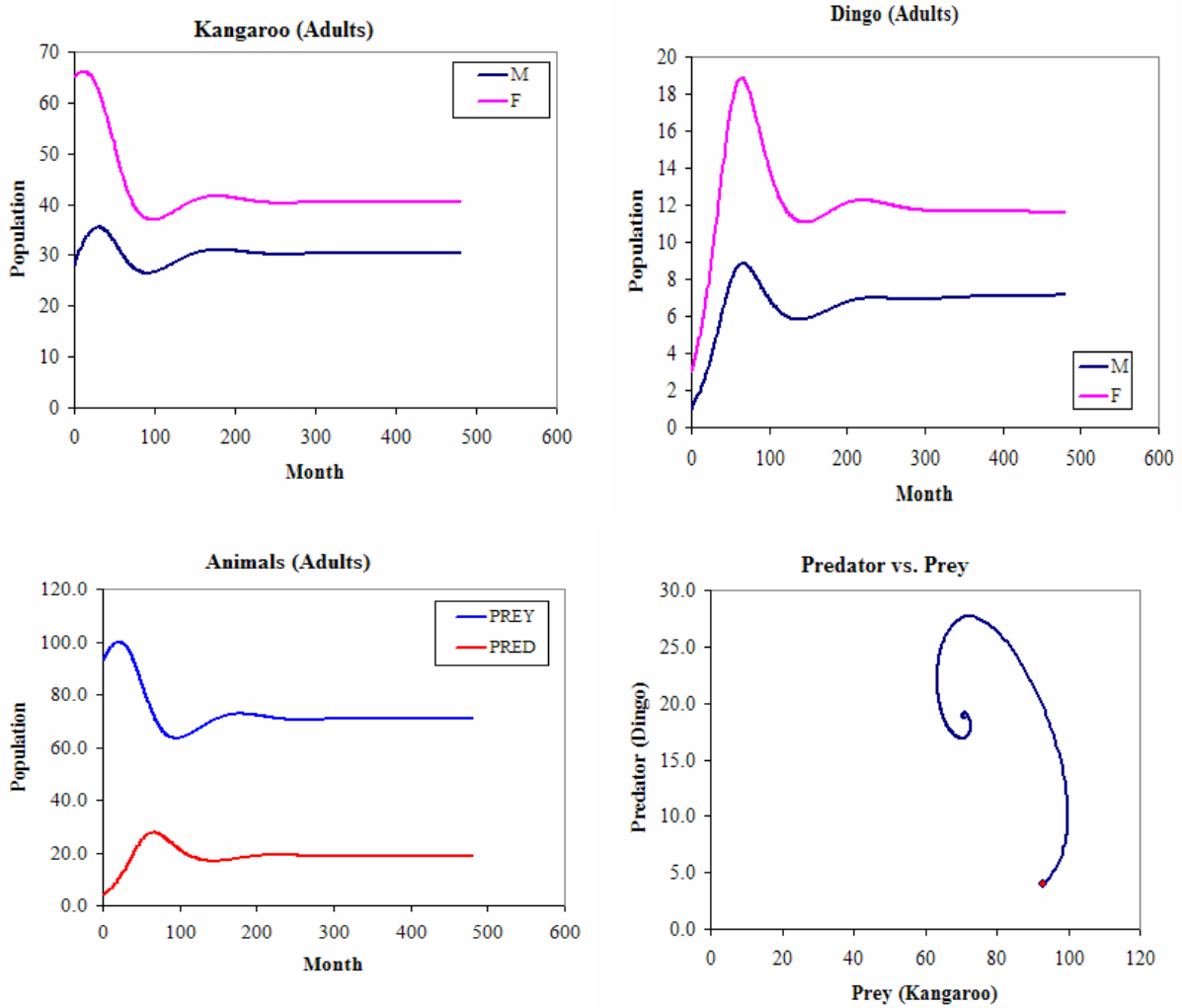
To jumpstart the work a simple solved population model – this will also prove useful in the students' initial validation efforts - is provided. In this assignment the students are asked to extend and enhance the basic model by incorporating additional effects and influences of their choosing. Simulation results after model changes still require both numerical and engineering validation.

Figure 4. Kangaroo Park Simulator



The capabilities of MS-Excel allow graphs (such as shown below) to update as a VBA program runs, allowing students to see system behavior (evolution) based on their parameterization of a mathematical model of a system.

Figure 5. Population Evolution: Predator and Prey



While the above represent the basic model even a simple predator-prey model provides examples of complex behavior that the students must contemplate. The mathematical models provide regions of stability where the systems come to a steady-state condition (as shown), may reach a pseudo-steady-state condition (population cycles through time), or may be unstable (divergent).

Generalized ‘Kangaroo’ Population Simulations

In the third ‘Kangaroo’ assignment, students are tasked with improving the population model through the addition of any effects they consider useful, necessary, or simply interesting (enjoyable).

Model changes enter in two possible methods: changes or improvements to the base model or random external effects on the population(s). The differences are clearer to the students as they make their own decisions, build their models into the program(s) and observe the effects.

Simulation Design and Software Development

Problem: Use of Applied Simulation Programs

Modeling: Unconstrained model application

Practical Solutions: Generalized parameterized programming

Numerical Method Background: First-order Euler’s method, or better method

Programming: Passed data, Simulation, Random values, Interfaces (Forms)

Engineering: Simulation, Model Parameterization, Validation, Ethics / Liability

This assignment (usually completed immediately after / along with the previous assignment) is open-ended. Students both have to consider what effects they are going to incorporate, develop a mathematical description of the effect(s), and then a programming solution of the modeling.

A few of the student changes and/or additions to population models have included:

- Killing half the predators when the population exceeds 10 (fixed time and effect)
- Mad prey disease every 10 years (fixed time, random effects)
- Prey can escape from the ranch (fixed time, random effects)
- Illness effects either / both predators and prey (random time and effects)
- Bacteria, radioactivity (random time, random effects)
- Including other prey/predators as populations (known effects / interaction each week)
- Prey is not in their natural environment (known effects / interaction each week)
- Explicit inclusion of Male & Female genders in calculations (known effects)

The final assignment also tasks the student’s to generalize their computer program and create a user-friendly, well-designed user interface. Essentially the students are designing and building a very basic simulation software package. The course includes a final project where these skills will be used again in developing a software application of the students choosing (software proposal) or a more open-ended assignment (project outlined by the instructor).

The improvement of the population balance ‘project’ as given to students also provides the opportunity to discuss engineering ethics, societal pressures, and engineering risk. While the basis of the problem may not be seen as contemporary engineering work a population balance, and the students’ thinking (as engineering students) provide a context from which to discuss both the ethics of the problem (each population balance assignment deliberately includes an ethical concern – protected species, paid work, questionable legal rights to change/control the population) and additional questions arise in the students improvements to the models.

Engineering Uncertainty – Simulations of Target Jumping

In the final ‘Kangaroo’ assignment, students are tasked with programming a simulation that includes more of the realities of the very first assignment (thus returning them to the place the sequence began) by introducing the concept of reproducibility and uncertainty.

Engineering Equations and Simulation of Outcomes

Problem: Development and Use of programs including and investigating the influence of uncertainty in models

Modeling: Known model application (previous work)

Practical Solutions: Randomized uncertainty

Numerical Method Background: No additional tools

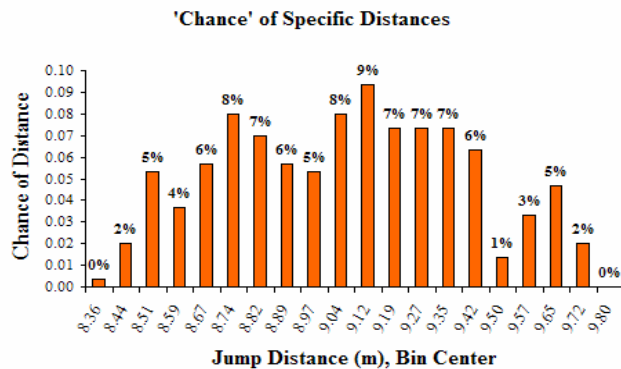
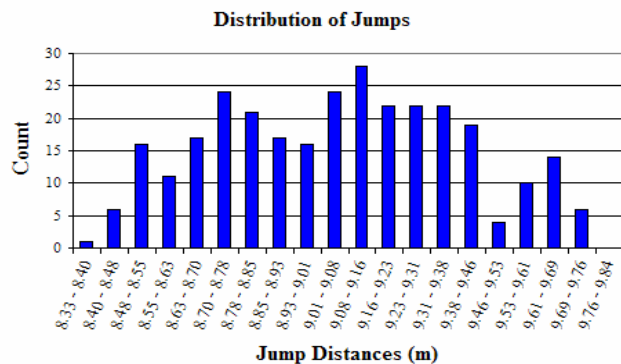
Programming: Loops, IF structures, Random Numbers, Efficient Array Use

Engineering: Simulation, Model Parameterization, Validation of Results

While the original model equation, indeed any mathematical expression may have a ‘right’ answer, engineering mathematics are not as straightforward. The idea of mathematics as a model, and engineering as uncertain, and the computer as a simulation tool are further developed. Figure 6 below shows the real complications that arise in engineering mathematics. The students are being introduced to concepts of engineering and design – specifically uncertainty – through a task that has a goal (precision jumping).

Figure 6. Model Uncertainty and Simulation

Number of Jumps to Attempt		Attempt Multiple Random Jumps	
300			
Bin Limits (m)	Bin Center (m)	Num Events	Chance of Jump
8.33 - 8.40	8.36	1	0.33%
8.40 - 8.48	8.44	6	2.00%
8.48 - 8.55	8.51	16	5.33%
8.55 - 8.63	8.59	11	3.67%
8.63 - 8.70	8.67	17	5.67%
8.70 - 8.78	8.74	24	8.00%
8.78 - 8.85	8.82	21	7.00%
8.85 - 8.93	8.89	17	5.67%
8.93 - 9.01	8.97	16	5.33%
9.01 - 9.08	9.04	24	8.00%
9.08 - 9.16	9.12	28	9.33%
9.16 - 9.23	9.19	22	7.33%
9.23 - 9.31	9.27	22	7.33%
9.31 - 9.38	9.35	22	7.33%
9.38 - 9.46	9.42	19	6.33%
9.46 - 9.53	9.50	4	1.33%
9.53 - 9.61	9.57	10	3.33%
9.61 - 9.69	9.65	14	4.67%
9.69 - 9.76	9.72	6	2.00%
9.76 - 9.84	9.80	0	0.00%



‘Kangaroo Thinking’ Assessments

Over the previous semesters students have participated in an end-of-course survey intended to evaluate both the educational objectives of the course as well as gain insight into student outcomes. The enrollment and the survey responses are shown in TABLEX. At the present time the current semester (Spring 2007) has an enrollment of nearly 200 students.

All surveys are given anonymously via an online course management system, and individual student responses are not linked to grades or assessment for the course.

Table 1. Course Enrollment and Survey Results

	Enrollment	Respondants; % of Students	
Fall 2005	104	54	52%
Spring 2006	117	60	51%
Fall 2006	131	92	70%

Some assessment of the effect of ‘Kangaroo Thinking’ aspects of the course are provided by the results of six survey questions as well as other additional student comments given in open-ended questions asked during the surveys.

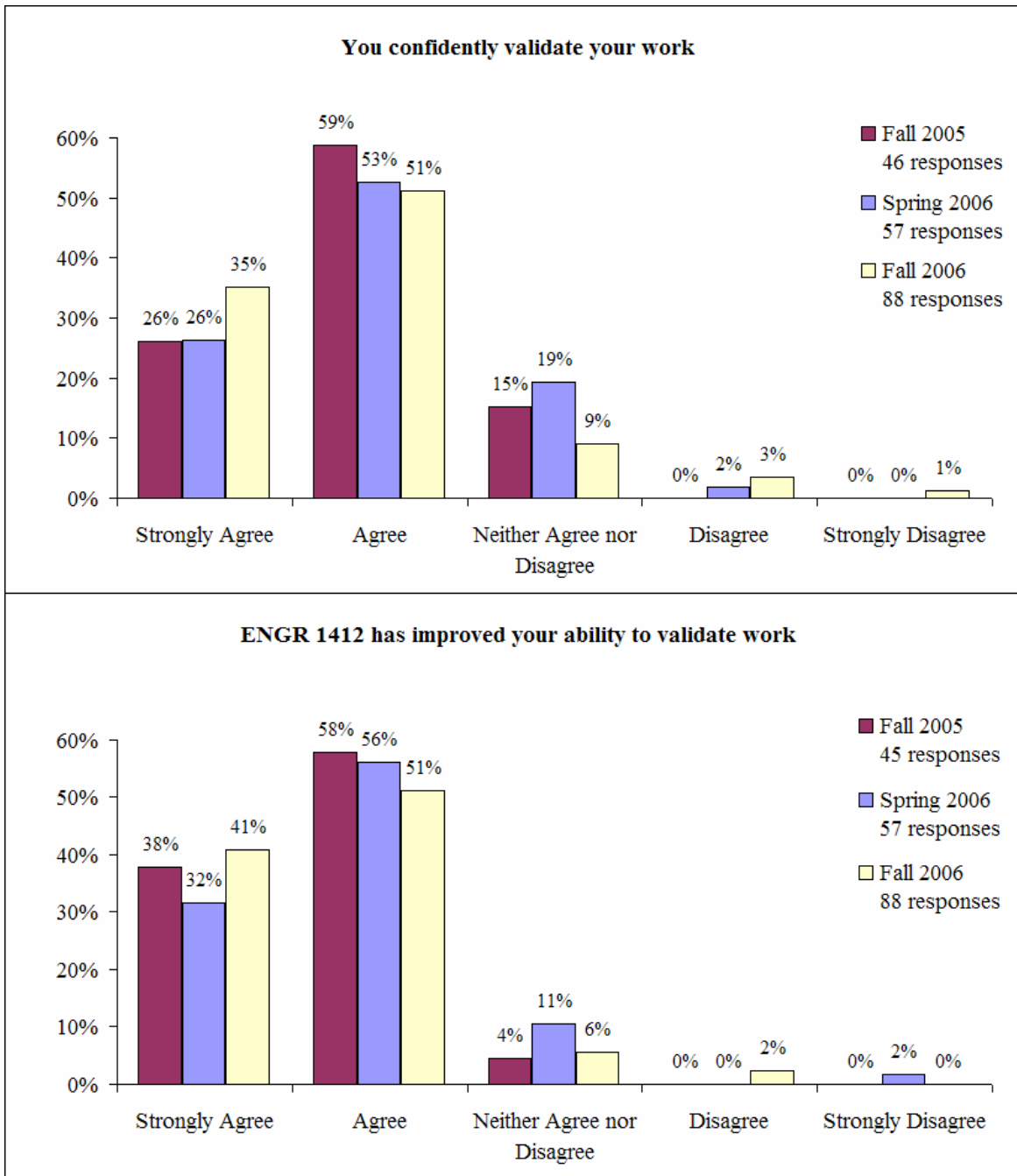
Numerous survey questions were posed as ‘general proficiency’ which was then followed by ‘effect of the current course’ assessments. Three groups of paired Likert format questions on the survey provide a measure the ‘Kangaroo Thinking’ instructional goals.

The first of the three groups assess the concepts of validation as incorporated into the course. While this is a general overall instructional objective which also relates to other aspects of course materials with the intent of both increases students’ awareness and ability to back-up, prove, and be confident in their work and their calculated results. As an overall goal this is an important aim of the course, the results of the surveys are shown in Figure 7.

The students report a significant confidence in both being able to validate their work as well as the course’s role in their improving their understanding of the concept. Overall these positive results indicate ‘Kangaroo Thinking’ may provide a sound system for gaining an effective perspective on mathematics and scientific applications necessary for good engineering.

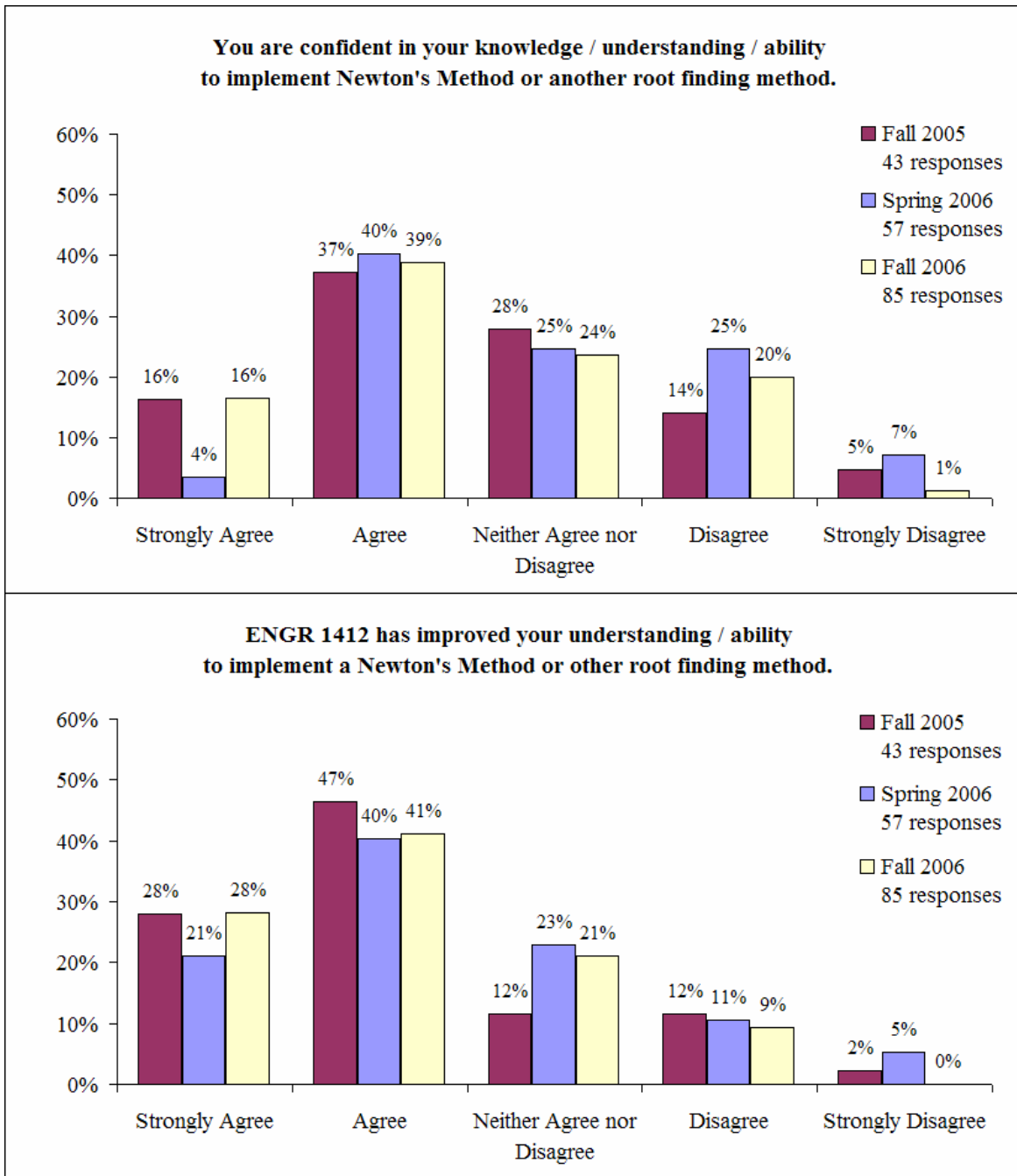
Student surveys of the root methods are shown in Figure 8. The results for root method instruction are less positive, but when combined with other open-ended survey responses it appears that for Freshmen-level students ‘Kangaroo Thinking’ is a more effective approach than simply mathematical/numerical method instruction.

Figure 7. Surveys of "Validation" Goals



While the students indicate an incomplete confidence//knowledge of root finding methods this is not really the specific purpose of the instruction – at this level of coursework simply gaining an understanding and knowledge of the methods is more valuable for students that continue on to later coursework. The secondary effect of the method is also to remove a mathematical anxiety – despite not fully appreciating the theory behind root determination the students overall have applied the theory and coded a programming solution of the method, within the concept of ‘it’s simply a mathematical relationship and procedure’ and not something to be feared.

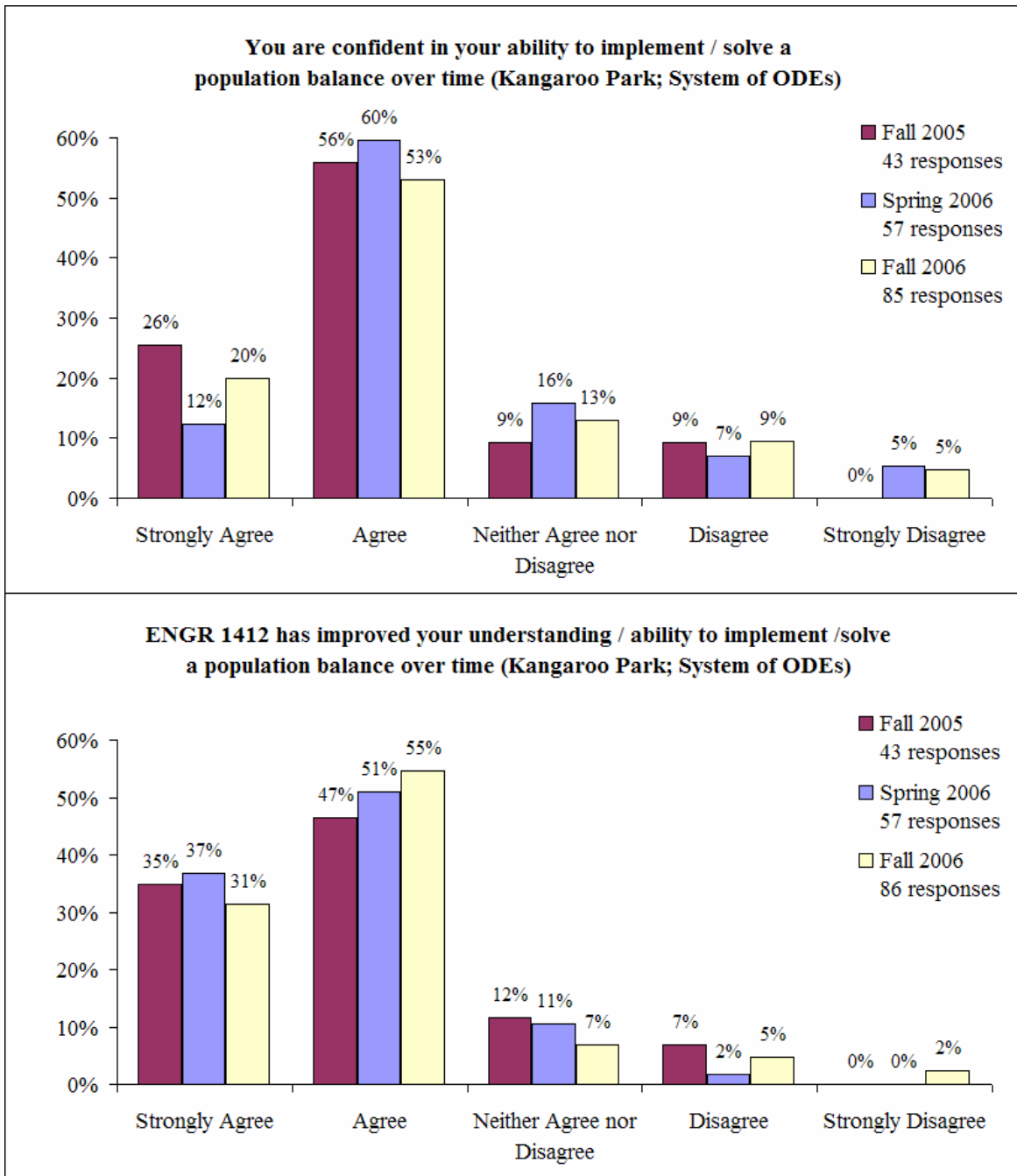
Figure 8. Surveys of "Root Methods"



Despite the lower confidence reports the students still assess that the ‘Kangaroo Thinking’ approach used in the course as aiding in their understanding of the use and need (this conclusion is based on a consideration of additional open-ended survey responses) for root finding methods in engineering.

The surveys regarding ‘systems of equations’ is shown in Figure 9. Surprisingly these results indicate that the ‘Kangaroo Thinking’ approach is quite effective with more than 70% of the students in any semester reporting that they are confident in their understanding and ability to solve a population balance (which they also understand is a system of ODE equations).

Figure 9. Surveys of "ODE Systems Methods"



The instructional influence on the student's confidence is also quite clear showing that 'Kangaroo Thinking' is an effective approach as an introduction to systems of equations and population balances.

Conclusions

In the reorganization of an Introduction to Computer Programming for Engineering course, we can successfully and effectively introduce freshman to concepts including numerical methods, modeling, and computer simulation. By using familiar animals in ways the students may personally relate with, course materials and objectives are more easily integrated allowing the course to form a foundation for large scale learning that will assist the students in later higher level courses and career skills. It is believed that the 'Kangaroo thinking approach' while requiring a lot of work from the students is also rewarding to them as they become personally motivated in the creation and design their own software applications. The approach also appears to reduce math and science anxiety, and effectively give students' an introduction to what it means to be an engineer and a glimpse into the importance of their educational coursework and what professional responsibilities the engineering profession requires.

References

- Bridgwood, M.A., " Guidelines for communication and engineering problem solving at the basic level", *IEEE Transactions on Professional Communication* Vol. 42, No. 3, 1999, pp. 156.
- David G. What Have We Not Learned about Teaching Programming? *Computer* 2006;39(10):81.
- Davenport D. Experience Using a Project-Based Approach in an Introductory Programming Course. *IEEE Transactions on Education* 2000;43(4):443.
- Edward, N., " Evaluations of introducing project-based design activities in the first and second years of engineering courses", *European Journal of Engineering Education* Vol. 29, No. 4, 2004, pp. 491-503.
- Ellermann CR, Kataoka-Yahiro MR, Wong LC. Logic Models Used to Enhance Critical Thinking. *Journal of Nursing Education* 2006;45(6):220.
- Hewett, T.T., and D.V. Porpora, " A case study report on integrating statistics, problem-based learning, and computerized data", *Behavior Research Methods, Instruments, & Computers* Vol. 31, No. 2, 1999, pp. 244.
- Kalkani, E.C., and L.G. Boussiakou, " Observations on an introductory engineering course and suggestions for enhancement", *European Journal of Engineering Education* Vol. 28, No. 3, 2003, pp. 317.
- Kenneth R. Tremblay J, Downey EP. Identifying and Evaluating Research-based publications: Enhancing Undergraduate Student Critical Thinking Skills. *Education* 2004;124(4).
- Kirkwood, M., " Infusing higher-order thinking and learning to learn into content instruction: a case study of secondary computing studies in Scotland", *Journal of Curriculum Studies* Vol. 32, No. 4, 2000, pp. 509-535.
- Lauer T. Teaching Critical-Thinking Skills Using Course Content Material. *Journal of College Science Teaching* 2005;34(6):34.
- Stover S, Mabry M. Merging Science and Society. *Journal of College Science Teaching* 2005;34(4):40.
- Susan, M., " Strategies for Teaching Modeling to Students", *Bioscience* Vol. 56, No. 4, 2006, pp. 299.
- Townend MS. Integrating Case Studies in Engineering Mathematics: a response to SARTOR 3. *Teaching in Higher Education* 2001;6(2):203-215.
- Vacaretu A-S. Mathematics: A Fresh Perspective. *Thinking Classroom* 2005;6(1):27.