AC 2007-1383: A STUDY ON THE EFFECTS OF TIMING ON ENGINEERING STUDENTS' ABILITIES TO SOLVE OPEN-ENDED PROBLEMS WITH COMPUTERS

Veronica Addison, University of South Carolina
Veronica Addison is a Ph.D. Candidate in Mechanical Engineering at the University of South Carolina and a former GK-12 Fellow. Her research and teaching interests include sustainability, the built environment, energy and environmental design.

Christian Hipp, University of South Carolina
Christian Hipp is a Ph.D. Candidate in Philosophy at the University of South Carolina. His research interests include applied ethics, demarcation criteria in genetic domains and convergent technologies.

Jed Lyons, University of South Carolina
Jed Lyons is a Professor of Mechanical Engineering and the Faculty Director of the Center for Teaching Excellence at the University of South Carolina. His passion is engaging learners of all ages in the processes of inquiry and discovery through active engagement and problem-situation learning.

A Study of the Effects of Timing on Engineering Students' Abilities to Solve Open-ended Problems with Computers

Abstract

This paper presents the design and preliminary results of an exploratory research project to determine the best ways to introduce computer algebra and symbolic manipulation software into the early undergraduate mechanical engineering curriculum. This paper discusses one component of the exploratory project that focuses specifically on how the timing of introducing MathCAD affects student attitudes and performance in a sophomore-level numerical methods course at the University of South Carolina. An experiment was conducted in the Fall semester of 2006 with a class of sixty students. The class was divided into two groups that received differentiated instruction at four times during the semester. The experimental group completed a computer assignment before going to lecture; the control group heard the lecture and then completed the computer assignment. Qualitative data was collected on each group by three participant observers who followed the students daily, and also through focus groups and interviews conducted by the authors. Quantitative data included student performance on the computer assignments, subsequent quizzes and other graded assignments. A preliminary analysis of the qualitative and quantitative data suggests that the students in the experimental group were less happy, but tended to perform better on some assignments, than those in the control group.

Background

Considerable research has been conducted on students' attitudes and abilities related to computers[1-7], and on the use of computers as supplements or extensions to more traditional teaching modes[8-13]. Some research has also been conducted on the role of the computer in developing student problem solving skills[14-18]. However, the authors are aware of only one study that deals specifically with the effect of how timing the introduction of a computer tool affects learning. Apkan[19] compared computer simulation of dissection of frog with actual dissection and reported that a simulation used before dissection led to better achievement performance than a simulation used after dissection. The research related in this paper contributes to our understanding of how timing the introduction of the computer as a solution tool affects student performance.

The context for this study is a numerical methods course for mechanical engineers at the University of South Carolina. This particular course involves extensive use of the software program MathCAD. The application of MathCAD software is of interest to many in the engineering education community. For example, the use of MathCAD is reported in over three hundred papers in the Annual Conference Proceedings of the American Society of Engineering Education between 1996 and 2006. In most cases, MathCAD is used as a course enhancement: students solve problems with the software after they have learned the theory, methodology, or concept being taught. This approach has been traditionally followed in our numerical methods course, also. However, in the Fall semester of 2006, an exploratory research project was initiated to investigate alternative approaches.

The Numerical Methods Course

This research is conducted in the context of EMCH 201 - Numerical Methods, a three-credit required sophomore-level course for mechanical engineers. EMCH 201 is an introduction and application of numerical methods to the solution of physical and engineering problems. Techniques include iterative solution techniques, methods of solving systems of equations, and numerical integration and differentiation. Topics covered include:

- <u>Background and Numerical Methods:</u> Significant figures, accuracy versus precision, error, round-off error, truncation error, Taylor series.
- <u>Linear Algebra:</u> Systems of equations, vectors, matrices, rank, determinant, matrix algebra and decomposition.
- <u>MathCAD:</u> Fundamentals, help screens, variables, range variables, graphing, insertion, deletion, functions, units, programming.
- <u>Systems of Linear Equations:</u> Applications of constrained systems.
- <u>Roots of Equations:</u> Graphical, bisection, Newton-Raphson, applications, optimization.
- <u>Programming Constructs:</u> Do/For loops, conditional statements, MathCAD programming.
- <u>Curve Fitting and Interpolation:</u> Linear regression, general least squares, power law and exponential applications, linear interpolation, quadratic and spline interpolations.
- <u>Numerical Integration and Differentiation:</u> Trapezoidal rule, Simpson's rule, finite difference schemes.

Student performance is evaluated by homework, written quizzes and exams, computer-lab worksheets and a computational project. Attendance at the two weekly 75 minute class sessions is not mandatory.

Research Design

The research reported here is part of a NSF funded, exploratory research project from the Course, Curriculum, and Laboratory Improvement Program (CCLI). The broad objective is to determine the best ways to introduce computer algebra and symbolic manipulation software into the early undergraduate mechanical engineering curriculum, specifically numerical analysis. This project aims to improve student computing abilities by moving students away from 'cookbook' approaches and finding ways to develop creative problem solvers. This will allow students to develop skills that will enable them to deal with more complex problems as engineers in the workplace. The NSF project's research questions are:

1. What computer experiences do students have when they enter college-level engineering class?
2. In what ways does varying the timing of the introduction of computer techniques affect students' expectations and creative use of these methods?
3. In what kinds of problems does the computer specifically enhance understanding? In what kinds of problems does the computer act as an obstacle to understanding?
4. How can we emphasize the importance of setting up problems for computer-aided solutions instead of emphasizing the results of the process?

In the Fall 2006 semester, a sixty-student class of EMCH 201 was divided into two sections in order to investigate research question 2 (the effect of timing). Originally, all students signed up for a single section. One of the authors then obtained each student's GPA, credit hours earned, Calculus grades and demographic data. Based on this information, two sections were created. The class list was sorted by GPA and grade in Calculus I in decreasing order, and every-other student was assigned to section 1 or 2, accordingly. The GPA of one transient student was unknown. The distribution of students by grade in Calculus I included eleven students with advanced placement (AP) credit and ten with no grade available to the authors, due to their status as transfer students. As shown in Figure 1, each section had similar groups of students in terms of their previous academic achievement and background.

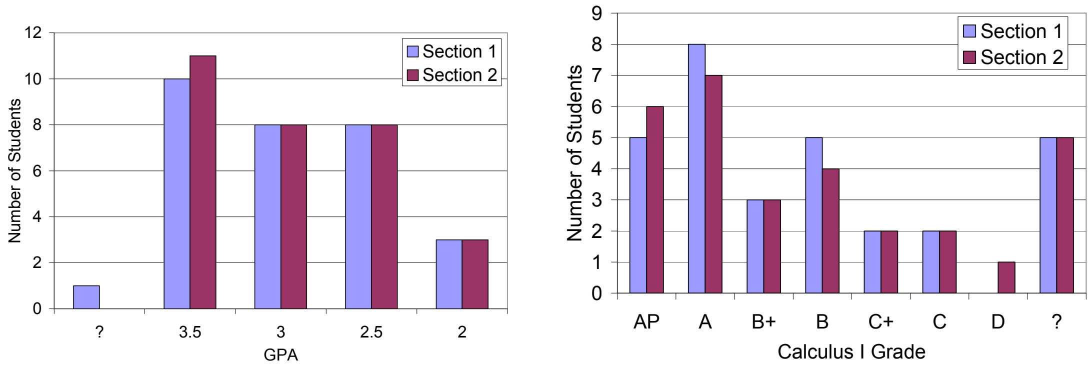


Figure 1. The distribution of grade point average (GPA) and grade in Calculus I for the two sections of EMCH 201 were similar.

Both sections were taught by the same professor and evaluated with the same methods. For most of the semester, both sections were taught at the same time in a sixty-person classroom. However, the two sections received differentiated instruction at four times during the semester. During each of these four weeks, the topics covered were:

1. Linear Systems
2. Non-linear Equations
3. Programming
4. Curve Fitting

For each of these topics, section 2 completed a computer assignment before going to lecture; the section 1 heard the lecture and then completed the computer assignment. Section 1 is therefore considered as the control group because MathCAD use has traditionally followed lecture in EMCH 201. Section 2 is considered the experimental group.

Details of the Differentiated Instruction

The four topics taught using differentiated instruction each had the same main components: lecture, computer assignment, an in-class quiz, and a test. Upon beginning a new topic, section 1 was initially presented with chalkboard lectures followed by in-class examples. In the next class period, the topic continued with a larger scale problem that could be implemented in a computer-based worksheet. For section 2, a new topic was initially introduced graphically on the computer and illustrated as a problem in the worksheet. The computer-based worksheet was followed by a lecture in the next class period, which explained the processes going on behind the scenes of the computer program. In the third class period, both sections attended together and took a short in-class quiz. The topic culminated with a test taken in class by both sections.

For example, during the topic of "Non-linear Equations" students in section 1 are introduced to the material through theory, equations and example problems. Students in section 2 are introduced to the topic graphically as a root-finding problem. In the computer-based worksheets, roots are defined graphically and built in root-finding packages are explained. Non-linear equations are then introduced in the context of root-finding. The MathCAD Worksheet for this topic is included as Appendix A. The text shown in italics are questions posed to the students, the highlighted portions of the worksheet represent the solutions that the student should provide.

Research Methods

A mixed-methods research data collection plan was used. Two graduate students and an independent professor were used as participant-observers who sat-in on every class or computer session. They keep class-by-class records of the students' classroom questions and reactions, as well as their own impressions of the student class experiences and responses to instructional choices. The participant observers also keep records of the computer lab experiences, noting student questions, student-instructor interaction, teamwork, and responses to assignments. Student skills are also assessed through analysis of computer-based worksheets, written and computer based homework assignments, a final project, quizzes, and written tests. Separate focus groups conducted mid-semester allowed students in both groups to voice their opinions regarding how having the lecture prior to or after the computer-based worksheets affects student understanding. Exit interviews were also conducted at the conclusion of the course after the final exam. Learning styles of the students were assessed using the Felder-Silverman Index of Learning Styles[20].

Research Results

A preliminary analysis has been performed on the data collected by the participant observers, during the mid-semester focus groups, and with grades on student assignments. The participant observers' field notes indicate that the students in Section 2, which performed the computer lab worksheet before receiving instructions in lecture, were more likely to pay attention during the subsequent lecture and asked better questions of the instructor. Students in section 1, the control group, were more likely to passively take notes during lecture.

An assessment of the focus group recordings indicates that the students in section 2 were not happy. They expressed a much higher level of frustration with the computer exercises than the students in section 1. The students in section 2 felt as if they were wasting time in the computer lab because they did not understand why they were doing the worksheet. Almost all of the students in section 2 wished that they were in a more traditional learning environment. Very few students in section 1 expressed a desire to be in the experimental environment. The perception of students in section 2 was that their grades were lower than the students' grades in section 1.

However, Table 1 below shows preliminary results of student performance based on average grades for various course components. It can be noted that section 2, performed significantly better than section 1 on worksheets, quizzes and the project. The implication is that when the students in section 2 were forced to struggle in the computer lab before receiving instructions in the lecture, they learned better. Further analysis of the data, including the effects of student learning styles on individual performance, will shed additional light on this statement.

Table 1: Average Grades

	Section 1 Lecture First	Section 2 Computer First
Worksheet Average	78%	89%
Quiz Average	59%	66%
Homework Average	87%	88%
Test Average	71%	74%
Project	88%	95%
Final Exam	67%	72%

Concluding Remarks

This research project explores how engineering students learn to solve problems with computers. Preliminary qualitative and quantitative data suggests that the students in the experimental group were less happy, but tended to perform better on some assignments, than those in the control group. The results are promising and suggest that future study is merited, particularly through analysis of the learning styles, grades, and classroom participation of individual students in the courses. These descriptive findings can provide a basis for making changes to the course to improve students' ability to learn numerical methods.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant Number 0536660. All opinions expressed within are the authors' and do not necessarily reflect those of the National Science Foundation. The authors also wish to thank Dr. Sarah Baxter for allowing this research to be conducted in her class.

Bibliography

1. Harrington, Jan L. (1988). The Computer Background of Incoming Freshmen: Looking for Emerging Trends. New York, NY: *ACM SIGCSE Bulletin*. 20 (1). pp 210-214.
2. Harrington, Jan L. (May 1990). Changes in the Computer Background of Incoming Freshman. *Collegiate Microcomputer*. v8. n2. pp 147-154.
3. Mawhinney, Charles H. An Instrument for Assessing the Computer Background of Entering College Freshmen (1992). In: *Proceedings of the 1992 ACM Special Interest Group on Computer Personnel Research (SIGCPR) Conference.* Cincinnati, Ohio. pp 309-313.
4. Johnson, Donald M., Ferguson, James A., Lester, Melissa L. (2000). Computer experiences, self-efficacy and knowledge: A comparison of freshman and senior agriculture students. *2000 ASAE Annual Intenational Meeting, Technical Papers: Engineering Solutions for a New Century*. vol: 2 2000. pp 5541-5550.
5. Stone, Jeffrey A., Hoffman, Mark E., Madigan, Elinor M., Vance, David R. (2006). Technology Skills of Incoming Freshmen: Are First-Year Students Prepared? *Journal of Computing Sciences* in Colleges. Volume 21. Issue 6. pp 117-121.

6. Palaigeorgiou, G. E., Siozos, P. D., Konstantakis, N. I.’ Tsoukalas, I. A.(2005). A Computer Attitude Scale for Computer Science Freshmen and its Educational Implications. Journal of Computer Assisted Learning, Oct2005. Vol. 21 Issue 5. pp 330-342.
7. Bruce Harley. (2001). Freshmen, information literacy, critical thinking and values. *Reference Services Review*, Volume 29, Number 4 (2001), pp. 301-306, <http://ejournals.ebsco.com/direct.asp?ArticleID=X14DA5K19Y06K66M2KV8
8. Wing, Jeannette M. (2005). Computational Thinking. *Communications of the ACM*. Mar 2006. Vol. 49. Issue 3. pp 33-35.
9. Roger Azevedo. (2005) Using Hypermedia as a Metacognitive Tool for Enhancing Student Learning: The Role of Self-Regulated Learning. *Educational Psychologist* 40:4, pp 99-209.
10. Roger Azevedo. (2005) Computer Environments as Metacognitive Tools for Enhancing Learning. *Educational Psychologist* 40:4, 193-197
11. vom Hofe, Rudolf. (Sep 2001). Investigations into Students' Learning of Applications in Computer-Based Learning Environments**.** *Teaching Mathematics and Its Applications*. v20 n3. pp 109-19.
12. David E. Meltzer, Inquiry-Based Instruction for Elementary Physics: High-Tech and Low-Tech. Accessed at http://www.physics.iastate.edu/per/talks/jacksonville99.pdf on September 5, 2006.
13. Sally James. Using Computer Technology to Complement Inquiry-Based Learning in the Science Laboratory. Accessed at http://www.ltu.unsw.edu.au/content/userDocs/gpc_1_james.pdf on September 5, 2006.
14. Daniel Joyce. (March 1998). The Computer as a Problem Solving Tool: a Unifying View for a Non-Majors Course. ACM SIGCSE Bulletin. Proceedings of the Twenty-ninth SIGCSE Technical Symposium on Computer Science Education SIGCSE '98. Volume 30 Issue 1.
15. de Jong, Ton. (2006). Technological Advances in Inquiry Learning. Science. Vol.312. Issue 5773. pp 532-533.
16. Diederen, J., Gruppen, H., Hartog, R., Voragen, A. (2005). Design and Evaluation of Digital Learning Material to Support Acquisition of Quantitative Problem-Solving Skills within Food Chemistry. Journal of Science Education & Technology. Vol. 14 Issue 5/6, pp 495-507.
17. Chang, Kuo-En, Sung, Yao-Ting, Lin, Shiu-Feng. (2006). Computer-assisted learning for mathematical problem solving. Computers and Education. Vol 46 Issue 2. pp 140-151.
18. Donald Merino, K. Abel (2003) “Evaluating the Effectives of Computer Tutorials versus Traditional Lecturing in Accounting Topics” Journal of Engineering Education, 189-194
19. Apkan, J. P. (2002) Which Comes First: Computer Simulation of Dissection or a Traditional Laboratory Practical Method of Dissection. *Electronic Journal of Science Education* v. 6 no. 4 (June 2002) pp. 1-20.
20. Felder-Silverman Index of Learning Styles. Accessed at http://www.ncsu.edu/felder-public/ILSpage.html on November 28, 2006.

ORIGIN := 1

Type ORIGIN:= 1 at the above this text box. This will tell Mathcad to start counting elements of vectors and matrices with 1 rather than 0.

GRAPHICAL DEFINITION OF A ROOT

Roots are where functions cross the x-axis (where y = 0. The easiest way to locate a root of an equation of one variable is to graph it. This means that if we write the formula as F(X) = 0 we are looking for the value of X that makes this true. All root finding problems can be written as F(X) = 0

Define and Graph the following functions on separate graphs. Make sure that the domain (range of x values) and range (range of y values) show where each cross the x-axis.

I suggest selecting the cross axis style for root finding, it puts the axis at zero.

a. f(x) = cos(x), show the first positive root.

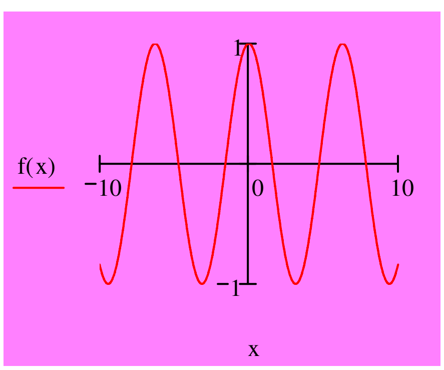
b. g(x) = 3x-5

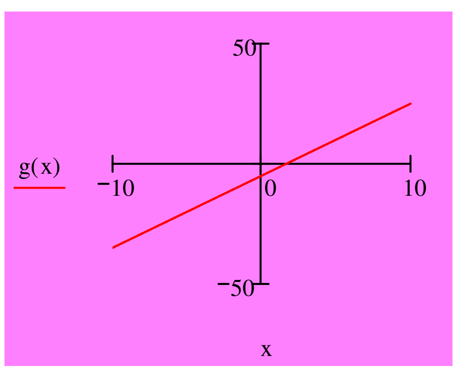
c. h(x) = cos(x)-x

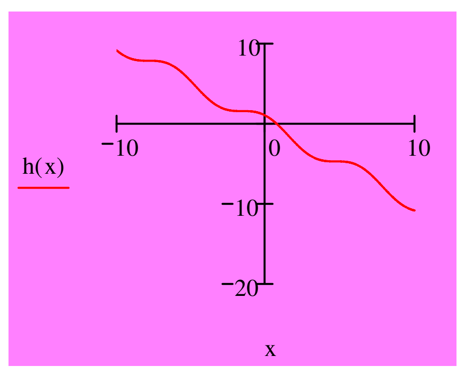
$f(x) := \cos(x)$

$g(x) := 3 \cdot x - 5$

$h(x) := \cos(x) - x$







BUILT IN PACKAGES TO ROOT FIND

MathCad has one called **roots.** This takes as its arguments (input) the function that you want to find the root of and an **initial guess.** This means that you might have to graph it or try a few values to get in the neighborhood.

root

$ff(x) := e^{-x} - x$ Define the function

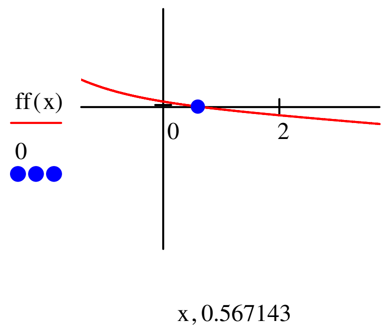
$x := .66$ Initial Guess

$a := \mathrm{root}(ff(x), x)$

Solve

$a = 0.567143$

$ff(a) = 0$ Check



Solve for the root of the following equations, ***graph*** *each one to determine an initial guess, check the accuracy of each one by plugging back into the equation.*

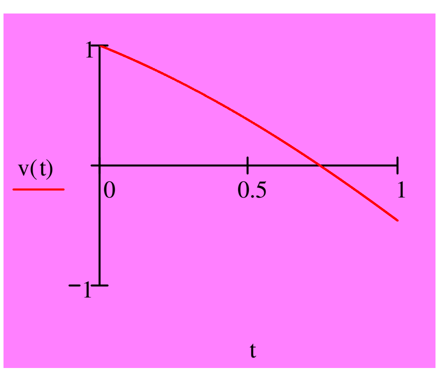
a. v(t) = cos(t)-t

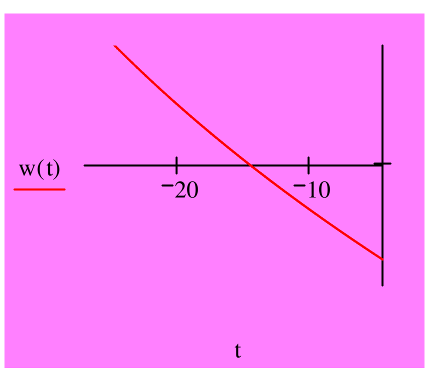
b. w(t) =

$$2000 \cdot \ln\left(\frac{150000}{150000 + 2700 \cdot t}\right) - 9.8 \cdot t - 750,$$

$v(t) := \cos(t) - t$

$$w(t) := 2000 \cdot \ln\left(\frac{150000}{150000 + 2700 \cdot t}\right) - 9.8 \cdot t - 750$$





$t1 := 0.739$

$t2 := -14.5$

$root(v(t1), t1) = 0.739085$

$root(w(t2), t2) = -14.550991$

$v(0.739085) = 0$

$w(-14.550991) = -0.000028$

WHEN DOES ROOT-FINDING COME UP?

Root finding can be a straight question of where a function crosses the x axis. In general this is only a problem when the function is non-linear.

But most **non-linear equations** can also be rewritten as a root finding problem.

If you put any equation into the form F(X) = 0 it becomes a root-finding problem and you can use the programs above.

For example the equations written below on the left are rewritten as root-finding problems on the right.

NON-LINEAR EQUATIONS

General root-finding form

$\cos(x) = x$ $\qquad$ $\cos(x) - x = 0$

$e^{x} - 1 = x$ $\qquad$ $e^{x} - 1 - x = 0$

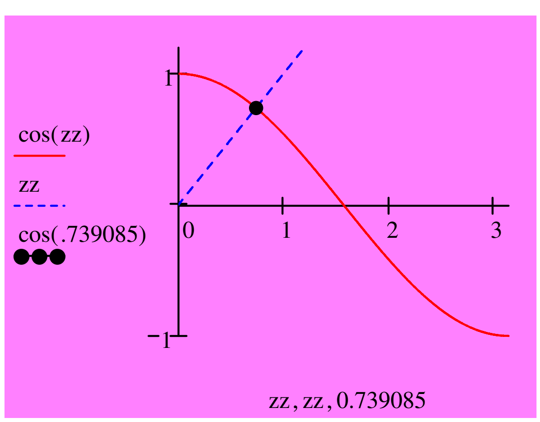
$F = \frac{R \cdot T}{v - b} - \frac{a}{v \cdot (v + b) \cdot \sqrt{T}}$ (the only unknown is v) $\qquad$ $F - \frac{R \cdot T}{v - b} - \frac{a}{v \cdot (v + b) \cdot \sqrt{T}} = 0$

INTERSECTIONS

Root finding can arise when you want to know where two curves intersect,

y1 = cos(x) and y2 = x translates into y1 = y2 or cos(x) - x = 0. Find the

intersection of these two lines; plot the two curves and the intersection point



$hh(x) := \cos(x) - x$

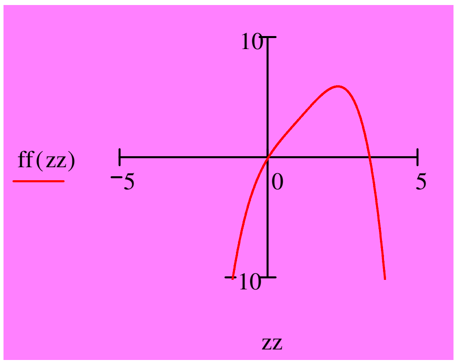
$x1 := .5$

$root(hh(x1), x1) = 0.739085$

OR OPTIMIZATION

or when you want to know
where a function has a **maximum or minimum, BECAUSE**
functions have an optimum or extremum where their **derivative is zero**!!
Plot the following function:

$$ff(z) := 4 \cdot z - 1.8 \cdot z^2 + 1.2 \cdot z^3 - 0.3 \cdot z^4$$



We want to find where the derivative of f(x) is zero so find the derivative

$$gg(z) := 4 - 3.6 \cdot z + 3.6 \cdot z^2 - 1.2 \cdot z^3$$

Where the derivative has its root, is where the function has a maximum.

Find the root, and plot the root and the maximum

$x2 := 2$

$root(gg(x2), x2) = 2.326352$