A Study on the Use of Knowledge Representation for Teaching Engineering Problem Solving

Gustavo J. Molina¹

Abstract - The author introduced in a sophomore design class the Fuller-Polya Diagram (FPD) for problem solving, a simple structured method for knowledge representation of deterministic engineering algorithms. The FPD proposes a formal language and graphics approach when the equations or procedures are known to exist, but they may be unknown (e.g. yet to be determined) or non-mathematical (e.g., meta-operations). A study is presented on the students experience when solving problems with and without the help of the FPD methodology; this work indicates the feasibility of introducing the FPD at the sophomore level. A standard questionnaire is used to evaluate the method and to survey students' opinions on the usefulness and helpfulness of the learning experience. Remarks for teaching the FPD and for further study are discussed.

Keywords: problem-solving, Fuller-Polya diagram, algorithm representation, knowledge mapping.

INTRODUCTION

A project-based teaching methodology is employed by the author in the sophomore class ENGR2431 Creative Decisions and Design. This Georgia Southern University course is part of the Georgia Tech Regional Engineering Program (GTREP) and Regents Engineering Transfer Program (RETP) for Mechanical Engineering majors. This type of project-based undergraduate courses is indicated [1] in freshman and sophomore curricula to early apply creative thinking and problem solving in engineering design. These classes also may serve as student retention tools in engineering programs. A number of innovative teaching techniques has been used for such purposes; the so-called "studio" methods are particularly successful to enhance student creativity and problemsolving skills [2].

In the author's class the taught problem-solving techniques integrate in team or individual design projects. The semester-class meets twice a week for 50-minute lectures, and once a week for three-hour studio-lab in an appropriate classroom for hands-on work. Students also have access to other labs for other activities (i.e., machining lab, robotics and measurement lab, etc.)

The three levels of design-problem complexity are introduced in the class, e.g., poorly defined problems, open-ended creative-design problems, and routine computation procedures (as element of machine computations presented in manuals, catalogues, standards and upper-level textbooks). Sophomores can readily integrate and apply the basic techniques for the first two design problem-types (e.g., brainstorming, QFD (Quality Function Deployment), generation of alternatives, morphological method, TRIZ (a Russian acronym for "the method of inventive solutions"), see reference [3]. But routine computation procedures can be surprisingly difficult for sophomores. One reason for such difficulty may be that such procedures combine standard math computations (where component-specific knowledge is assumed) with selection methods from lists or tables, and data intervals rather than constant values.

The experienced engineer can deal with such computations even in the absence of some data. But these routine engineering algorithms are difficult for students, particularly if they are not presented as straight "plug-data" computations (i.e., if formulas or procedures are not straightforward explicit computation of the required results from the given data). The lack of algorithm insight and manipulation abilities in college students has been the focus

¹ Department of Mechanical and Electrical Engineering Technology, Georgia Southern University, PO Box 8045, Statesboro, GA 30460-8045, gmolina@georgiasouthern.edu

of recent education research, because it can be a critical factor for problem-solving development in advance classes. One such lack of abilities is symbol manipulation. As an example, the author and colleagues [4] studied the "disconnection" between the classic mathematics teaching with "x and y" to the use of other variable names in engineering courses. They determined that such disconnection is a significant factor in students' failure with problem solving. It seems that students largely use "formula pattern matching" to the used symbols instead of to the implied concepts. It also seems that college students see little relationship between symbolic manipulations in math to the type of concept-manipulations required in engineering or physics problems [4].

Redish et al. [5] noted that, in a typical calculus-based physics class the equations shown in the first week had from three to six symbols or more and they specified a connection with something physical. But equations in Calculus classes typically include just two symbols. They also noted that students in introductory physics have a strong inclination to put numbers into equations as soon as they know them: a clear explanation is that this makes the equations look more like the one-variable expressions in their math classes. However, by putting numbers in at the beginning students do not realize an equation as a relationship between actual measurements. They also miss the chance of realizing the whole ensemble of results for different combinations of data.

Experienced professionals (e.g., engineers, physicists, etc) use their knowledge to see "meaning of engineering things" (or of "physics ones") in the "symbols of math" when they interpret equations [6]. College students often miss and do not use for problem-solving the meaning carried by used symbols or equations. Redish et al. [6] observed six different types of difficulties with assigning meaning to the mathematics in a problem context. These difficulties showed in failures to (a) relate symbols to measurement, (b) note variations associated to similar symbols or equations, (c) understand equations as relationships, (d) treat equations as representations of reality, (e) be able to parse equations and (f) assign consistent coordinate references to time and space. The first five relate to the work of this paper.

Tuminaro et al. [7] identified the the so-called "plug-and-chug" cycle as being most common student problem-solving scheme. This recursive epistemic scheme starts by identifying an equation that could solve the problem, to plug in it the available data. If such expression does not yields the result (or if there are no sufficient data), a new formula is sought and the cycle continues in the hope of a solution. Klebanoff et al. [8] discussed how the type of work that students are asked to perform in mathematics classes does not prepare them for applying mathematical concepts in engineering contexts. In general, several disconnections can be identified between the student's physical understanding of the problem and their mathematical model manipulation abilities.

The author observed in his design classes that students would often limit to the data at hand when asked to apply handbook-type engineering computations for a design project, or when required to carry out one as a practical exercise. They used the typical "plug-and-chug" sequence, while no insight was built into the relationships between variables. These author's and colleagues' observations are the rationale for introducing a mapping technique for deterministic algorithms, the Fuller-Polya diagram, as part of the class contents.

THE FULLER-POLYA DIAGRAM

Diagrams and graphs play an important role in problem solving, and they are essential tools for teaching engineering problem-solving. Graphs are an important part of formal communication in science and engineering. Diagrams naturally appear in problems involving physics and engineering as powerful solving tools for subproblems involving, for instance, geometry, vector quantities or time. Larkin and Simon [9] describe the psychological and computational advantages of using diagrams in problem solving. Among these advantages are that (i) diagrams minimize unwanted information, displaying only meaningful data and unknowns, (ii) they focus attention on elements relationships and they reduce search because related elements are usually grouped together, (iii) diagrams facilitate perceptual inferences and recognition of problem-solving methods that may be applicable, and (iv) they allow quick procedure checks. Good problems solvers sketch as early as possible graphs of possible relationships between involved variables and, if it applies, of their time evolution.

The Fuller-Polya diagram (FPD) for problem solving is a simple structured method that was outlined by Fuller [10] from a Polya's [11] suggestion and further formalized by Kardos [12]. It graphically organizes the variables and their relationships in the computation, including any non-algebraic procedure, with no need for explicitly state the math formulas (or the procedures details, if any of them were a non-strictly mathematical, i.e., a selection between discrete values, a value read from a table or graph, etc.).

The FPD methodology defines the following standard symbols:

Variable is a known or unknown "value" (it may be a number, an interval or even a code that identifies a standard part). The symbol for variable is a circle (or oval shape) enclosing the variable name as presented in Figure 1. Reversible algorithm is a computation or a sequence of computations that can be carried out in any "direction", even if algebra or mathematics manipulation may be needed to "reverse" such direction. The symbol for reversible algorithm is a square with an order number inside as presented in Figure 1.

Irreversible algorithm (or meta-operation) is a procedure that must be carried out only in a given "direction" (i.e., a double-entry table is in general an irreversible algorithm because the same output may result from different sets of inputs). The symbol for irreversible algorithm is a diamond with an order number inside as presented in Figure 1.





Flowlines connect each variable symbol to every algorithm symbol where the variable value is used in or produced from; arrowheads can be used when irreversible algorithms requires data to be input and output in one direction. Several examples of FPDs are presented in the work of Kardos [12]. Figure 2 presents student work in the author's class for multiple-leaf spring; the calculation procedure can be found in a standard handbook [13]. The figure shows at left-hand side a computation sketch and at right-hand side the FPD that was developed.



Figure 2. Student work: computation sketch and FPD for multiple (n) leaf spring (See [13]).

A consecutive numbering for the reversible and irreversible algorithm symbols relate to corresponding algorithm descriptions that are listed on a side. It is not needed to know the exact form of the algorithms to construct a FPD, but it is enough to know that such relationships should exist (i.e., from physical reasoning) or that they can

be established (i.e., by measurement and/or experiment). A list of variables names (and their descriptions) may be included.

The FPD is indicated for deterministic problems for which a solution could be produced from available data by using defined relationships, and provided that such data is sufficient and non-contradictory. The diagram is not useful, however, as a start-up point for open-end problems (as the typical creative-design and generation of alternatives problems) for which requirements may be poorly defined and/or be contradictory.

The FPD shows the structure of the solution independent of the computations and procedures while it focuses on the existence of relationships between variables rather than on any fixed "recipe" to produce an answer. It does not make any difference between "data" and "result", therefore making the diagram a general one of relationships between variables; some of them can be data or results for a particular case. The FPD can be very helpful for early identifying relations between "variables" (which are understood in a very general sense, for instance, they can be dimensions, parameters, factors, intervals, part numbers, models, materials, etc.) and "computations". FPD is general enough to show if a path to a solution can or cannot be produced from given data, even if the relationships (algorithms) are not yet fully available.

There is a suggested methodology to construct a FPD, which is based on the Polya's four-step method for problem solving [11]. In the context of FPD these four steps are: (i) the problem must be understood in its physical and geometrical meaning, (ii) the data (either input or output) must be identified, but not necessarily its values, (iii) the relationships between variables must be identified, but not necessarily be known in detail (it is enough to know that a relationship must exist) and (iv) symbols are drawn according to (ii) and connected by flowlines according to (iii). If the above does not produce a entirely correct map at the beginning, the diagram and its use should help clarifying the problem.

The FPD methodology is not an alternative to more heuristics problem-solving methods as the Polya's work, and further development by Wales et al. and their GENI (an acronym for Goal-Equation-Need-Information) method [14]. These more fundamental methods help planning a mathematical reasoning from the goal (e.g., the unknown) to the equations for solving and, unlike FPD, they may be used at the beginning of problem solving activity for poorly defined problems.

The FPD, however, proposes a formal language and graphics approach when the algorithms (i.e., equations or procedures) are known to exist for deterministic problems. Applications of the FPD methodology are surprisingly scarce in the available literature: Vidal and Becker [15] applied the Fuller-Polya map to successfully illustrate in a simple diagram the rather complicated Lagrange multiplier optimization of a coolant distribution system.

TEACHING OF THE FPD METHOD IN A SOPHOMORE MECHANICAL-DESIGN CLASS

Examples of student applications of the FPD

The class ENGR2431 Creative Decisions and Design has been taught each spring semester since 2004. The FPD methodology was briefly introduced to the authors' class in that spring semester with the purpose of exploring the applicability of the method at the sophomore level. Examples of student work from that spring 2004 class were presented by Molina [16]. In the following spring semesters of years 2005 and 2006 the author extended his teaching of the FPD. For each semester, the initial half of a 3-hour long lab was scheduled for teaching the FPD, selected examples from those sessions are presented in Figures 3 to 5.

Figure 4 shows a relatively complex problem that is above sophomore background and experience (the computation of helical compression or tension spring, see reference [17]); however, the FPD is complete and the application of the method allowed the student identifying all variables and algorithms, and all relationships are displayed. Figure 5 shows an example work in which the student did not realized some variables at first attempt (left hand-side diagram). But by refining the diagram with further inclusion of all data, the student found out connections between algorithms (right hand-side diagram).



Figure 3. Example of student work: FPD for the diameter computation of a hollow cylindrical shaft under critical speed constraint, see Appendix, and [16,17].



Figure 4. Example of student work: FPD for the computation of helical compression or tension spring, see [17].



Figure 5. Example of student work: FPDs for diameter computation of solid cylindrical shaft under the constraints of maximum torsion stress and critical speed, see Appendix, and [16, 17].

Exploratory study on feasibility and usefulness of introducing the FPD

The author continued his exploration on the teaching of FPD in the years 2005 and 2006. For each spring semester class, the initial half of one lab was employed for teaching FPD, and the second half for assessing students' opinions on the method helpfulness as compared to solving a problem not using it. The author first assigned each student a problem to solve before they were introduced to the FPD. The exercises are chosen from an engineering formula handbook [17] or an advanced machine-component-design textbook, as Shigley's [13]. Problems of varied difficulty are assigned, from some relatively simple ones to others which include concepts beyond sophomore background. Students are given half-hour to understand the computation and to develop a numerical solution for a given set of data. This first "problem-solving session" could be a rough "control" group for further inference.

The FPD method is then presented in a lecture format with the help of a handout and visual aids. It takes about 35 minutes of lecturing, including the discussion of examples. Students are then assigned to a second "problem-solving session" where each must apply the FPD method to one of the exercises of the previous first session. Twenty minutes are allowed to understand the computation and developing corresponding FPD. In the Spring 2005 semester different exercises were assigned to each student in the first and second sessions, while in the Spring 2006 semester the same exercise was assigned to each student in both sessions.

The students' opinions were surveyed by an anonymous questionnaire; ten statements on the method were rated in a scale from 1 to 5, which respectively corresponded to a qualitative scale from "strongly disagree" to "strongly agree". Space was provided for students to handwrite comments about advantages, disadvantages and suggestions about the teaching of the method. The survey statements were the following:

- 1. Without the knowledge of FPD, the assigned problem was difficult to solve.
- 2. By using the FPD, the assigned problem was easier to solve.
- 3. The FPD increased my understanding of the problem.
- 4. It would be difficult to solve some problems without the help of the FPD.
- 5. At times the FPD idea may be confusing.
- 6. I enjoyed using the FPD.
- 7. FPD helped me see the whole picture for the problem.
- 8. FPD is an appropriate topic for the class.
- 9. I look forward using the FPD in the future.
- 10. Using the FPD increased my understanding of standard computations in the practice.

A summary of the students' opinions is presented in Tables 1 and 2. For each of the ten survey statements (listed as order 1 to 10) the average, median, mode and standard deviation are listed in Tables 1 and 2. The sample sizes (i.e., class sizes) were respectively 16 and 19 students.

Table 1. Summary of students' opinions

for Spring 2005 class					
Order	Avg.	Median	Mode	Standard deviation	
1	2.25	2	2	0.77	
2	3.25	3.5	4	0.93	
3	3.00	3	2	1.10	
4	2.94	3	2	0.93	
5	3.81	4	4	0.83	
6	3.13	3	4	0.89	
7	3.25	4	4	0.93	
8	4.06	4	4	0.68	
9	3.25	3	4	0.77	
10	3.13	3	3	0.81	

for Spring 2006 class Standard Order Avg. Median Mode deviation 1 2.37 2 2 0.85 2 3.26 3 4 0.90 3 3.47 4 4 0.80 4 3.53 4 4 0.87 5 3.74 4 4 0.73 6 3.21 3 4 0.93 4 4 7 3.58 0.93 8 4.32 4 4 0.49 9 3.37 3 3 0.61 3.37 4 4 0.80 10

Table 2.	Summary	of students'	opinions

Tables 1 and 2 show differences between average, median and mode values for same statement. Given the small sample sizes, the following discussion refers to mode values (because mode is a more robust parameter in the presence of outliers).

Statements 1, 2, 3, 4, 7 and 10 surveyed students' opinion on FPD usefulness. The low rate for statement 1 ("Without the knowledge of FPD, the assigned problem was difficult to solve") clearly indicates that some of the problems were too simple for students to see any advantage in the use of the FPD. This result, however, is very helpful for an assessment of the set of exercises, and for preparing a new set of more challenging exercises. Rate for statement 2 ("By using the FPD, the assigned problem was easier to solve") indicates that students found that the FPD can help them on solving the problem. There were slightly higher mode values for Spring 2006, when students had to solve the same problem without and with help of the FPD.

Statements 3 and 4 ("The FPD increased my understanding of the problem" and "It would be difficult to solve some problems without the help of the FPD") showed important increases in Spring 2006 with respect to Spring 2005. As with statement 2, these results suggest that having students to solve the same problem without the FPD and later with it, could increase their appreciation of the method usefulness. The high rates for statements 7 and 10 ("FPD helped me see the whole picture for the problem" and "Using the FPD increased my understanding of standard computations in the practice") indicate that FPD may help deeper student understanding of computation meaning and it also can boost their confidence with unknown computations and concepts. There were higher mode values for Spring 2005.

Statements 5, 6 and 8 surveyed students' opinion on the learning experience with the FPD method. Ratings for statements 6 and 8 ("I enjoyed using the FPD" and "FPD is an appropriate topic for the class") are consistently high and they encourage the teaching of the topic at the sophomore level. However, ratings for statement 5 ("At times the FPD idea may be confusing") strongly suggest that a more thorough introduction to the method is needed. In this, the author is preparing new handouts, examples and teaching aids for the method. Students' work with the FPD reveal that they have difficulties to realize the differences between reversible and irreversible algorithms (or

that they not consider it that a relevant difference). These difficulties are likely related to little exposure of sophomores to non-strictly mathematical procedures. The author believes that there is a need to emphasize the understanding and use of the latter (i.e., by including tables, graphs, etc., in the exercises).

Statement 9 ("I look forward using the FPD in the future") surveyed students' willingness of using FPD in the future. For this statement students' opinions are between "unsure" and "agree". As with the case of statement 5, the author believes that an improved teaching approach and having student more extensively use the method can help their appreciation of FPD advantages. These results are very helpful to guide further implementation of the FPD technique in classes.

The introduction of the FPD methodology in the author's design class proved promising for guiding student analysis and understanding of the problem at hand. The methodology prompts students to first identify variables and relationships and to "see the whole picture" before attempting any solution. Feedback from students' comments has been generally positive: they pointed out (a) the fact that they can see a path to the problem solution as resulting from the data without carrying out any computation and (b) that the method boosts their confidence that they can do standard-design computations, even if they had no full or previous knowledge of the topic, or if the involved mathematics were not clear to them from the start.

CONCLUSIONS AND REMARKS FOR FUTURE RESEARCH

The author introduced in his sophomore design class the Fuller-Polya Diagram, a structured method for mapping deterministic algorithms. This mapping allows insight of engineering standard computations beyond a numerical calculation for a given set of data. FPD helps the identification of data and procedures even in the absence or ill definition for part of them. The rationale for teaching the method is that it should help student to avoid the typical "plug and chug" approach to deterministic problem solving.

An exploratory study of students' performance and opinions was carried out by (i) assigning problem solving without and with the use of the FPD and (ii) a questionnaire after teaching of the method. Analysis of students' opinions indicates that FPD can help them "seeing the whole picture" for the problem and it can increase their understanding of standard engineering computations. Students' opinions also encourage the teaching of the topic at the sophomore level, but they suggest that a more thorough introduction to the method is needed. Having student more extensively use the method may help their appreciation of its advantages. These results are very helpful to guide the author's further implementation of the FPD technique in future classes. He is preparing new handouts, examples and teaching aids for the method.

The author also is considering a more formal assessment of the technique in his present design class. However, comparing the absolute performances (i.e., the "grades") of students when both understanding and solving multiple-step problems with and without the help of the FPD methodology can be complex, the author is not aware of appropriate metrics for such assessment. Also it is difficult for a small class to define a fair educationassessment experiment with appropriate "control data" (e.g., students without previous knowledge of FPD) and "experimental data" (i.e., students with knowledge of FPD). The student success and difficulties when dealing with the exploratory exercises of this paper can be helpful to design such assessment. As per suggestion of one anonymous reviewer of this work, the author plans to explore the use of different statistical formulations (e.g., Ftest, P-test) for testing statistical significance of such future assessment.

The author acknowledges that the usefulness is yet to be further proved of introducing the FPD as a general engineering problem-solving tool. He plans to also investigate other uses of the FPD methodology in engineering problem solving. The FPD provides a graphics way of checking data sufficiency and/or data redundancy, but this property has not yet been studied.

REFERENCES

- [1] Raviv, D., "Do We Teach Them How to Think?", *Proc. of the 2002 ASEE Annual Conf. and Exposition*, Montreal, Quebec, Canada, June 2002, Session 2630.
- [2] Little, P., Cardenas, M., "Use of "Studio" Methods in the Introductory Engineering Design Curriculum", *ASEE Journal Of Engineering Education*, Vol.90, No3, 2001, 309-318.

- [3] Ullman, D., *The Mechanical Design Process*, McGraw-Hill, New York, NY, 2003.
- [4] Clark, G., Mitra, A., Molina, G., "Effects of Variable Mix on Student Mathematics Performance", *Proc. of the 2004 ASEE Annual Conf. and Exposition*, Salt Lake City, UT, June 2004, Session 3565.
- [5] Redish E. F., "Problem Solving and the Use of Math in Physics Courses", Proc. of the World View on Physics Education in 2005: Focusing on Change Conference, Delhi, India, August 21-26, 2005, Invited talk.
- [6] Redish E. F., Steinberg, R.N., Saul, J. M., "Student difficulties with math in physics: Giving meaning to symbols", http://www.physics.umd.edu/perg/papers/redish/talks/aapt96m.htm, Proc. of the AAPT Meeting, College Park, MD, August 9, 1996.
- [7] Tuminaro, J., Redish, E.F., "Student Use of Mathematics in the Context of Physics Problem Solving: A cognitive model", *U. of Maryland preprint*, 2005, http://www.physics.umd.edu/perg/papers/redish/T&R.pdf
- [8] Klebanoff, A.D. and Winkel, B.J., Technology-based problems in calculus from science and engineering, *Proc. of the 2006 ASEE Annual Conf. and Exposition*, Chicago, IL, June 1996, Session 1265.
- [9] Larkin, J., Simon, H. A., "Why a Diagram is (Sometimes) Worth 10,000 Words", *Cognitive Science*, Vol.11, 1987, 65-99.
- [10] Fuller, M., Collected Papers of the ERM Division ASEE, 1975, 114-128.
- [11] Polya, G., How to Solve It?, Princeton University Press, Princeton, New Jersey, 1945.
- [12] Kardos, G., "Problem Solving with the Fuller-Polya Diagram", *International Journal of Applied Engineering Education*, Vol.1, No.2, 1985, 105-111.
- [13] Shigley, J.E., Mischke, C.R., Budynas, R.G., *Mechanical Engineers Design*, McGraw-Hill, New York, NY, 2003.
- [14] Wales, C. E., Stager, R. A., *Thinking with equations: Problem solving in math and science*, Franklin KY: The Writers Block, Center for Guided Design, West Virginia University, Morgantown, 1990.
- [15] Vidal, R.V.V., Becker, P.W., "On Lagrange Multipliers in Work with Quality and Reliability Assurance", *IEEE J. on Selected Areas in Communication*, Vol.SAC-4, No.7, 1986, 1143-1148.
- [16] Molina, G. J., "Use of Fuller-Polya diagram for teaching engineering problem solving in undergraduate design classes", *Proc. of the 2005 ASEE Annual Conf. and Exposition*, Portland, OR, June 2005, Session 2625.
- [17] Hicks, T.G., Mechanical Engineers Formulas Pocket Guide, McGraw-Hill, New York, NY, 2003.

GUSTAVO J. MOLINA

Dr. Molina obtained a Diploma in Mechanical and Electrical Engineering from National University of Cordoba, Argentina in 1986. Until 1992 he acquired expertise in standard testing, PVD coatings, vacuum techniques and the design of experimental equipment. In 1994 he received a Master's degree in Mechanical Engineering from the University of Ottawa, Canada, where he conducted research on the nondestructive characterization of low-energy impact properties of polymers and composites. In 2000 Dr. Molina obtained a PhD in Mechanical Engineering from Virginia Tech for his work on the characterization of electron triboemission from ceramic surfaces. Dr. Molina is currently an Associate Professor of Engineering Studies in the Department of Mechanical and Electrical Engineering Technology, Georgia Southern University. His current teaching interests include creative design, problem-solving techniques and the development of studio methods in engineering. He also teaches mechanics, graphics communication, CAD and computer applications in engineering.

Appendix

Computations presented in Figure 3 and 5: Pump shaft design [See reference [17], pages 272-273]

(i) Torsional stress S_S is computed:

For solid shafting: $S_s = (16 \text{ T}) / D_0^2$

For tubular shafting: $S_{S} = (16 \text{ T}) / (D_{0}^{2} (1 - (D^{4} - D_{0}^{4})))$ Where:

- $S_{S}\!:$ torsional stress
- T : transmitted torque
- D : shaft inner diameter
- D_0 : shaft outer diameter

(ii) Critical speed $N_{\mbox{crit}}$ is computed from:

(ii.a) Shaft deflection y under own weight:

 $y = (5 \text{ w } \text{L}^4) / (384 \text{ E I})$

Where:

- w : shaft own weight per unit length
- L : shaft length between supports
- $E: Young \ modulus$
- I : moment of inertia for solid shafting: I = $3.1416 D_0^4 / 64$ and for tubular shafting: I = $3.1416 (D_0^4 - D^4) / 64$

(ii.b) Then critical speed N_{crit} in rpm:

 $N_{crit} = 187 \left(\ 1 \ / \ y \ \right)^{1/2} \ ; \qquad for \ y \ in inches.$