

AN ABSTRACT OF THE THESIS OF

Lawrence D. Fields for the degree of Master of Science in Chemistry
presented on November 7, 1985.

Title: Some Recent Developments in the Theory of Significant Figures
and their Applications to Chemical Instrumentation

Abstract approved: *Redacted for Privacy*

Significant figures have been a nebulous area in the physical sciences. The results of a literature survey, and of a questionnaire distributed among the Chemistry Department faculty at Oregon State University, revealed widely differing interpretations regarding the proper role of significant figures. The present study has a two-fold nature.

First the existing theory was extended to include the Root Rule, which unambiguously assigns the correct number of significant figures to the result of a chemical or physical calculation.

Then significant figure theory was applied in developing improved methods for recording (and transmitting) tables of measurement data. The upshot is two novel data compression techniques—Decimal Coded Binaries (DCB) and Eight Code—which efficiently combine a measurement and its uncertainty into a single number.

An investigation of the computer interfacing requirements of DCB and of Eight Code vis-a-vis a pH meter and a spectrophotometer demonstrated that these new communications tools are easily implemented.

© Copyright by Lawrence D. Fields
November 7, 1985

All Rights Reserved

Some Recent Developments in the Theory
of Significant Figures and their
Applications to Chemical
Instrumentation

by

Lawrence D. Fields

A THESIS

submitted to

Oregon State University

in partial fulfillment
of the requirements
for the degree of

Master of Science

Completed November 7, 1985

Commencement June 1986

APPROVED:

Redacted for Privacy

Professor of Chemistry in charge of major

Redacted for Privacy

Head of Department of Chemistry

Redacted for Privacy

Dean of Graduate School

Date thesis is presented _____ Nov. 7, 1985

Typed by Lawrence D. Fields for _____ Lawrence D. Fields

ACKNOWLEDGEMENT

I would like to acknowledge the many people whose direct and indirect contributions have made this research possible. First and foremost, I am grateful to my parents, who have always encouraged my curiosity about the natural world.

I was also very fortunate in finding a major professor who allowed me the freedom to pursue my own research interest in the capital-intensive field of analytical chemistry. Without Stephen Hawkes' insight, wise counsel, and generosity of spirit, the project would never have come to a mature conclusion.

Michael Schuyler's course on microcomputer interfacing was a major source of inspiration for the idea of Decimal Coded Binaries. Mike also loaned me his AIM box, and provided his DVM program, the skeleton of which handled the interfacing requirements of the Eight Code program.

I am indebted to Edward Plepmeler for his encouragement and valuable advice. Moreover I appreciate the helpful suggestions made by John Arthur, Kenneth Hedberg, David Shoemaker, Lawrence Thomas, and John Westall.

I would also like to thank the Chemistry Department faculty members who participated in the significant figure survey.

TABLE OF CONTENTS

INTRODUCTION	1
SIGNIFICANT FIGURE FUZZINESS IS MINIMIZED BY A ROOT TEN RULE	4
PART I. MULTIPLICATION AND DIVISION: Introduction	5
What is Uncertainty?	7
The Meaning of Significant Figures	9
The Simple Rule	12
The Root Rule	13
The 1-5 2-8 Rules	19
The Three Rule	22
Significant Figures and Science Education	23
Part I Summary and Conclusions	24
PART II. SIGNIFICANT FIGURES AND SIMPLE UNCERTAINTY	
PROPAGATION THEORY:	26
The Extended Root Rule	26
More Boundary Conditions	31
Part II Summary and Conclusion	36
APPENDIX	37
DECIMAL CODED BINARIES: A TECHNIQUE FOR COMPRESSING A	
MEASUREMENT AND ITS UNCERTAINTY INTO A SINGLE QUANTITY	43
Abstract	44
Introduction	44
Linear Code	50
Decimal Coded Binaries	55
Experimental Section	63
Results and Discussion	65
Summary and Conclusion	65
Appendix	69
DCB PROGRAMS FOR THE HP15C SCIENTIFIC CALCULATOR	73
EIGHT CODE	79
Theory	79
Experimental and Results Section	83
BIBLIOGRAPHY	84
APPENDIX	85
Questionnaire	85
Eight Code Programs for the HP15C Scientific Calculator	86
Computer Tapes	93

LIST OF TABLES

<u>Table</u>		<u>Page</u>
I	A Comparison of the Significant Figure Rules	25
II	Relationship Between Uncertainty Code and Estimated Measurement Uncertainty with the ± 0.5 to ± 5 Rounding Convention	51
III	Relationship Between Estimated Measurement Uncertainty, Rounding Ratio, and Inflation	52
IV	Relationship Between Uncertainty Code and Estimated Total Uncertainty with the ± 0.5 to ± 5 Rounding Convention.	53
V	Relationship Between Uncertainty Code and Estimated Measurement Uncertainty for the ± 2 to ± 20 Rounding Convention	54
VI	Connections Between the AIM 65 Microcomputer and the Orion 701A Ionalyzer	64
VII	A Comparison of DCB with Linear Code	67

SOME RECENT DEVELOPMENTS IN THE THEORY OF SIGNIFICANT FIGURES AND THEIR APPLICATIONS TO CHEMICAL INSTRUMENTATION

INTRODUCTION

Significant figures have long been a delinquent stepchild of the physical sciences. Not only is there a paucity of literature on the subject, but the results of an informal survey of the Chemistry Department faculty at Oregon State University indicates a polarization of views on the proper use of significant figures. Thus significant figures have been elevated to the status of a sacred cow. As with most sacred cows, there is a consensus on the sacredness, but the nature of the cow is not clear. We need a more cogent theory of significant figures.

If the components of a calculation are represented by **stand-alone significant figures** (measurement-based quantities, for which the uncertainties are not given directly, and must be inferred from the number of significant figures used to represent them), then we should be able to represent the result of the calculation in the same way. We will see that the **Root Rule** (described herein), unlike previous significant figure rules, successfully integrates common arithmetic operations with propagation-of-uncertainty analysis, thereby providing a sensible and unambiguous criterion for deciding how many digits should be used to represent the result of any calculation in the physical sciences.

However the utility of the theory is limited by the number system being used. Because of our ten fingers, the estimated uncertainty of a quantity represented by stand-alone significant figures can be resolved only within a ten-fold range. Would some other number system be better?

If we ignore the fact that binary numbers take up so much space, we might do well to adapt the theory of significant figures to the base two number system. Then the uncertainty of a quantity represented by stand-alone significant figures could be pinned down within a two-fold range. Fortunately it is possible to construct a nomenclature which

incorporates this virtue of binary into the base ten number system, a completely new application of significant figure theory.

In this new approach, two number systems are combined in order to incorporate two types of information—a measurement and its uncertainty—into a single number. The result is a format, which looks and reads like ordinary base 10, and which gives the same information about uncertainty as binary. This type of nomenclature is called **Decimal Coded Binaries (DCB)**.

One possible application of DCB would be in the reporting of tables of measurement data. If a table has a DCB format, then reporting the uncertainty of a measurement along with the measurement itself will be economical, and will not decrease the readability of the table.

However DCB may have a larger impact in the area of data compression and transmission. Using DCB, a collection of measurement data can be stored more compactly in memory devices, and can be telecommunicated with greater efficiency, than would be the case if each measurement and its uncertainty were treated as two separate numbers.

There is a second data compression technique, called Linear Code, which is very simple. Linear Code serves primarily as a auxiliary technique, which helps to clarify DCB and its advantages. Moreover Linear Code can be hybridized with DCB. Therefore Linear Code is included as a section in the chapter on DCB.

The bulk of this thesis (which uses the optional publication format) consists of two papers on stand-alone significant figures and on DCB (with Linear Code). The first has been accepted for publication in the *Journal of College Science Teaching*. The second has been submitted for publication to *Analytical Chemistry*. The thesis also contains some material (like Eight Code), which has not been submitted for publication.

Just as DCB combines two number systems (2 and 10) in order to compress a measurement and its uncertainty into a single number, a third technique—Eight Code—accomplishes the same objective, using three number systems (2, 4, and 10). At one point in our investigation, Eight Code was a competitor of DCB. However mature consideration showed that Eight Code is a prototype for a more sophisticated approach (combining N number systems), which is beyond the scope of the present research. Here we will consider only data compression techniques which can be programmed into an HP15C scientific

calculator. Listings of these programs are also included. All three techniques are original.

DCB has commercial implications. If DCB becomes generally accepted, then scientific calculators will have to be redesigned to accomodate DCB coding and decoding routines, as well as DCB arithmetic, which will draw upon the Root Rule concept. Moreover computer languages used in scientific work may need a new type of declaration for DCB variables, in addition to the usual integer and floating point variables.

**SIGNIFICANT FIGURE FUZZINESS IS
MINIMIZED WITH A ROOT TEN RULE.**

By Lawrence Fields and Stephen Hawkes

Department of Chemistry
Oregon State University
Corvallis, Oregon 97331

accepted for publication by the Journal of College Science Teaching

PART I: MULTIPLICATION AND DIVISION

Introduction.

"Why it's perfectly obvious!" "It's just a matter of common sense." These are typical responses to the question of how many significant figures should be used to express the result of a calculation in the physical sciences. In an attempt to measure the ubiquity of such common sense, we conducted an informal survey of the Chemistry Department faculty at Oregon State University. Each participant was given a questionnaire, which asked him or her to write the following product with the proper number of significant figures:

$$1.43 * 4.098 = 5.86014$$

Only one fifth of the respondents answered correctly.

In all fairness, we must add that one third of the respondents had reasonable complaints about ambiguity in the question. Even so, our questionnaire indicated that "significant figure" is similar to "democracy", in the sense that both concepts mean different things to different people.

This type of question, and the confusion it engenders, is the raison d'être for this paper. Some scientists would like to pontificate **stand-alone significant figures** out of existence. Yet significant figures are mentioned in the first chapter of many general chemistry and general physics texts, and are often a sub-problem in undergraduate lab reports (and in the real world). Since stand-alone significant figures are a fact of life, it would behoove us to treat them systematically.

It has been said that a teacher should be like an iceberg, in the sense that 90% of the essence is beneath the surface. In accordance with this philosophy, we are attempting a two-fold education project. First we seek to give the teacher a solid understanding of the theory of significant figures, with special emphasis on calculations in which the component uncertainties are not given directly. This will be the main thrust of Part I, whose cornerstone is the **Root Rule** for

multiplication and division calculations, having a single identifiable **least precise factor**.

Moreover we have distilled some teachable significant figure algorithms (both of which are based upon the Root Rule), which should be appropriate for college-level courses in chemistry and physics. For teachers, the most pertinent results are the **Three Rule** and the **1-5 Rule** in Part I, which also contains a section on "Significant Figures and Science Education".

We are not attempting to usurp the role of rigorous uncertainty analysis; rather our topic is a specialized area within this broader field. The basic idea of this paper is that the number of significant figures used to express the product or quotient of a physical science calculation should reflect the fact that the percent uncertainty of the result will be approximately equal to percent uncertainty of the least precise factor.

Part I is addressed largely to college science educators seeking useful and teachable knowledge to impart to their students. Part II is more comprehensive, and uses more complicated procedures (all of which can be written into computer programs). It is of interest to the scientific generalist within each of us (and to the manufacturers of computers, scientific instruments, and calculators).

A survey of the literature on significant figures suggests that Newton's giants are relatively few in number, and that they have rather narrow shoulders. This gives us the opportunity to explore more than the unusual amount of virgin territory in this paper.

What is Uncertainty?

Uncertainty is a key concept in the theory of significant figures. There are two main reasons why we might feel uncertain about a reported measurement which we encounter in the literature.

- 1) **Systematic error** (sometimes called **determinate error**). Skoog and West¹ have delineated the following three categories of systematic error:
 - (a) personal error (such as a number bias on the part of the experimenter)
 - (b) instrumental error
 - (c) method error
- 2) **Random error** (also known as **indeterminate error** or **imprecision**). If the random error is large then the measurement is not very reproducible.

Now that we have mentioned some of the sources of **measurement uncertainty**, we need to define the term. For the sake of simplicity, we will consider only two of the more popular definitions of uncertainty.

- A. Uncertainty is the absolute limit of error. This definition is easy to understand, and is the raw material for the worst case method² of uncertainty analysis. Definition A has the advantage of taking both random and systematic errors into account.

However this definition has very limited utility in the real world. The absolute limits of error are often unknown and/or unknowable. Moreover if the random errors are normally distributed, as we so often assume, then the absolute limits of error don't even exist!

- B. Uncertainty is the sample standard error of the mean (s/\sqrt{N}) for a series of measurements, or the population standard deviation (σ) for a single measurement. Unlike A above, this definition applies to the real world, since the standard error of the mean can be calculated quite readily. The standard error definition is especially useful

when the systematic error is thought to be negligible in comparison with the random error. However the standard error of the mean says nothing about systematic error.

In the remainder of Part I, we will use Definition B for two reasons. First, the standard error of the mean is the more sensible definition. Secondly, we will introduce some vest-pocket algorithms (like the 1-5 Rule) which are not always appropriate for the absolute limit of error definition of uncertainty.

In Part II, we will extend our approach to include significant figure determinations for functions of n variables. In the more general case, the equation for propagation of uncertainty will depend upon the definition of uncertainty. We will favor the standard error of the mean definition.

We have already mentioned that the percent uncertainty implied by the number of significant figures used to represent the product or quotient of a physical science calculation should be in the same ballpark as that of the least precise factor. In the next two sections we will develop some concepts, which will enable us to see why the traditional approach to significant figures does not fully come to grips with this important idea.

The Meaning of Significant Figures

There is a consensus among physical scientists that the reporting of a measurement should include at least the following two components:

- 1) the measurement itself
- 2) an estimate of the measurement's uncertainty.

Sometimes the estimated uncertainty is given indirectly in terms of the number of significant figures in a measurement-based quantity. In the following discussion, " \pm " will denote the uncertainty.

There seem to be four main schools of thought about the meaning of significant figures. The Conservatives insist that the last significant figure must be known with certainty. For example, a measured value of 1.43 is understood to mean 1.43 ± 0.005 at the worst. In essence, the Conservative is saying, "I want information I can trust."

The Mainstreamers believe that there can be appreciable uncertainty only in the last significant digit, but there is disagreement about how much uncertainty is allowed in the last digit (± 1 , ± 2 , ± 3 , etc.).

The Liberals feel that there can be appreciable uncertainty in the last two significant digits. The liberal is saying, "I want the maximum amount of useful information."

The central concepts of this paper will be illustrated using the Mainstream approach. However little generality would be lost if the Liberal approach were used instead. We should mention one final school of thought about significant figures.

The Independents are of the opinion that significant figures should never be used by themselves to indicate an estimated uncertainty. However there is a major problem with the Independent position. Organic chemists do not always report \pm values for percent yields. Physicists and analytical chemists do sometimes omit uncertainties from data tables. In the real world, stand-alone significant figures are often used to indicate (in a round-about way) the uncertainty of measurement-based quantities. However this usage has not been very systematic.

In some instances it is reasonable to allow stand-alone significant figures to supplant directly reported uncertainties. In data tables reported uncertainties take up extra space, and can decrease

readability. Moreover, if a chemical analysis is performed on only two or three samples drawn from a large population, then the direct reporting of the uncertainty may be slightly presumptuous.

When the uncertainties of the components in a calculation are available, it is a relatively simple matter to assign what one considers to be the correct number of significant figures. Significant figures are fuzziest when they stand alone, when \pm values are not available. In this paper, we are mainly concerned with these fuzzy significant figures. Another important source of significant figure fuzziness is politics.

Influential laymen may insist on knowing the results of scientific measurements in very simple terms. This can result in politically motivated biases and double standards with respect to significant figures. The number of significant figures assigned to a quantity may depend on its intended use. If a result will be used by decision-makers who will not understand its uncertainty, or while understanding, will ignore it for political reasons, while blaming the writer for any unfortunate consequence, then it is better to round, such that the last reported digit can be rigorously defended. If the decision-maker needs the most detailed information and understands its limitations, then uncertainty in the last two digits may be justified.

Sometimes a writer's intention is simply to give the reader a general feeling for the magnitude of a quantity. Often two or three significant figures are sufficient for this purpose, even though four might be scientifically justifiable. In the remainder of this paper, we will try to ignore such political considerations.

There is an obvious lack of consensus on the meaning of stand-alone significant figures. However we will formulate valid significant figure rules which are relatively independent of \pm conventions.

We will tentatively assume, without loss of generality, that the final significant digit in an inexact number (i.e. a measured value) can readily be one higher or one lower than face value. For example, "7.2" is understood to mean 7.2 ± 0.1 . The "0.1" is the **implied uncertainty**. The **implied relative uncertainty** is $0.1/7.2 = 1/72$.

It would be nice if the number of significant figures in a product (or quotient) could be adjusted such that the implied relative uncertainties of the product and of the least precise factor are equal. However this is usually impossible to do. In the general case, one of two things is done:

- 1) Make a seat-of-the-pants judgment.
- 2) Apply a significant figure algorithm.

Of course no significant figure algorithm can exist in a vacuum. It must be consistent with propagation-of-uncertainty analysis. In our earlier example

$$1.43 * 4.098 = 5.86014$$

"1.43" has a larger implied relative uncertainty than "4.098". Thus "1.43" is the **least precise factor**. In general the least precise factor is the factor with the least number of significant figures.

Exact numbers, which come from definitions and from ordinary counting, have infinitely many significant figures. For example in the kinetic energy formula,

$$E = 0.5 mv^2$$

the "0.5" can never be the least precise factor.

If two factors have the same number of significant figures, then the least precise factor is the one whose **mantissa*** is the smaller of the two. However in this case, we should bear in mind that both factors will contribute an appreciable amount of uncertainty.

As a first approximation, we can usually ignore the uncertainty contribution made by the other factor. In this case, it is intuitively obvious that the relative uncertainty of the product should approximate the relative uncertainty of the least precise factor. This fact is the basis for the significant figure algorithms of Part I.

*For example in 3.45×10^{-2} the "3.45" is the mantissa.

The Simple Rule

The **Textbook Rule** for significant figures in multiplication and division, usually given in physics and chemistry texts is similar to the following:

The number of significant figures used to express a product or quotient should not exceed that of the least precise factor.

However this begs the question. The rule which is actually taught is usually similar to the following **Simple Rule**.

The number of significant figures in a product or quotient is equal to that of the least precise factor.

(Note that the Conservative approach is the tacit frame of reference for the first Textbook Rule, while the second Simple Rule is consistent with both the mainstream and Liberal approaches.)

The Simple Rule carries with it four implicit assumptions:

- 1) A single least precise factor exists.
- 2) The \pm value is not known, but the same \pm convention applies in determining the uncertainty of each factor. (For instance the 1.43 and 4.098 in our example can have uncertainties of 0.01 and 0.001 respectively, or 0.02 and 0.002, but not 0.01 and 0.002.) Thus the least precise factor can be readily identified.
- 3) The least precise factor is the only significant source of uncertainty in the product.
- 4) The \pm convention is not smaller than about ± 1 . (See Part II)

We will see that even when these criteria are met, the Simple Rule frequently gives absurd results.

Pinkerton and Gleit³ have taken a more sophisticated approach. They argue that the number of significant figures in a quantity is logarithmically related to the reciprocal of its relative uncertainty.

Although this helps to clarify the concept of significant figures, it has little to say about the practical problem at hand.

In the next section we will carry this concept to its logical conclusion, by introducing the **Root Rule** for significant figures in multiplication and division, which is the basis for the methods used in the General Chemistry program at Oregon State University.

The Root Rule

A significant figure algorithm should be unbiased. If the algorithm has an optimistic bias, then the accuracy of one's measurements is exaggerated. A pessimistic bias means that information is being thrown away unnecessarily. The Simple Rule, as applied to the Mainstream and Liberal approaches, is essentially unbiased when it is used correctly. However it has boundary condition problems, which we will illustrate using the following **1-9 Paradox**. Consider the division calculation:

$$1.01/1.020 = 0.990196$$

According to the Simple Rule, the quotient should be written as 0.990 (three significant figures). The implied relative uncertainty of this number is calculated as follows:

$$0.001/0.990 = 1/990 = 0.1\%$$

The implied relative uncertainty of the least precise factor is:

$$0.01/1.01 = 1/101 = 1\%$$

The two implied relative uncertainties differ by nearly an order of magnitude (a factor of ten). A more sensible answer would be 0.99, since the implied relative uncertainties of both quotient and least precise factor would be about 1%. Now we will develop the conceptual framework for an algorithm which does not lead to such absurdities.

Let P be the implied relative uncertainty of the product (or quotient), which has been tentatively rounded according to the Simple Rule, and let L be that of least precise factor. By adjusting the number of significant figures used to express the product (or quotient), we can manipulate P , such that P is as close (in a logarithmic sense) to L as possible. But how close is possible in the general case? According to the Simple Rule, it is always possible to adjust P , such that

$$0.1 < P/L < 10$$

If the product (or quotient) is rounded to one less digit than this, then the implied relative uncertainties will be L and $10 P$. Usually either P/L or $10 P/L$ will be nearer to one. In the borderline case,

$$\frac{P}{L} = \frac{L}{10 P}$$

and solving for P/L yields

$$P/L = 1/\sqrt{10}$$

When the product (or quotient) is given one more significant figure than the Simple Rule would suggest, the two implied relative uncertainties will be L and $0.1 P$. Now there is the possibility of a second type of borderline case in which

$$\frac{P}{L} = \frac{L}{0.1 P} = \sqrt{10}$$

This then is the framework mentioned above for the result, which is discussed below as the **LPF Root Rule** (LPF = least precise factor). Our two borderline cases lead us to the following rule for significant figures in multiplication and division.

The LPF Root Rule:

If the uncertainties are not given, and if

$$\frac{1}{\sqrt{10}} < \frac{P}{L} < \sqrt{10}$$

then the product (or quotient) has been rounded to the correct number of significant figures. (The same four restrictions, which apply to the Simple Rule, also apply to the LPF Root Rule.)

The Simple Rule allows an implied relative uncertainty discrepancy of nearly an order of magnitude in either direction. The LPF Root Rule says that this is too much, that the discrepancy should be no more than half an order of magnitude.

We will return to our original example in order to illustrate the LPF Root Rule. We will also show that the LPF Root Rule is quite general, and is relatively independent of \pm convention (provided that it is not too small).

$$1.43 * 4.098 = 5.86014$$

First we will use the ± 1 convention.

The implied relative uncertainty of 1.43 is $0.01/1.43 = 1/143$.

The implied relative uncertainty of 5.9 is $0.01/5.9 = 1/59$.

$$1/59 \div 1/143 = 143/59 = 2.42 < \sqrt{10}$$

Therefore the product has been correctly rounded to 5.9.

Note that the implied relative uncertainties of least precise factor and product differ by $\log 2.42 = 0.38$ of an order of magnitude. It is the best we can do in this example, as we will see in the next paragraph.

Suppose that the product had been rounded to 5.86 (three significant figures) instead. The implied relative uncertainty of 5.86 is $0.01/5.86 = 1/586$. The new implied relative uncertainty ratio would be

$$1/143 \div 1/586 = 586/143 = 4.1$$

and the implied relative uncertainties of least precise factor and

product would differ by $\log 4.1 = 0.61$ of an order of magnitude (as compared with 0.38 of an order of magnitude for two significant figures). Thus "5.9" is the lesser of two evils.

This result is not affected if some other \pm convention is used instead of ± 1 . To demonstrate this, we will apply a ± 2 convention.

The implied relative uncertainty of 1.43 is $0.02/1.43 = 2/143$.

The implied relative uncertainty of 5.9 is $0.02/5.9 = 2/59$.

$$2/59 \div 2/143 = 143/59 = 2.42 < \sqrt{10}$$

Therefore our approach really is "without loss of generality". This makes the LPF Root Rule relatively independent of \pm convention, provided that all of the inexact numbers used in the original calculation follow the same \pm convention. There is no a priori way to know that this will be true, but it is the only assumption we can make if the uncertainties are not given directly. We leave it as an exercise for the reader to show that the LPF Root Rule leads to the correct number of significant figures in the 1-9 Paradox ($1.01/1.020 = 0.990196$).

The LPF Root Rule says that in most cases we can (and should) adjust the number of significant figures in the product or quotient such that uncertainty in the final digit of the product or quotient is 0.3162 to 3.162 times the uncertainty of the final digit of the least precise factor—even though we do not know what this latter uncertainty is!

In terms of the Pinkerton-Gleit definition of significant figures, the LPF Root Rule says that a product or quotient has been rounded correctly if it is within ± 0.5 significant figures of the least precise factor, i.e. if

$$\left| \log \frac{1}{\text{IRU (LPF)}} - \log \frac{1}{\text{IRU (result)}} \right| < 0.5$$

where **IRU** represents "the implied relative uncertainty of", and **LPF** represents "the least precise factor".

Sometimes either rule may seem either too optimistic or too pessimistic to an outside omniscient observer. However the Root Rule is a reasonable and unambiguous method for significant figure determinations, when the component uncertainties are not known.

Another limitation of the LPF Root Rule will be illustrated with the following example:

$$2 \times 4.4 = 8.8 \text{ which rounds to } 9$$

The least precise factor "2" has only one significant figure. According to the LPF Root Rule, the number of significant figures in the product should be one less than this, or zero, which is absurd. The best we can do is to round the product to 9.

One-significant-figure-quantities can be quagmires of uncertainty, especially in subtraction calculations, involving two nearly equal quantities, each having appreciable uncertainty. In such cases \pm conventions have little meaning.

The LPF Root Rule for significant figures is a right-brain approach, which may not be suited to left-brain people. Therefore we will give some equivalent formulations and approximations to the Root Rule. The casual reader should choose the most comfortable of these new rules, rather than trying to master and to remember all of them. (We recommend either the 1-5 Rule or the Three rule, given in the next two sections.) For the benefit of computer programmers, the LPF Root Rule may be expressed by the following LPF Root Rule Algorithm:

- 1) Express all numbers in scientific notation. For example, 0.0345 is 3.45×10^{-2} in scientific notation. The 3.45 shall be called the **mantissa**.
- 2) Find the least precise factor.
 - A. The least precise factor is the factor with the least number of significant figures.
 - B. If two factors have the same number of significant figures, then the least precise factor is the one whose mantissa is the smaller of the two. (Strictly speaking, neither the Simple Rule nor the LPF Root Rule applies in Case B, since the least precise factor does not contribute essentially all of the uncertainty. If we want to avoid introducing an optimistic bias into our results, then we must use the methods of Part II.)
 - C. Obviously an exact number cannot be the least precise factor because an exact number has infinite precision.
3. Compare the mantissas of the answer and of the least precise factor.

- A. Divide the larger mantissa by the smaller mantissa.
- B. If this quotient is less than $\sqrt{10}$, then the answer will have the same number of significant figures as the least precise factor.
- C. If the quotient is $\sqrt{10}$ or more, then adjust the number of significant figures in the answer mantissa, such that the smaller mantissa has one more significant figure than the larger mantissa. (Do not give the answer mantissa less than one significant figure.)

It can be shown that the LPF Root Rule over-rides the Simple Rule about 25% of the time in two-factor multiplication and division calculations. This means that when the component uncertainties of a multiplication or division calculation are not known, the Simple Rule can be expected to give unreasonable results (as in the 1-9 Paradox) about one fourth of the time. (See Appendix.)

The LPF Root Rule and the above algorithm are too complicated for most general chemistry students. In the next section, we will derive a simplified procedure.

The 1-5 2-8 Rules

The purpose of this section is to develop the 1-5 2-8 Rules, two 'quick and dirty' rules of thumb for the LPF Root Rule. In particular, the 1-5 Rule is easy to apply, easy to remember, and very useful. But first we must introduce an intermediate result, which follows from the LPF Root Rule. The starting point for the 1-5 2-8 Rules is the following 1-7 Rule.

The Simple Rule does not apply when the least precise factor begins with a "1", and the product or quotient begins with "7", "8", or "9" (or vice versa). In this case, the quantity beginning with "1" should have one more significant figure than the other quantity. As a mnemonic, we can say, "One gets one more." (Remember that a quantity cannot have less than one significant figure.)

The 1-7 Rule is justified as follows:

- 1) For the borderline case in which the 1-7 Rule barely over-rides the Simple Rule, the mantissas of the two numbers will be 1.999 and 7.00.
- 2) $7.00/1.9999 = 3.5 > \sqrt{10}$
- 3) Therefore by the Root Rule, the correct number of significant figures has been assigned.

We have seen that the 1-7 Rule is always correct when it supersedes the Simple Rule. However it doesn't supersede often enough. The 1-7 Rule lacks comprehensiveness. In the next few paragraphs, we will argue that the following procedure would be better than the 1-7 Rule.

The 1-5 2-8 Rules:

- A. If the least precise factor begins with a "1" and the product or quotient begins with "5", "6", "7", "8", or "9" (or vice versa) then give the quantity beginning with "1" one more significant figure than the other quantity. (Mnemonic: one gets one more.)
- B. If the least precise factor begins with a "2" and the product or quotient begins with "8" or "9" (or vice versa) then give the quantity beginning with "2" one more significant figure than the other quantity.
- C. If neither A nor B applies then use the Simple Rule.
- D. The product or quotient must have at least one significant figure.

For purposes of the 1-5 2-8 Rules, the first digit of a quantity is never zero. For example, the number 0.05916 is said to begin with a "5".

The original 1-7 Rule has annexed four new combinations: "1-5", "1-6", "2-8", and "2-9". Unlike the original 1-7 Rule, the 1-5, 2-8 Rules sometimes supersede the Simple Rule when they shouldn't. However they are in closer agreement with the Root Rule. Let's see why this is so.

Let the **critical numbers** of a multiplication (or division) calculation be defined as the mantissas of the least precise factor and of the product or quotient. In order to illustrate our approach, we'll look at the marginally useful 1-5 combination. The lower critical number begins with a "1", and the higher with a "5". If the lower critical number is less than

$$5/\sqrt{10} = 1.581$$

then the 1-5 Rule will agree with the LPF Root Rule. We will illustrate this point with two borderline cases. In the following example:

$$1.58 * 3.165 = 5.0007$$

the two critical numbers are 1.58 and 5.0007. Since they differ by a factor of 3.165, which is barely greater than $\sqrt{10}$, the LPF Root Rule and the 1-5 Rule barely agree that the product should be rounded to 5.0. In the following example, the two rules will barely disagree:

$$1.59 * 3.145 = 5.00055$$

(Use a similar line of reasoning.)

The 1-5 Rule and the LPF Root Rule sometimes agree when the lower critical number is greater than 1.581, but less than

$$5.999/\sqrt{10} = 1.897$$

as shown by the following example:

$$1.89 * 3.174 = 5.99886$$

in which both rules agree on "6.0".

Thus we would expect the 1-5 combination to conform to the LPF Root Rule in more than 58% (because of the "58" in our magic number, 1.581) of all possible cases for a given number of significant figures.

Now here is the other side of the coin. Suppose that we had used the 1-7 Rule. Since the 1-7 Rule does not cover the 1-5 combination, the Simple Rule would apply. Loosely interpreted, the above argument says that this would be wrong more than 58% of the time. Similar arguments can be constructed for the 1-6, 2-8, and 2-9 combinations.

If one had to choose the single most important thing to remember from this paper, then either the 1-5 Rule or the Three Rule (given in the next section) would be that choice. The 1-5 Rule is nice in the following ways:

- (a) It is easy to apply and to remember.
- (b) It rectifies the most blatant absurdities of the Simple Rule.

The Three Rule

The pedagogical weakness of the 1-5 Rule is that it is a magic formula; it gives no insight into the concept of keeping track of uncertainties in a calculation. For this reason we have developed the Three Rule, which places greater emphasis on propagation of uncertainty than either the Root Rule (on which it is based) or the 1-5 Rule.

The Three Rule:

- A. Locate the least precise factor. Calculate its relative uncertainty using the ± 1 convention.
- B. Convert the product or quotient to scientific notation, and multiply its mantissa by the relative uncertainty. This is the uncertainty of the product mantissa.
- C. Note the position (relative to the decimal point) of the first (non-zero) digit in the uncertainty of the product mantissa. Tentatively assume that the last significant digit of the product mantissa will be in the same position.
- D. Note the value of the first (non-zero) digit in the uncertainty of the product mantissa. If it is "1" or "2", everything is OK. Otherwise if it is "3" or more, then round off one more digit. (The rounded product or quotient must have at least one significant figure.)

As an example, let's use the Three Rule to round the product in our original problem:

$$1.43 * 4.098 = 5.86014$$

- A. The least precise factor is 1.43. It's implied relative uncertainty is $0.01/1.43 = 1/143$
- B. Uncertainty of the product mantissa = $(1/143) * 5.86014 = 0.04098$
- C. The uncertainty is in the hundredth's place of the product mantissa.
- D. The number in the hundredths' place (4) is not 1 or 2. Therefore we can only write the product mantissa with one significant digit after the decimal point. When properly rounded, the product is 5.9.

The Three Rule is slightly pessimistic, which allows for a significant contribution of uncertainty from factors other than the least precise factor. Unlike the 1-5 Rule, the Three Rule helps the user to develop the knack for keeping track of uncertainties in a calculation. The Three Rule is more comprehensive, but less facile than the 1-5 Rule.

Significant Figures and Science Education

Since 1981, the rudiments of significant figure theory have been taught in Chem. 104 at Oregon State University. Chem. 104, 105, 106 is the lower track General Chemistry sequence (for students whose high school backgrounds are weak in terms of chemistry and mathematics). Both the 1-5 Rule and the Root Rule Algorithm have been used at different times.

The Root Rule Algorithm took too much time to teach, and most students simply memorized the algorithm, without grasping the underlying concept. However the 1-5 Rule is quite teachable.

Chem. 104 students also learn that an exact number can never be a least precise factor, because an exact number has infinite precision. From our experience at Oregon State University, we feel that both the 1-5 Rule and the exact number convention are worthwhile and realistic learning objectives for lower division college science courses.

It may also be a good idea to remind students not to throw away information in the middle of a multi-step calculation, by overzealous rounding. Even if one is pressed for time, he or she should always retain at least two more significant figures in an intermediate result than in the least precise factor.

Unlike the 1-5 Rule, the Three Rule demands that students develop a certain level of mathematical maturity. This means that the Three Rule should have greater educational value than the 1-5 Rule, but that it may be less teachable. Therefore the Three Rule may be appropriate for more advanced courses, like physical chemistry.

The Three Rule may be more teachable than the Root Rule Algorithm because its syntax is simpler, because the magic number is simpler, and because its logic is more obvious. However the Three Rule has not yet been tested in the college science classroom.

Part I Summary and Conclusions

Since significant figure rules depend on propagation of uncertainty, we must define what we mean by "uncertainty". The most reasonable approach is to define uncertainty as the standard error of the mean.

We have presented several new rules for significant figures, in the attempt to avoid the Scylla of invalid generalization (the **Simple Rule**) and the Charybdis of ambiguity (the **Textbook Rule**). The basis for all of the new procedures is the **Root Rule**. The **Root Rule Algorithm** is a computer's eye view of significant figures, which allows the neophyte the opportunity to gain practice in applying the LPF Root Rule correctly, in order to master the underlying concept. This algorithm is the mathematical equivalent of training wheels on a bicycle. The **1-7 Rule** is a preliminary result applied in developing the 1-5 Rule, and is not useful out of this context. Although the **1-5 Rule** (the 'quick and dirty' approach) and the **Three Rule** (which places the most emphasis on propagation of uncertainty) follow from the Root Rule, they are easier to use and may have greater educational value than the parent rule.

The fundamental idea behind any valid significant figure rule is that the percent uncertainty of the product (or quotient) will approximate that of the least precise factor in a calculation. Therefore the number of significant figures used to express the product must imply an uncertainty, which is as close as possible to that implied by the least precise factor.

Algorithms based upon the Root Rule are an alternative to the oversimplified significant figure rule, generally taught in freshman chemistry and physics. The disadvantage of the Root Rule Algorithm is its complexity. The Three Rule is simpler than the Root Rule, and it offers greater insight into the mechanics of propagation of uncertainty. However it has not been tested in the college science classroom.

The 1-5 Rule is more facile than the Three Rule. Since the 1-5 Rule has a simpler syntax and is easier to memorize than the Three Rule, it may also be more teachable. However the Three Rule is more comprehensive. Moreover the Three Rule has greater heuristic value.

Experience has shown that it is practical to teach the 1-5 Rule in General Chemistry courses. Perhaps the Three Rule would be more appropriate for courses at a higher level (eg. physical chemistry). Table I summarizes the relative merits of the Simple Rule and the three major new rules for significant figures in multiplication and division calculations.

Table I. A Comparison of the Significant Figure Rules.

	Correct and comprehensive	Easy to apply, to remember, and to teach	Facilitates understanding of underlying principle
Root Rule	****	*	***
Three Rule	***	**	****
1-5 Rule	**	***	**
Simple Rule	*	****	*

In Part II, we will see that it is possible to articulate a more comprehensive rule, which will integrate significant figure theory with simple propagation of uncertainty for the general case of functions of n real variables, each of which contributes appreciable uncertainty.

PART II: SIGNIFICANT FIGURES AND SIMPLE UNCERTAINTY PROPAGATION THEORY

The Extended Root Rule

In Part I, we considered the LPF Root Rule as a refinement of the Simple Rule for significant figures. Both rules are quite limited in scope. They apply only to multiplication and division calculations in which

- (a) a single least precise factor (LPF) exists and can be identified
- (b) the least precise factor contributes essentially all of the uncertainty to a product or quotient
- (c) the standard error of the mean definition of uncertainty is used.

It would be desirable to have a more comprehensive significant figure rule.

In the general case, it is necessary to arrive at an **estimated relative uncertainty** of the calculated result, before assigning significant figures. Unless political considerations dictate otherwise, one may use the following rule.

The Extended Root Rule:

When the result of a calculation is properly rounded, its estimated relative uncertainty to implied relative uncertainty ratio must be between $1/\sqrt{10}$ and $\sqrt{10}$ (except when this would mandate a result with less than one significant figure).

The estimated relative uncertainty can be calculated using two different propagation-of-uncertainty formulas, one for each of the two definitions of uncertainty (from Part I). Equation II.1 is for the absolute limit of error type of uncertainty. Equation II.2 is for the standard error of the mean type of uncertainty (which we will emphasize).

$$\text{ERU} = \frac{\Delta f}{f} = \frac{1}{f} \cdot \sum_{i=1}^n \frac{df}{dx_i} \cdot \Delta x_i + \text{RUM} \quad (11.1)$$

$$\text{ERU} = \frac{\Delta f}{f} = \frac{1}{f} \left[\sum_{i=1}^n \left\{ \frac{df}{dx_i} \cdot \Delta x_i \right\}^2 \right]^{1/2} \quad (11.2)$$

where the x_i in 11.2 are the measured values of n mutually independent quantities, the Δx_i are the absolute values of the implied uncertainties of the x_i , f is a function of the x_i , Δf is the absolute value of the uncertainty of f , and $\Delta f/f$ is the absolute value of the estimated relative uncertainty. **RUM** is the maximum relative uncertainty of the method. In a purely theoretical example—such as the area of a perfect rectangle—we may not need to concern ourselves with this. However in some practical applications—like real gas behavior—the calculation method is an approximation, and is itself a source of error. Since the standard error of the mean is really a standard deviation, Equation 11.2 boils down to additivity of variance. In using both equations we must assume that the Δx_i are all reasonably small (which is not always true, especially for x_i which have only one significant figure).

If the components of a physical calculation are represented by stand-alone significant figures, then the Extended Root Rule and Equation 11.2 will allow us to express the result in the same way. Moreover the uncertainties of any of the calculation's components and of the calculation's result can always be pinned down within a ten-fold range.

A consequence of the Extended Root Rule and of Equation 11.2, is that at least one significant digit will be lost for **homogeneous addition** calculations in which 10 or more quantities are summed. Consider the following addition calculation:

$$\begin{array}{r}
 12.3 \\
 57.1 \\
 11.3 \\
 17.1 \\
 92.3 \\
 29.3 \\
 13.7 \\
 41.4 \\
 34.7 \\
 \underline{53.5} \\
 362.7
 \end{array}$$

Since we are adding 10 quantities, all of which have appreciable uncertainty in the tenths' place (hence the homogeneity), the sum should be rounded to 363. Here we are assuming that the uncertainty in the tenths' place is the same for each quantity. If we insist on using the absolute limit of error type of uncertainty with homogeneous addition, then we will lose a significant figure when there are four or more terms!

As another example, let's suppose that $x = 1.111$ and that $f(x) = \log(x) = 0.045714$. By Equation 11.2,

$$\frac{\Delta f}{f} = \frac{df}{dx} \cdot \frac{\Delta x}{f} = \frac{\log e}{x \log x} \cdot \Delta x = \frac{0.001 * \log e}{1.111 * 0.045714} = 0.00855$$

Therefore when correctly rounded, the logarithm is 0.046, since

$$\frac{0.001/0.0457}{0.00855} = 2.56 < \sqrt{10}$$

Note that two significant figures were lost.

As another example, let's apply the Extended Root Rule to a gas chromatography calculation. The affinity of a particular compound for a given stationary phase is commonly expressed in terms of the **retention index I** which can be calculated as follows (writing "ln"

instead of the conventional base 10 log for mathematical simplicity, since it makes no difference in the resulting value of I):

$$I = 200 * \frac{\ln x - \ln t_6}{\ln t_8 - \ln t_6} + 600$$

where t_6 and t_8 are the sorbed times of n-hexane and n-octane, which happen to precede and to follow the sample peak, whose sorbed time is x . The numerical values of t_6 , x , and t_8 are 10.5, 18.8, and 53.0, respectively. The units are "mm of chart paper". Converting these units to units of seconds or minutes will not affect the value of I .

$$I = 200 * \frac{\ln 18.8 - \ln 10.5}{\ln 53 - \ln 10.5} + 600 = 671.95946$$

$$\frac{dI}{dt_6} = 200 * \frac{[(\ln t_8 - \ln t_6) - (\ln x - \ln t_6)] * (-1/t_6)}{(\ln t_8 - \ln t_6)^2}$$

$$= \frac{200 * (\ln x - \ln t_8)}{t_6 * (\ln t_8 - \ln t_6)^2} = \frac{200 * (\ln 18.8 - \ln 53)}{10.5 * (\ln 53 - \ln 10.5)^2} = -7.532$$

$$\frac{dI}{dx} = \frac{200}{x * (\ln t_8 - \ln t_6)} = \frac{200}{18.8 * (\ln 53 - \ln 10.5)} = 6.571$$

$$\frac{dI}{dt_8} = \frac{-200 * (\ln x - \ln t_6)}{t_8 * (\ln t_8 - \ln t_6)} = \frac{-200 * (\ln 18.8 - \ln 10.5)}{53 * (\ln 53 - \ln 10.5)} = -0.839$$

Without loss of generality, we will use a ± 1 convention for all of the above variables. (A different \pm convention would not have a great effect on the significant figure determination.)

Thus $\Delta t_6 = \Delta x = \Delta t_8 = 0.1$ By Equation 11.2,

$$\frac{\Delta f}{f} = \frac{\Delta t}{f} * \left[\left\{ \frac{dI}{dt_6} \right\}^2 + \left\{ \frac{dI}{dx} \right\}^2 + \left\{ \frac{dI}{dt_8} \right\}^2 \right]^{1/2}$$

$$= \frac{0.1 * [(-7.532)^2 + 6.571^2 + (-0.839)^2]^{1/2}}{671.96} = 0.001493$$

This is the estimated relative uncertainty. The next step is concerned with the implied relative uncertainty. We need to determine the number of significant figures necessary to minimize the discrepancy between the values of the implied relative uncertainty and of the estimated relative uncertainty (**ERU**). It turns out that the properly rounded answer is 672 (three significant figures). The implied relative uncertainty (**IRU**) of 672 is $1/672 = 0.001488$. Then

$$\frac{\text{ERU}}{\text{IRU}} = \frac{0.001493}{0.001488} = 1.04 < \sqrt{10}$$

and the Extended Root Rule is satisfied.

More Boundary Conditions

In Part I we showed that the LPF Root Rule was relatively independent of \pm convention. The same applies to the Extended Root Rule. The purpose of this rather arcane section is to deal with those few cases where we do have to worry about the \pm convention. Our frame of reference will be the standard error of the mean definition of uncertainty.

The Extended Root Rule applies when not all of the numerical values of the uncertainties (Δx_i) are available. If the numerical values for all of the Δx_i are given, then what is the point in being limited to stand-alone significant figures? As in Part I, we assume that the same \pm (hindsight) convention applies for estimating the unknown uncertainties of the x_i . We are free to choose any reasonable \pm convention. Usually a ± 1 convention is easiest to work with. For ordinary significant figure determinations, our results are not usually affected by this choice. However the \pm convention is relevant for **mixed significant figure determinations**, in which some—but not all—of the components' estimated uncertainties are available.

Thus far, we have approached significant figures from a **hindsight** perspective. The experimenter deposits information about estimated uncertainty within the number of significant figures of a measurement-based quantity. Then we see the number, and interpret it in hindsight, inferring some uncertainty not specified by the experimenter. In the multiplication and division calculations of Part I, our task has been to transfer this inferred uncertainty from the least precise factor (LPF) to the product or quotient.

Now we will ask a new kind of question. What is the most reasonable hindsight (\pm) convention? Our answer will depend upon which rounding convention we choose, that is how much uncertainty is allowed in the last significant digit of the data being manipulated.

We will choose the most generous rounding convention allowed by the Mainstream approach. The maximum uncertainty we will allow on the last significant digit of a measurement-based quantity is ± 5 . Consider the following example:

$$9.23 \pm 0.06$$

The absolute value of the uncertainty (6) on the tentative last significant digit (on the "3") is greater than 5. Therefore the "3" should be rounded off, leaving only two significant figures.

$$9.23 \pm 0.06 \rightarrow 9.2$$

If an experimenter follows the custom that estimated uncertainties greater than ± 5 on any digit will be eliminated by rounding off that digit, then we may infer that the last reported digit has an estimated uncertainty no greater than ± 5 , and no less than ± 0.5 . (Obviously the upper uncertainty limit does not always apply to the anomalous one-significant-figure-quantities.) We have no way of knowing from the reported datum what the estimated uncertainty really was, except that it was between these limits. Since the limits are an order of magnitude apart, we will argue from the same logic that led to the Root Rule, that the best guess of the estimated uncertainty on the last digit will be half an order of magnitude into the range, or $\pm 5/\sqrt{10} = \pm 1.58$.

Now we'll take an indirect route back to the hindsight conventions we have just described. We will use the following borderline case of the LPF Root Rule to pin down some uncertainties:

$$22.207/2.65 = 8.38$$

Our argument will hinge upon the common-sense notion that if a calculation is borderline with respect to the LPF Root Rule, then it should also be borderline with respect to the most reasonable hindsight convention. Let A and B be the expected* uncertainties on the last digit of the least precise factor, and of the tentative last significant digit of the quotient, respectively. Then

$$\frac{A}{2.65} = \frac{B}{8.38} \qquad B = \frac{8.38}{2.65} * A \approx A \sqrt{10} \qquad A \approx \frac{B}{\sqrt{10}}$$

*In this argument, we are not using "expected" in its mathematical sense; here "expected" refers to the median, rather than the mean.

Notice that in this borderline case the least precise factor and quotient mantissas differ by a factor of approximately $\sqrt{10}$. Since this is near the maximum allowed by the LPF Root Rule, it suggests that the uncertainty on the tentative last significant digit of the quotient has been stretched to the allowable maximum. Thus we would expect the uncertainty on the final "8" (in the quantity "8.38") to be approximately ± 5 . (We would not expect this uncertainty to be ± 6 or ± 4 , because then the quotient would definitely have two significant figures, or three significant figures, respectively.) Thus

$$A = 5/\sqrt{10} = 1.58$$

For our ± 0.5 to ± 5 rounding convention, the most reasonable hindsight convention is approximately ± 1.58 . Note that 1.58 is the geometric mean of 0.5 and 5. In general, the hindsight convention value is approximately equal to the geometric mean of the rounding convention limits.

In a mixed significant figure determination, where only some of the components' estimated uncertainties are known, the ideal procedure should be as follows:

- 1) Apply the hindsight (\pm) convention value to the last significant digit of any calculation component, whose estimated uncertainty is not known.
- 2) Plug these values, together with the given estimated uncertainties, into Equation 11.2.
- 3) Then proceed as before, using the Extended Root Rule.

In the absence of standardized rounding and hindsight conventions, this variation on our basic theme is not yet possible. However the hindsight convention concept has broader implications. In this discussion, we have ignored the effect of rounding error, a situation which we now will remedy.

The reporting of a measurement-based quantity involves the following three types of error:

- 1) Random error.
- 2) Systematic error.

- 3) **Rounding error.** An example of a pure rounding error is that which arises when π is rounded to 3.14 (or to 3.141592654 for that matter). The very fact of rounding introduces an error, and sometimes this must be taken into consideration. Rounding error must be distinguished from **measurement error**, which includes (1) and (2) above.

According to Sheppard⁴, the variance of a rounded variable equals the variance of the unrounded variable plus $R^2/3$, where $\pm R$ are the rounding error limits. Recall that the ± 1.58 hindsight convention (associated with the ± 5 rounding convention) refers to measurement uncertainty only. The corresponding **total uncertainty hindsight convention** would be estimated as follows (remembering that 1.58 is an approximation of $5/\sqrt{10} = \sqrt{2.5}$):

$$\pm\sqrt{2.5 + 0.5^2/3} = \pm 1.61$$

which is not very far from ± 1.58 .

At this point, we have three choices. First we could amend (and complicate) the Root Rule in order to take rounding error into account. Secondly we could continue to ignore the effect of rounding error. Lastly, we could retain the Root Rule, and multiply the right hand side of Equation 11.2 by a correction factor. In this case, the correction factor would be calculated as follows:

$$F_{1.58} = \left[\frac{2.5 + 1/12}{2.5} \right]^{1/2} = \left[\frac{31}{30} \right]^{1/2} = 1.0165$$

It must be understood that the rounding error correction factor is a function of the hindsight convention (for measurement uncertainty). For example, if our hindsight convention is ± 1 (which goes along with $\pm 1/\sqrt{10}$ to $\pm\sqrt{10}$ rounding convention limits), then this correction factor would be calculated as follows:

$$F_1 = \left[\frac{1^2 + 1^2/12}{1^2} \right]^{1/2} = \left[\frac{13}{12} \right]^{1/2} = 1.0408$$

Our main purpose in exhuming rounding error is to show that it is relatively inconsequential for ordinary significant figure determinations involving the standard error of the mean type of uncertainty. However rounding uncertainty would be relatively important for mixed determinations in which any of the uncertainties given are substantially less than the ± 1.58 hindsight convention would suggest. Since the above correction factor (F) increases as the hindsight (\pm) convention decreases, rounding uncertainty must be taken into account if the absolute value of the hindsight convention is substantially less than unity.

For significant figure determinations which use the absolute limit of error type of uncertainty, the total uncertainty hindsight convention is

$$\pm(1.58 + 0.5) = \pm 2.08$$

Therefore the LPF Root Rule would have a decidedly optimistic bias if the absolute limit of error type of uncertainty was employed, unless the Liberal approach was used also. (See Part I.)

In this section, we have seen how the concept of rounding convention leads to that of hindsight convention, which should be taken into account during mixed significant figure determinations. We also considered rounding error, and its effects on the total uncertainty hindsight convention vis-a-vis the absolute limit of error and standard error of the mean definitions of uncertainty (See Part I). All of these considerations underscore the importance of establishing a standardized rounding convention.

Part II Summary and Conclusion

We have seen that the Root Rule is a rational basis for significant figure determinations. The Extended Root Rule says that we must compare the estimated relative uncertainty and implied relative uncertainty for the rounded numerical result R of a chemical or physical calculation. The estimated relative uncertainty and implied relative uncertainty of R must be in the same ballpark, i.e. within half an order of magnitude of each other. If they are, then R has been rounded to the correct number of significant figures.

In Part II, we saw that the value of the estimated relative uncertainty we obtain depends on which definition of uncertainty we use. This needs to be standardized.

In Part I, we considered only the special case where the estimated relative uncertainty of a product or quotient is approximately equal to the implied relative uncertainty of the least precise factor for a multiplication or division calculation, in which the standard error of the mean definition of uncertainty is used. This leads to the LPF Root Rule (and to the handy 1-5 Rule). Thus the LPF Root Rule of Part I is a special case of the Extended Root Rule.

All of these procedures can be incorporated into computer programs. Some calculators (eg. the Hewlett-Packard 15C) allow the user to choose in advance the number of significant figures which will appear in the display. Perhaps scientific calculators of the future will have the capability of doing significant figure determinations. This raises the possibility of eliminating extraneous-digit noise from modern electronic calculators (and from freshman chemistry lab reports).

APPENDIX

First we will show that the LPF Root Rule and the Simple Rule give different results more than 18% of the time in two-factor multiplication calculations. We will make the following assumptions:

- (a) All (base 10) mantissas are equally likely.
- (b) The two factors do not have the same number of significant figures.

This second assumption is reasonable. If both factors have the same number of significant figures, then it is not true that the LPF contributes essentially all of the uncertainty. Therefore by hypothesis, neither the Simple Rule nor the LPF Root Rule should be used. Moreover we will ignore the problems created by the anomalous one-significant-figure-quantities.

Let M_1 and M_2 be the smaller and larger mantissas of the two factors. The LPF can override the Simple Rule only when the following conditions are both met:

- 1) $M_1 < \sqrt{10}$
- 2) $M_2 > \sqrt{10}$

Suppose that Conditions 1 and 2 are true. Then we have the following cases and subcases to consider:

- I. The product mantissa is less than $\sqrt{10}$.
- II. The product mantissa is greater than $\sqrt{10}$.
 - A. The LPF mantissa is less than $\sqrt{10}$.
 - B. The LPF mantissa is greater than $\sqrt{10}$.

There will be an override (of the Simple Rule) if and only if either IB or IIA is true. In Equations A.1-3, we will assume that Conditions 1 and 2 (above) are true. Therefore the probabilities (represented by P) in these equations are conditional. By Conditions 1 and 2,

$$P(IA) = P(IB) \text{ and } P(IIA) = P(IIB) \quad (A.1-2)$$

Therefore $P(1B) + P(1IA) = P(1A) + P(1IB)$ (A.3)

Thus the conditional probability (P_1) of an override, given (1) and (2), is exactly 0.5.

The probability that Conditions 1 and 2 are both true is given by the following:

$$P_2 = \frac{\sqrt{10} - 1}{10 - 1} \cdot \frac{10 - \sqrt{10}}{10 - 1} \cdot 2 \quad (A.4)$$

Therefore the unconditional override probability is given by the following:

$$P = P_1 P_2 = 0.5 \cdot \frac{\sqrt{10} - 1}{9} \cdot \frac{10 - \sqrt{10}}{9} \cdot 2 = 18.3\% \quad (A.5)$$

In division calculations, the LPF Root Rule will override the Simple Rule nearly 30% of the time. Here, we need to consider the following cases and subcases:

- I' The least precise factor is in the numerator.
- A' The quotient mantissa (Q) is less than the least precise factor mantissa (L).
- B' $Q > L$
- II' The least precise factor is in the denominator.

Case IA' Suppose that the least precise factor is in the numerator. Let M be the mantissa of the most precise factor. Let $P(IA')$ be the probability of a IA' division override. Override will occur if and only if $\sqrt{10} < M < L$. Therefore

$$P(IA') = 0.5 \left[\frac{10 - \sqrt{10}}{10 - 1} \right]^2 = 5 \left[\frac{\sqrt{10} - 1}{9} \right]^2 \quad (A.6)$$

Case IB' Override will occur if and only if $L < M < \sqrt{10}$. Therefore

$$P(IB') = 0.5 \left[\frac{\sqrt{10} - 1}{10 - 1} \right]^2 \quad (A.7)$$

$$P(I') = P(IA') + P(IB') = 5.5 \left[\frac{\sqrt{10} - 1}{9} \right]^2 \quad (A.8)$$

Case II' is somewhat more complicated. Let x be the denominator mantissa. We must consider the following four subcases:

- 1) $1 < x < 10^{.25}$
- 2) $10^{.25} < x < \sqrt{10}$
- 3) $\sqrt{10} < x < 10^{.75}$
- 4) $10^{.75} < x < 10$

Let \bar{P}_i' be the probability that subcase i applies. The \bar{P}_i' are calculated as follows:

$$\bar{P}_1' = \frac{10^{.25} - 1}{10 - 1} = 0.08647549 \quad (A.9)$$

$$\bar{P}_2' = \frac{\sqrt{10} - 10^{.25}}{10 - 1} = 0.1537775833 \quad (A.10)$$

$$\bar{P}_3' = \frac{10^{.75} - \sqrt{10}}{10 - 1} = 0.2734595102 \quad (A.11)$$

$$\bar{P}_4' = \frac{10 - 10^{.75}}{10 - 1} = 0.4862874164 \quad (A.12)$$

Let P_i' be the conditional probability that there will be an override, given that subcase i applies.

In subcase 1, override will not occur if and only if the numerator mantissa is between x and $x^2\sqrt{10}$. If x is fixed, then

$$1 - P_1'(x) = \frac{x^2\sqrt{10} - x}{10 - 1} \quad (\text{A.13})$$

The subcase 1 frequency function is as follows:

$$f_1(x) = 1/(10^{25} - 1) \quad (\text{A.14})$$

Therefore the average (expected) value P_1' of $P_1'(x)$ is calculated as follows:

$$\begin{aligned} P_1' &= 1 - \int_1^{10^{25}} P_1'(x) f_1(x) dx = \int_1^{10^{25}} \frac{x^2\sqrt{10} - x}{9(10^{25} - 1)} dx \\ &= 0.458581526 \end{aligned} \quad (\text{A.15})$$

The subcase 2 calculations are as follows:

$$P_2'(x) = \frac{x - x^2/\sqrt{10}}{10 - 1} \quad f_2(x) = \frac{1}{\sqrt{10} - 10^{25}} \quad (\text{A.16-17})$$

$$\begin{aligned} P_2' &= \int_{10^{25}}^{\sqrt{10}} P_2'(x) f_2(x) dx = \int_{10^{25}}^{\sqrt{10}} \frac{x - x^2/\sqrt{10}}{9(\sqrt{10} - 10^{25})} dx \\ &= 0.05445476057 \end{aligned} \quad (\text{A.18})$$

The subcase 3 calculations are as follows:

$$P_3'(x) = \frac{x^2/\sqrt{10} - x}{10 - 1} \quad f_3(x) = \frac{1}{10^{.75} - \sqrt{10}} \quad (\text{A.19-20})$$

$$P_3' = \int_{\sqrt{10}}^{10^{.75}} P_3'(x) f_3(x) dx = \int_{\sqrt{10}}^{10^{.75}} \frac{x^2/\sqrt{10} - x}{9(10^{.75} - \sqrt{10})} dx$$

$$= 0.206723907 \quad (\text{A.21})$$

The subcase 4 calculations are as follows:

$$P_4'(x) = \frac{x - x^2/\sqrt{1000}}{10 - 1} \quad f_4(x) = \frac{1}{10 - 10^{.75}} \quad (\text{A.22-23})$$

$$P_4' = \int_{10^{.75}}^{10} P_4'(x) f_4(x) dx = 1 - \int_{10^{.75}}^{10} \frac{x - x^2/\sqrt{1000}}{9} dx$$

$$= 0.3520532293 \quad (\text{A.24})$$

The override probability (given that the least precise factor is in the denominator) is calculated as follows:

$$P'(II) = \sum_{i=1}^4 \bar{P}_i' P_i' = 0.2760191157 \quad (\text{A.25})$$

If we assume that the least precise factor will end up in the numerator as often as in the denominator, then the unconditional override probability (P') for division calculations is given by the following:

$$P' = 0.5 [P'(I) + P'(II)] = 29.7\% \quad (\text{A.26})$$

If we assume that division and multiplication are equally frequent, then the overall override probability would be about 24%, since 24% is the average of 18.3% (for multiplication) and 29.7% (for division).

In the above arguments, we have assumed that the mantissa frequency function f is a constant ($1/9$ in the unrestricted case). A more realistic approach would set $f = c/x$, where x is the mantissa, and c is a constant ($c = 1/\ln 10$ in the unrestricted case). A similar argument, using this second assumption leads to the result that

$$P = P'(I) = P'(II) = 25\% \text{ exactly} \quad (\text{A.27})$$

Thus the Simple Rule not only breaks down, but it does so with embarrassing frequency.

**DECIMAL CODED BINARIES: A TECHNIQUE FOR COMPRESSING A
MEASUREMENT AND ITS UNCERTAINTY INTO A SINGLE QUANTITY**

By Lawrence D. Fields and Stephen J. Hawkes*

Department of Chemistry
Oregon State University
Corvallis, Oregon 97331
(503) 754-2081

submitted for publication to Analytical Chemistry

ABSTRACT

The theoretical and applied aspects of Decimal Coded Binaries, a new method for compacting scientific data into a tabular format, are explored. This technique efficiently combines a measurement and its uncertainty into a single easy-to-read quantity. DCB compares favorably with a pedestrian reference technique for accomplishing the same end (and with ordinary stand-alone significant figures). The DCB algorithm has been programmed into a hand-held scientific calculator, and into a simple microcomputer which is used as an interface with chemical instruments commonly utilized in quantitative analysis.

INTRODUCTION

Suppose that you are giving complicated street directions, and that the other person's attention seems to be flagging. Unconsciously you may raise the pitch of your voice for the last word of a sentence. ("And then you turn left.") A casual eavesdropper will understand the "turn left", but may jump to the conclusion that you are asking a question, when in fact your tone of voice is a polite way of saying, "Listen to me!" Of course, the intonation change in the word "left" will require a little extra time because the vowel is drawn out slightly, but less time than it would take to say, "Listen to me!" Thus the use of inflection can enhance the efficiency of communication. In the above parable the way we end the "turn left" sentence has an indirect meaning, above and beyond the meaning of the words themselves.

We will describe a new method for compressing scientific data—Decimal Coded Binaries, or DCB, and an auxiliary technique, Linear Code. Like the sentence in the parable, a quantity which has been coded by either technique will have both a direct and an indirect meaning.

The direct meaning is the number itself, which is approximately equal to the value of the measurement. The "indirect meaning" is the information about the measurement's uncertainty, which we will infer from the last digit of a Linear Coded quantity, and from the last several binary digits of a DCB quantity.

Like the sentence in the parable, DCB is efficient. DCB requires appreciably less space on paper or in memory devices, and less time for telecommunicating than would be required to write, store, or transmit the value of the measurement and its uncertainty separately. Both techniques are user-friendly in the sense that it is easy to program coding and decoding routines into ordinary microcomputers and HP15C calculators. Before we describe these coding techniques, we will discuss the reporting of measurement-based data.

There is a consensus among physical scientists that when a measurement is reported, some indication of its trustworthiness should also be given. There are two widely-used methods for expressing the trustworthiness of a measurement.

In the direct approach, the experimenter gives a full accounting of the measurement and its limitations. He or she will report how reproducible the results are and also discuss what is known or surmised about the systematic error. NBS Handbook 91⁵ gives a very comprehensive treatment of the direct approach. However this scientific prescription for description is honored more in the breach than in the observance.

Sometimes it is impractical to be this explicit, especially in data tables. When brevity and readability are the chief concerns, stand-alone significant figures are often used instead of the direct approach, but there are serious problems associated with this usage of significant figures.

Significant figures are a sacred cow in the sense that most scientists can agree about the sacredness, but few can agree on the nature of the cow! Almost every physical scientist has a strong opinion about significant figures, and regards his or her interpretation as being obviously correct. However there are truly fundamental differences of interpretation to be found in the literature.

The current NBS usage⁶ of stand-alone significant figures is quite narrow. There is an upper bound on the maximum relative uncertainty allowed on the last reported digit of a measured quantity, but there is no lower bound. Here the last reported (significant) digit must be known with certainty; i.e. the estimated uncertainty on the last reported digit must not exceed ± 0.5 . (We will attempt to define uncertainty shortly.) In an extreme case, a mass measurement of 123.4 mg, with an estimated uncertainty of 0.6 mg, would have to be reported as 1.2×10^1 mg since the third digit could be "3" or "4".

In the following example, we will consider the question: What is the proper number of significant figures that should be used to write the atomic weight of aluminum, according to the NBS?

The atomic weight of aluminum is

- (a) 27, correct to two significant figures.
- (b) 27.0, correct to three significant figures.
- (c) 26.98, correct to four significant figures.

According to the NBS, all of the above are syntactically correct. Therefore there is no unique answer to the question we have just posed. In the NBS Weltanschauung, significant figures are not supposed to give very much information about the uncertainty of a measurement.

Skoog and West⁷ take a broader approach, in which stand-alone significant figures are used to give more information about the estimated uncertainty. Here the estimated uncertainty on the last significant (reported) digit must be between ± 0.5 and ± 5 . Using this ± 0.5 to ± 5 rounding convention, the mass measurement example would be rounded to 123 mg. The "3" in "123" has meaning even though it is not known with certainty.

In this general approach, the reader of the data can infer that the estimated uncertainty must be within a particular ten-fold range (which is not to be confused with a confidence interval). This is not true for the narrow NBS approach.

A second problem is that there is a lack of consensus on the meaning of uncertainty. Let's review some fundamentals.

There are two sources of measurement error: **random error** (sometimes called imprecision) and **systematic error** (sometimes called determinate error). Random error occurs when repeated measurements of the same thing yield somewhat different values. The smaller the random error, the greater the precision. A measurement will be relatively accurate if both random and systematic errors are small.

Skoog and West have suggested that systematic (determinate) errors may be classified with respect to their sources and their effects.

"It is not possible to list all conceivable sources of determinate error; we can, however, recognize that they have their origin in the personal errors of the experimenter, the instrumental errors of his measuring devices, the errors that repose in the method of analysis he employs, or any combination of these."⁸

"Determinate errors generally fall into either of two categories, constant or proportional. The magnitude of a constant error is independent of the size of the quantity measured. On the other hand, proportional errors depend in absolute magnitude upon, and increase or decrease in proportion to, the size of the sample taken for analysis."⁹

We will add a third classification scheme. All systematic errors come under one of the following two categories:

- 1) suspected systematic errors
- 2) those we are completely unaware of, i.e. lurking systematic errors. By definition, lurking systematic errors cannot be compensated for, or taken into account.

Uncertainty is a key concept in this discussion. Both random error and systematic error can undermine our sense of certainty in a measurement, but what does uncertainty really mean? Unfortunately, a direct answer to such a philosophical question is beyond the scope of this paper. However we can approach the question in a pragmatic fashion. What kind of uncertainty is most useful to us? We will compare four types (definitions) of uncertainty.

- 1) The first type of uncertainty is the absolute limit of error in a measurement. The advantages of using this kind of uncertainty are that both random error and suspected systematic error are included, and that the simple worst case method¹⁰ of error analysis can be used to trace the propagation of uncertainty in a calculation. By definition, lurking systematic error cannot be taken into account; hence the "absolute limit of error" of a measurement is unknown and unknowable. Moreover if the random error is normally distributed, as we so often assume, then these limits don't even exist, since the

Gaussian frequency function will yield a non-zero probability density for any real value of the random variable. In an extreme case, this means that there is a non-zero (albeit very small) probability that any truly normal random variable will be 10 million standard deviations away from the mean of the distribution.

- 2) The second type of uncertainty we will consider is the maximum suspected systematic error. Unlike (1) above, this kind of uncertainty could apply to the real world. If the suspected systematic error is large, relative to the imprecision, then the reader of the data would like to know about it. However it would be difficult to reduce even the suspected systematic error to a single number. If the suspected systematic error is mentioned at all when a measurement is reported, the experimenter will usually attempt to discuss it thoroughly. The next two classes of uncertainty are functions of the imprecision.
- 3) Our index of uncertainty could be one half of a 95% confidence interval, the **half-span**. However there are two limitations:
 - (a) Systematic error cannot be taken into account, since this kind of uncertainty applies only to random error.
 - (b) Confidence intervals do not lend themselves to propagation-of-uncertainty analysis. (See Appendix.)
- 4) The standard error of the mean is a better way of expressing the uncertainty stemming from imprecision. This definition, like the preceding one, applies only to random error, and not to systematic error. However propagation-of-uncertainty analysis is relatively simple under this definition (unlike Definition 3); it applies to the real world (unlike Definition 1); and the standard error of the mean can be written quite compactly (unlike a suspected systematic error, which may require several paragraphs of explanation and caveats).

Our primary objective in developing these data compression techniques is to provide a compact format, which will enable the experimenter to report a measurement, together with an index of its trustworthiness. If neither the random error nor the suspected systematic error are negligible, then both types of error limit the

trustworthiness of the measurement. A satisfactory definition of uncertainty must make sense and it should take both types of error into account.

None of the above definitions are fully adequate, although (4) is the most reasonable, and it is this standard error of the mean definition that we will emphasize. We will denote this type of uncertainty by **Um**, an abbreviation for the **estimated uncertainty of the measurement**.

Now that we have exhausted uncertainty definitions, we must consider one final problem with stand-alone significant figures, as they are ordinarily used. Even with the general approach of Skoog and West (which we will use in modified form), the estimated uncertainty on the last significant digit of a measurement-based quantity can be anywhere within a ten-fold range (because of the base ten number system). In the next two sections, we will describe two types of data compression techniques, which address this problem. In order to make Decimal Coded Binaries (DCB) more understandable, we will develop an auxiliary technique in the next section. DCB will be introduced in the section after that.

LINEAR CODE

Now we will return to the linguistic parable introduced in the previous section, where we had briefly discussed inflection, a commonly-used speech compression technique. Clearly this is not the only way to condense communications. The obvious approach is first to use abbreviations whenever possible, then to eliminate the non-essential words (eg. articles), then finally to delete the periods at the ends of sentences and the spaces between words. We will contemplate how such a sequence of compaction steps could apply to quantitative data. Then we will see how **Linear Code** puts this Western Union approach into practice.

As a first step, we could decide how many digits to report for a measured value. (How many significant figures does the quantity have?) Then we would round off the number accordingly.

Secondly, we could use a one-digit code to indicate the imprecision on the last reported digit of the measured value. This **uncertainty code** would give the reader of the data the upper and lower bounds for the estimated standard error of the mean.

The last steps in this pedestrian data compaction technique would be to condense the format for reporting the uncertainty code. Instead of writing "standard error of the mean = " or " \pm ", we could simply leave a space between the measured value and the uncertainty. Finally we could delete even the above space, by tacking the uncertainty code onto the end of the **mantissa** of the number representing the measured value. (Suppose that a number expressed in scientific notation is 3.16×10^{-2} . The "3.16" is called the mantissa.)

Now let's talk about the actual technique. The following two steps summarize the procedure for expressing a measurement in Linear Code:

1. Round the value of the measurement-based quantity to the correct number of significant figures, using scientific notation if necessary.
2. Insert a digit after the last significant figure. This extra digit is the **uncertainty code**, and it has nothing to do with the measurement's value.

Table II illustrates the uncertainty code concept, as tentatively applied to the ± 0.5 to ± 5 rounding convention of the previous section.

Table II Relationship Between Uncertainty Code and Estimated Measurement Uncertainty with the ± 0.5 to ± 5 Rounding Convention.

Uncertainty Code										
0	1	2	3	4	5	6	7	8	9	
.5	.63	.79	1	1.3	1.6	2	2.5	3.2	4	5
Estimated measurement uncertainty										

Note that Table I uses a semilogarithmic scale for the bottom row. This allows the estimated measurement uncertainty to be pinned down within a $10^{0.1}$ -fold range. For instance, if the uncertainty code is 4, then the estimated measurement uncertainty on the last significant digit would be between 1.3 and 1.6. As an example, the Linear-Coded quantity "59.4 ppm" translates into plain English as "59 ppm, with an estimated measurement uncertainty between 1.3 ppm and 1.6 ppm". (This is not the same as a confidence interval.) However there is a fly in the ointment.

Thus far, we have only considered uncertainties which come from measurement errors; **rounding errors** also need to be taken into account. An example of a pure rounding error is that which arises when π is rounded to 3.14 (or to 3.14159265 for that matter). According to Sheppard¹¹, the **total standard deviation** s_T of a reported measurement can be calculated as follows:

$$s_T = (s^2 + R^2/3)^{1/2} \quad (1)$$

where s is the standard deviation of the measurement, and R is the maximum possible rounding error (which is ± 0.5 on the last significant

digit). Now we will introduce the concepts of **rounding ratio** and **estimated total uncertainty** U_T .

$$\text{rounding ratio} = U_m/R \quad (2)$$

$$U_T = (U_m^2 + R^2/3)^{1/2} \quad (3)$$

where U_m is the estimated measurement uncertainty, as defined in the previous section. Note that Equations 1 and 3 are isomorphic.

Eisenhart¹² recommends that a quantity be rounded such that the rounding ratio is at least two. This suggestion can be appreciated with the help of the concept of the **% inflation** (of uncertainty).

$$\% \text{ inflation} = 100(U_T - U_m)/U_m \quad (4)$$

The minimum estimated measurement uncertainty (U_m) we have been allowing on the last reported digit is ± 0.5 . By Equations 3 and 4, this would result in a 15% inflation; i.e. the total estimated uncertainty would be 15% more than the estimated measurement uncertainty. This would be very frustrating for the poor experimenter who has been so diligent in controlling temperature, pressure, humidity, and other extraneous variables, only to have a rounding convention inflate his or her imprecision by 15%! Table III summarizes the relationship between estimated measurement uncertainty (U_m), rounding ratio, and inflation.

Table III Relationship Between Estimated Measurement Uncertainty, Rounding Ratio, and Inflation

U_m	0.5	1	2	3	5
Rounding ratio	1	2	4	6	10
% inflation	15.47	4.08	1.04	0.46	0.17

Table IV shows the relationship between uncertainty code and estimated total uncertainty (U_T) with our current rounding convention (± 0.5 to ± 5).

Table IV Relationship Between Uncertainty Code and Estimated Total Uncertainty with the ± 0.5 to ± 5 Rounding Convention.

Uncertainty code										
0	1	2	3	4	5	6	7	8	9	
.58	.69	.84	1	1.3	1.6	2	2.5	3.2	4	5
Estimated total uncertainty										

Now compare Tables I and III. Note the difference between U_m and U_T in the lower uncertainty codes. Our task would be simplified greatly if this discrepancy were minimized. We could accomplish this by changing the rounding convention.

For purposes of Linear Code, we will adopt the convention that the estimated measurement uncertainty (U_m) on the last significant digit will be between ± 2 and ± 20 , in order to have a reasonably small inflation. The minimum rounding ratio allowed by this rounding convention is calculated as follows:

$$\text{min. rounding ratio} = 2/0.5 = 4$$

According to Table III, this corresponds to an inflation of 1.04%, which is tolerable, even by Swiss standards. Thus Table II is merely a trial balloon for a widely-used rounding convention. For the actual application of linear coding (with the ± 2 to ± 20 convention), Table V must be used.

Table V Relationship Between Uncertainty Code and Estimated Measurement Uncertainty for the ± 2 to ± 20 Rounding Convention.

Uncertainty code										
0	1	2	3	4	5	6	7	8	9	
2	2.5	3.2	4	5	6.3	8	10	12.6	15.9	20
Estimated measurement uncertainty										

If we mistake a Linear Coded quantity for an ordinary base 10 number, then we lose out on the uncertainty. But the value of the measurement does not suffer appreciably from the misinterpretation. In the worst case scenario, the misinterpretation error on the last reported digit is 9, as compared with the measurement uncertainty on the last reported digit which is at least 15.9. Therefore the maximum misinterpretation error is

$$9/15.9 \approx 5.7\%$$

of the uncertainty. So Linear Code is very forgiving to the reader of a data table, who has never heard of Linear Code, and mistakes it for ordinary base 10 notation.

Economy is the primary rationale for using any kind of data compaction technique (or even stand-alone significant figures) in lieu of directly reported uncertainties. Therefore it would be prudent to calculate the cost of Linear Code relative to the ordinary (stand-alone) usage of significant figures.

There are two cost components, the first of which is the one extra digit for the uncertainty code. Therefore Cost 1 = 1 digit. The second component is equal to the probability that a quantity, which has been tentatively rounded in accordance with the ± 0.5 to ± 5 convention, will need an extra digit in order to satisfy the new ± 2 to ± 20 convention. This may be calculated as follows:

$$\text{Cost 2} = \log(2/0.5) = 0.602 \text{ digits}$$

Now we will calculate the total relative cost of linear coding.

$$\text{Total cost} = \text{Cost 1} + \text{Cost 2} = 1 + 0.602 = 1.602 \text{ digits}$$

Thus Linear Code is more expensive than stand-alone significant figures. However Linear Code gives much more information about the estimated measurement uncertainty of an experimental parameter. Moreover Linear Code is a very understandable method. Eventually we will see that it is mainly useful as a reference technique. In the next section, we will introduce a data compaction method, which is more efficient than linear code. We will see that this technique also has convenient human engineering features.

DECIMAL CODED BINARIES

We will return once again to our linguist parable. We saw that the essence of "Listen to me!" is found in the altered tone of voice, rather than the words themselves. When we combined the meanings of the actual street directions and this auxiliary sentence in a compact form, all we did was to add an upscale glissando to the end of the otherwise monotonic "Then-you-turn-left" in order to create the desired effect (lifting the other person out of his or her lethargy). In this example, the key to efficient data compaction is the ending of the sentence.

As we mentioned earlier, there is a striking parallel between the linguistic parable and our data compression techniques. In the Linear Coding technique of the previous section, we focused on the base 10 endings of numbers, which is a reasonable approach. However a number can have more than one kind of ending.

The fact that we usually write in base 10 does not prevent a base 10 number from having a base n ending. For example, if we divide the base 10 number 249 by 10, the remainder is 9 (the base 10 ending). However this fact does not prevent us from considering the base 7 ending (remainder), which is 4.

In this section, we will explore a more efficient data compression technique, **DCB**, or **Decimal Coded Binaries** (which are not the same as the binary coded decimals of computer science). As the name suggests, DCB hybridizes the base 10 and base 2 number systems. We will see that DCB quantities can be thought of as base 10 numbers with base 2 endings.

In the ordinary significant figure theory of the binary number system, the minimum estimated measurement uncertainty (U_m) allowed on the last reported digit is ± 0.5 , as in base 10, but the maximum U_m on the last reported digit is ± 1 . However we will use a ± 2 to ± 4 rounding convention in order to guarantee that the rounding ratio is always at least 4, thereby minimizing the rounding error. (The minimum rounding ratio for all of our data compression techniques is 4.) Therefore after we round off a binary number (using the ± 2 to ± 4 convention), it will have uncertainty on the last three significant binary digits.

In some ways, binary significant figures are nicer than base 10 significant figures. In binary, the estimated measurement uncertainty on the last reported digit is limited to a $\pm 4/\pm 2 = 2$ -fold range, as compared with a ten-fold range for base 10. The major disadvantage of binary numbers is that they require too much space to write out. For example, "512" in base 10 is "1 000 000 000" in binary. However it is possible to combine the virtues of the base 10 and binary number systems, and have the best of both worlds.

Basically, the DCB algorithm converts an **integerized*** base 10 quantity into binary, rounds to the correct number of binary digits, and then converts back to base 10. We can represent this concept as "2-R-10" in symbolic shorthand. However there is one difficulty with this oversimplified picture, which we will illustrate in the next paragraph.

Suppose that the value of a measurement is 302 mg, and that only two significant figures can be justified, which means that we must round to 300 mg. How is a third party to know which, if any of the

*For now, our explanation of **integerization** will be slightly oversimplified. The number 0.0345 is 3.45×10^{-2} in scientific notation. Remember that we are calling the "3.45" the **mantissa**. The **integerized** mantissa is 345, and the **integerized** form of the original number is 345×10^{-4} .

terminal zeros in "300" are significant figures? This is the **terminal zero problem**.

There are two ordinary ways to deal with this situation. First we could express the quantity as " 3.0×10^2 mg", using scientific notation. Alternatively we could write "0.30 g". Neither approach will solve the terminal zero problem for DCB. Instead we will use a binary **marker** as an uncertainty location bit. The full-fledged algorithm for converting base 10 data to DCB notation is as follows:

- 1) Integerization: Let **M** and **Um** represent the absolute value of the mean of a series of measurements, and its standard error, respectively.
 - (a) If **Um** or **M** is less than 4, then multiply both **M** and **Um** by 10. Repeat as needed, until the **Um** becomes ≥ 4 .
 - (b) Set a counter variable **J** (which is initially zero) equal to the number of times (a) is repeated.
 - (c) and (d) These steps (to be given later) are used only with a DCB variation called **Scientific DCB**.
- 2) Convert the integerized (base 10) **M** to binary.
- 3) Round the result of Step 2 such that the adjusted **Um** on the last significant binary digit is between 2 and 4. Be sure to remember the position of the last significant binary digit!

The fifth step will be to convert back to base 10, but there is one hurdle to clear first. Suppose that the result of Step 2 is 11011000 in binary (216 in base 10). How is a third party to know which binary digit is the last significant figure? It could be the right-most "1" or it could be any of the three final zeros. We have come back to the **terminal zero problem**, which we will solve by flagging the location of the final significant binary digit.

- 4) Flagging. Put a "1" immediately after the last significant binary digit, and keep it there.

This means that the right-most "1" has absolutely nothing to do with the actual value of the number, and is merely a **marker**, which says that there is uncertainty on the next three binary digits to the left.

- 5) Convert the result of Step 4 to base 10. (This will always be an integer.) The marker is now a part of the new base 10 number.
- 6) Bookkeeping. Take the result of Step 5 and move the decimal point J places to the left (or -J places to the right). Do not discard or add any terminal zeros when moving the decimal point. If necessary, express in a scientific-like notation. If the measurement's sign is negative, then remember to put a (-) sign in front of the DCB quantity.

This completes the procedure for translating the results from a set of scientific measurements (i.e. the mean value and its uncertainty) into DCB notation. We may summarize the complete DCB coding procedure as

"Int-2-R-F-10-B". A listing of the programs for coding and decoding DCB on the HP15C calculator may be obtained by writing to the authors.

There is one important boundary condition for DCB. This notation can only be used to represent a quantity, whose standard error of the mean is less than the absolute value of the mean. Now we will give some examples.

Our first example will serve two purposes. First we will get some practice in decoding DCB (i.e. translating DCB into prose). Secondly we will gain some insight into the importance of terminal zeros.

Example 1.

Let $A = 2.3$ and $B = 2.30$, both in DCB notation. Is A equivalent to B?

Working backwards, we will translate A into prose.

Obviously $J = 1$, and the Step 5 result is 23 (in base 10).

The Step 4 result is 10111 (in binary).

The marker in "10111" is in the ones' place. This means two things.

- (a) There is uncertainty in the twos', fours', and eights' places; i.e. that the adjusted U_m is between ± 4 and ± 8 .
- (b) The Step 3 result (rounded binary quantity without marker) is 10110.

We do not know the value of the Step 2 result (except that it is between binary 10101.1 and binary 10110.1), because a little information was lost in rounding. So we will use the rounded value, 10110.

The Step 1 result is 22 in base 10.

Since $J = 1$, we divide both "22" and the adjusted U_m by 10^1 in order to arrive at the prose equivalent of A, which is "2.2, with an estimated measurement uncertainty between 0.4 and 0.8". (Note that "between 0.4 and 0.8" is not a confidence interval.)

Similarly, it can be shown that B has a prose equivalent of "2.28, with an estimated measurement uncertainty between 0.08 and 0.16".

Therefore A is not equivalent to B. This is why we must be so careful with terminal zeros when we are using DCB.

If one is doing arithmetic on DCB numbers, then each quantity should be decomposed into its actual value and uncertainty. Thus there are four distinct operations to perform:

- 1) the decomposition (decoding)
- 2) the calculation itself
- 3) a propagation-of-uncertainty analysis
- 4) combining (coding) the results of (2) and (3) into a single DCB quantity.

Since a DCB quantity looks like an ordinary base 10 number, it is possible to confuse the two. If one is constructing a table of scientific data, using DCB notation, then it would be best to state this explicitly at the outset.

If the person reading such a table is not familiar with DCB, and mistakes a DCB quantity for an ordinary number, then he or she loses out on the information about the uncertainty. However the value of the measurement does not suffer much, as we will see in the next example.

Example 2.

Suppose that the mean value of a series of measurements is 35.20 mV, and that the systematic error is assumed to be negligible in comparison with the standard error of the mean, which is 0.71 mV.

- 1a) $U_m = 0.71 < 4$. Therefore the adjusted mean is 352, and the adjusted U_m is 7.1.
- 1b) $J = 1$
- 2) The (base 10) number 352, expressed in binary is 101100000.
- 3) This is rounded to the correct number of binary digits already.
- 4) The last significant binary digit is the second "0" from the right, in the twos' place. Therefore we will replace the right-most "0" with a "1" (the marker). The result is 101100001.
- 5) Converting the result of (4) back to base 10 gives 353.
- 6) Moving the decimal point $J (=1)$ places to the left yields 35.3 mV, in DCB notation.

Comparing this result with the original base 10 quantity, we see that the two numbers (35.2 and 35.3) are quite close. In general, their difference will be no greater than $U_m/4$; i.e. the penalty is small when one mistakes DCB for prosaic notation. Thus DCB is very forgiving (but not as forgiving as Linear Code).

For many data tables—such as reduction potentials—where most of the values are in the same ballpark, DCB notation will be adequate. However it would be rather cumbersome to use DCB per se in tables of equilibrium constants. For data tables, which would ordinarily require scientific notation, there is an alternative, **Scientific DCB (SDCB)**. This is very similar to DCB, but is a scientific-like notation. In Scientific DCB, the following additions are made to the DCB procedure.

Scientific DCB Addenda:

- 1c) If $U_m \geq 40$ then divide the base 10 quantity (and U_m) by 10.
Repeat as needed.
- 1d) Let $J = -1$ multiplied by the number of times 1c is repeated (if 1a does not apply).

In the next several paragraphs, we will discuss the comparative economics of Scientific DCB. We are considering Scientific DCB rather than DCB, because DCB, like **unscientific** notation, can be rather wasteful. (Imagine having to write the Avogadro constant that way!) Going back to Example 2, the original base 10 quantity in ordinary scientific notation, used in conjunction with ordinary stand-alone

significant figures, would be expressed as 3.5×10^1 mV, which requires one less digit than Scientific DCB notation (3.53×10^1 mV).

Scientific DCB and scientific notation (used in conjunction with ordinary stand-alone significant figures) will have the same digital cost only when U_m on the last significant base 10 digit is between 4 and 5. The probability of this being the case is

$$\log(5/4) = 0.09691$$

Therefore the absolute cost of Scientific DCB is

$$1 - 0.09691 = 0.90309 \text{ digits}$$

Obviously Scientific DCB is cheaper than Linear Code, whose cost is 1.60206 digits, but linear code gives more information about U_m . In linear code, U_m can be pinned down within a $10^{0.1} = 1.26$ -fold range, as compared with a two-fold range at worst for Scientific DCB. (If the binary marker of an SDCB quantity is in the fours' place, then the estimated uncertainty is pinned down within a $40/32 = 1.25$ -fold range.)

However we can construct **Extended DCB** (a modification of Scientific DCB), which will allow us to make a fair comparison between the two methods of data compression. In order to demonstrate that SDCB is a more efficient technique than Linear Code, we will show that Extended DCB gives more information about the estimated uncertainty, and is cheaper than Linear Code. Since complete mastery of Extended DCB is not necessary for the reader who wants a general understanding of DCB, we will give only a brief outline of Extended DCB.

Modifications to SDCB Procedure for Extended DCB.

- A. Change the magic numbers in Step 1 of the SDCB procedure from 4 and 40 to 16 and 160.
- B. Reserve the last two binary digits in the Step 4 result for a quaternary uncertainty code, analogous to the base 10 uncertainty code used for linear code in the previous section. The result is an SDCB-Linear Code hybrid, Extended DCB.

Extended DCB will allow us to pin down U_m within a $2^{1/4} = 1.19$ -fold range at worst, an improvement over linear code. The additional cost of Extended DCB (over Scientific DCB) is

$$\log 4 = 0.60206 \text{ digits}$$

Thus the total cost of Extended DCB is

$$0.90309 + 0.60206 = 1.50515 \text{ digits}$$

which is cheaper than Linear Code. We have seen that when Scientific DCB is linearized, the resulting code is more economical than Linear Code alone. It would be reasonable to infer that it was the Scientific DCB component which contributed the increase in cost-effectiveness. Therefore Scientific DCB is inherently more economical than Linear Code.

Thus DCB is an efficient and forgiving technique. The essence of DCB is to combine the compactness of base 10 with the uncertainty specificity of binary. A mnemonic for the DCB algorithm is "Int-2-R-F-10-B".

In a later section, we will see that the DCB variations have an additional virtue, flexibility. We will argue that DCB is a better overall technique than Linear Code.

EXPERIMENTAL SECTION

In an attempt to integrate the theory of decimal coded binaries with the real world, the DCB algorithm was programmed into an AIM 65 microcomputer, which was interfaced with an Orion 701A Ionalyzer (pH meter), via its digital output edge connector. The program, which was written in (interpreted) BASIC, included a machine language subroutine for reproducibly recognizing when practical equilibrium would have been attained, had the apparatus been used with an actual sample solution.

Measurements of pH were simulated by means of the calibration knob on the Orion. The uncertainty for a given simulated measurement was manually entered from the keyboard of the AIM. It would have been just as easy (albeit much more time-consuming, because of the practical equilibrium subroutine) to arrange for the program to calculate the standard error of the mean from a series of measurements. After taking the data, the program expressed the result in a DCB format.

Table VI shows the details of the hardware interfacing.

Table VI Connections Between the
AIM 65 Microcomputer and
the Orion 701A Ionalyzer

AIM 65	ORION 701A
PB0	.1
PB1	.2
PB2	.4
PB3	.8
PB4	1
PB5	2
PB6	4
PB7	8
PA0	10
PA1	20
PA2	40
PA3	80
PA4	100
PA5	200
PA6	400
PA7	800
CA1	DATA READY
CA2	HOLD
GROUND	GROUND

RESULTS AND DISCUSSION

The entire program was well within the AIM's 4K of memory. About a minute was required for the recognition of practical equilibrium. After that, the program took only a few seconds to combine the simulated mean and simulated standard error of the mean into a DCB number (in an integerized scientific-like notation), which could be viewed in the AIM's LED display. This experiment demonstrates that the computer interfacing requirements of DCB do not pose a major problem in the implementation of this data compaction technique. Even if the innards of a given chemical instrument are not readily accessible, DCB coding and translating routines can readily be programmed into an HP15C calculator.

SUMMARY AND CONCLUSION

We have introduced two classes of data compression techniques: Linear Code and Decimal Code Binaries (DCB, Scientific DCB, and Extended DCB). Either method can provide a readable and economical format for reporting the two essential components of a scientific measurement—the measurement itself and an index of its trustworthiness.

Now we will compare the relative merits of linear code and DCB. These two methods will be evaluated in terms of the following criteria:

- 1) Efficiency
- 2) Flexibility
- 3) Forgiveness
- 4) Simplicity

Using a linguistic parable, we compared linear code with the data compaction techniques used in an ordinary telegram, while DCB was compared with the use of vocal inflection, the latter approach being somewhat more efficient.

The next example will illustrate the **flexibility** of DCB.

Example 3. Experimental mean = 123456 bacteria/L. The standard error of the mean (U_m) is 789 bacteria/L.

Linear code	1.2355X10 ⁵	bacteria/L
DCB	123520	bacteria/L
Scientific DCB	1.235X10 ⁵	bacteria/L

Example 3 shows that it is sometimes necessary to use a scientific-like notation with linear code. This is because of the terminal zero problem, which we discussed in an earlier section. In principle, it will always be possible to express DCB quantities in ordinary unscientific notation, which is highly readable in data tables. However we have the option of using Scientific DCB if we want to save space when constructing tables of widely spaced quantities, like solubility products.

Both linear code and DCB are forgiving, in the sense that the reader of scientific data tables does not suffer much if he or she has never even heard of these coding techniques, since the value of the measurement-based quantity is distorted very little in the coding process. Of course the information about the measurement's uncertainty is available only to those who know that it is there! We have already pointed out that DCB is more forgiving than linear code.

Conceptual simplicity is the least important of the four criteria. Since coding and decoding procedures for DCB can be programmed into an HP15C calculator, the user of the technique need not worry about the fine details. Since linear code is merely doing the obvious, it is more understandable than DCB, although DCB programs tend to be short (about 100 bytes for both coding and decoding in an HP15C calculator). Table VII graphically summarizes this comparison.

Table VII A Comparison of DCB with Linear Code.

	Efficient	Flexible	Forgiving	Simple
Linear code	*	*	**	**
DCB variations	***	***	*	*

Table VII shows that DCB is a better method in most respects. Linear Code serves as a learning tool, as an auxilliary technique, and as a Brand X or reference method, which highlights the attractive features of the more sophisticated technique, DCB. In the next several paragraphs, we will discuss the potential role of DCB in science and technology.

At present, there is a double standard among scientists about the reporting of measurement data. On the one hand, we pay lip service to the idea that we should always include an estimate of the uncertainty, when we report a measurement. Yet we tend to forget this when we publish our data in tabular form—and for good reason. A data table would be rather cluttered if we included a detailed error analysis for each datum.

In some circumstances, the present practice is tolerable. If a chromatographer is reading a table of GC retention indices, then he or she will probably have some feeling for the limitations of the measurements, even if the uncertainties are not stated explicitly.

However an organic chemist may be using these tables to assist in the identification of a newly synthesized compound. He or she will want to compare his or her retention index (for a given stationary phase) with the literature value. The question is: "How close is close enough?" This will not be a problem if the literature value is reported in a DCB format.

When we put our data into tables, DCB will allow us to practice what we preach about the reporting of uncertainty. DCB is the middle ground between stand-alone significant figures and the prosaic reporting of measurement data. It is very efficient at communicating the essentials of measurement data. Thus far, we have emphasized the

saving of ink. However the potential uses of DCB extend far beyond the printed page.

In the previous section, we argued that the programming and computer interfacing requirements of DCB are not prohibitive. DCB could improve the efficiency of scientific data storage in computer memory (or on a hard disk or on a floppy). Moreover there is no reason to preclude the use of standard data compression techniques in tandem with DCB. If DCB promotes more efficient use of computer memory, then it should also increase the speed of electronic communication. Let's take an example.

Suppose that NASA sends a probe to Titan (a satellite of Saturn), and that there are 100 experiments running simultaneously inside the landing module. It would be important to transmit the data to Earth as rapidly as possible. DCB (or even Hexadecimal Coded Binaries) could be an important part of the overall plan to optimize the communication rate, by combining in one signal both the datum and the value of the measured noise.

Of course the potential applications of DCB are not limited to the exploration of outer space. The usefulness of most chemical instruments with digital readout could be enhanced if DCB was a built-in feature.

In large projects, technicians are often sent out into the field with portable chemical instruments. The logging time could be reduced if the technician wrote a single DCB number into his or her notebook, rather than having to write both a mean and its standard error.

In this paper we have presented two new techniques for compressing scientific data—DCB and linear code. The best method, DCB, does not have a very high level of specificity for the uncertainty. If more information about uncertainty is required for a given application, then a tailor-made technique could easily be constructed, by hybridizing DCB and linear code, as we did with Extended DCB in an earlier section. Therefore DCB is really an array of techniques.

To summarize, DCB should increase the effectiveness of automated communication—written and electronic—among scientists, especially across disciplinary boundaries.

APPENDIX

Both classes of data compaction techniques discussed in this paper combine a measured quantity and its uncertainty within a single number. In contrast with ordinary stand-alone significant figures, these techniques have a high level of specificity for the uncertainty. It would be of interest to be able to do a simple propagation-of-uncertainty analysis on a calculation involving quantities which have been coded by either method.

For example, in the second section we introduce the **Linear Coding** technique. If two Linear-Coded quantities are multiplied together, it would be nice to be able to express the product in Linear Code also. This will be possible only if we have chosen a definition of uncertainty, which permits straightforward propagation-of-uncertainty analysis.

We will argue that propagation of uncertainty is quite simple for the absolute limit of error and standard error of the mean definitions of uncertainty, but that there are serious problems with the confidence interval definition, because the sample size must be taken into account.

If uncertainty is defined to be the absolute limit of error, then propagation of uncertainty can be handled by the following equation:

$$RU = \Delta f / f = [\Sigma(df/dx_i)\Delta x_i] / f + RUM \quad (A.1)$$

where **RU** is the relative uncertainty of **f** (a function of the x_i), Δf is the absolute value of the uncertainty of **f**, Δx_i is the absolute value of the uncertainty of x_i , and **RUM** is the relative uncertainty of the calculation method. Sometimes the functional relationship **f** is only approximate and is itself a source of error. (For example Beer's Law neglects the effect of stray light.) It is also assumed that **f** is well behaved, and that each Δx_i is small relative to x_i .

If uncertainty is defined to be some multiple of the sample standard error of the mean, then we may use the following equation:

$$U = A\Delta f = A\{\Sigma[(df/dx_i)\Delta x_i]^2\}^{1/2} \quad (A.2)$$

where U is the uncertainty, Δf is the absolute value of the estimated standard deviation of f , each x_i is the mean value of a series of measurements (of the same parameter), Δx_i is the absolute value of the standard error of the mean of x_i , and A is a positive real number. Thus propagation of uncertainty is very simple for both the absolute limit of error and standard error of the mean definitions of uncertainty, because Equations A.1 and A.2 are independent of sample sizes and of the component frequency functions (i.e. it doesn't matter whether they are Gaussian, Poisson, rectangular, etc.).

If uncertainty is defined to be half the span of a 95% confidence interval, then the propagation of uncertainty is dependent on the sample size and frequency function associated with each of the x_i . The implication is that in this case, there is no simple formula for propagation of uncertainty, which will work for all cases. To prove this rigorously would be beyond the scope of our present discussion. However in the next several paragraphs we will explore the obvious approaches. All of these blind alleys will become "the exceptions which prove the rule". Let's suppose that

- (a) the data points for each of the x_i are normally distributed, and
- (b) the sample size is known for each of the x_i .

Then we may use a t Table to help us calculate each of the standard errors of the mean (Δx_i), which we plug into Equation A.2 in order to find Δf . But then how do we calculate the 95% confidence interval for f ? There are two obvious possibilities.

If the $\Delta x_i(df/dx_i)$ are all equal, and if the sample sizes (of the calculation's components) are all equal, then the calculation of the confidence interval will be a radial error problem, which can be solved using a chi-square technique¹³.

On the other hand, the t Table can be used in this final step only if we can make the simplifying assumption that essentially all of the uncertainty is contributed by a single **least precise factor, x** . However in this special case, there is no need for a t Table, or for assumptions (a) or (b). Instead we may use the following equation:

$$\Delta f = \Delta x(df/dx) \quad (A.3)$$

with Δx and Δf representing the half-spans of the 95% confidence intervals of x and f , respectively.

An even bigger problem is that the sample sizes will not always be available when we are using other people's data in our calculation.

There is a simple way to obtain a confidence interval for the result of a calculation, but it will not be the 95% confidence interval. In the following artificial example we will suppose that the measured base and height of a rectangle are 987 cm and 654 cm, respectively, that the 95% confidence interval of each quantity is ± 1 cm, and that no information is available about sample size. The estimated area is

$$987 * 654 = 645498 \text{ cm}^2$$

From the definition of the 95% confidence interval, the compound probability that the base is greater than $(987+1)$ cm, and that the height is greater than $(654+1)$ cm is

$$[(1 - 0.95)/2]^2 = 6.25 \times 10^{-4}$$

which is also equal to the probability that the 95% confidence intervals fail to bracket both base and height at the low ends. Therefore the probability P that the true mean area (i.e. the calculated area, based upon the true mean values of the base and height measurements) is not bracketed by the interval

$$[(654 - 1)(987 - 1), (654 + 1)(987 + 1)] = [643858, 647140]$$

may be calculated as follows:

$$P = 2(6.25 \times 10^{-4}) = 1.25 \times 10^{-3}$$

Thus the above interval is the

$$100 (1 - 1.25 \times 10^{-3}) = 99.875\%$$

confidence interval for the true mean area of the rectangle. However we were seeking the 95% confidence interval.

Again, we have failed to find a general method for obtaining the 95% confidence interval for f , from the 95% confidence intervals of the x_i . If such a method does exist, then it is not obvious. It would be of theoretical interest to find such a method, or to prove its nonexistence. However the main point of this discussion is that propagation of uncertainty in the general case is very simple for the standard error of the mean and the absolute limit of error definitions of uncertainty, but it is not straightforward for the confidence interval definition.

DCB PROGRAMS FOR THE HP15C SCIENTIFIC CALCULATOR

Since HP15C programming is rather primitive, there are some limitations, which we must contend with. There are no character variables, nor is there any string manipulation. In particular, the calculator cannot recognize a terminal zero in a decimal fraction; i.e. it cannot distinguish between "1.2" and "1.20". Therefore at times the directions will tell you to ignore decimal points, to pretend that real numbers are integers. At other times, the instructions will tell you how to use your common sense in order to put the decimal point in its proper place. Since the simplest concepts are often the most difficult to explain, the directions may seem more complicated than they really are.

Program for Expressing Measurement Data in a DCB Format

This first program transforms experimental data into DCB notation. We will give instructions for using the program before listing the program itself. Suppose that you are taking a series of measurements of the same thing (eg. the spectrophotometric absorbance of a solution).

- A. First key in "f Σ " to clear the statistical registers of the calculator.
- B. Enter the measurements or their mantissas* into the calculator, as you would enter ordinary statistical data using the $\Sigma+$ key after each entry.

Most of the time you will have the option of entering the measurements themselves, rather than their mantissas. However if the numbers are very large (in the neighborhood of 10^9), then you may get a spurious result. If you use mantissas (the safest approach), then you will also get the maximum compression of your data.

*For example, the **mantissa** of the Avogadro constant is "6.022".

Common sense. If you are using the measurement mantissas rather than the measurements themselves, then all of the measurements, when expressed in scientific notation, should have the same exponent. If they do not, then it will be necessary for you to force them all into the same mold.

The program, which has a bad case of integer obsession, will sometimes ignore the decimal points in nonintegers. Therefore you may need to do some scaling up or scaling down after the program has finished running.

- C. When all of the measurement data has been entered, then key in "f C". "C" (which stands for "Coding") is the name of the program.
- D. After about 7 seconds, the integerized DCB mantissa (the DCB mantissa without the decimal, which is understood to be between the first and second digits) will be displayed.
- E. The final DCB quantity is the displayed mantissa, with a decimal point inserted between the first and second digits, followed by the exponential part (the $\times 10^?$).

Do not add or discard any terminal zeros! For example, if the number in the display is "180", then the DCB mantissa may be written as "1.80". Do not write "1.8". The DCB quantities "1.80" and "1.8" have completely different meanings.

- F. Write the complete DCB quantity into your lab notebook. If this can be done without adding any terminal zeros, then you have the option of using an ordinary unscientific DCB notation. However a scientific-like notation, which includes both a DCB mantissa and an exponential part, is always correct.
- G. If you are constructing a data table using DCB notation, then you should state this explicitly in the heading of the table. Then your readers will not confuse it with ordinary notation, and they will be able to take full advantage of the information about the measurement's uncertainty embedded within the DCB quantity.

The algorithm has two boundary conditions. If the absolute value of mean is much smaller than its standard error, then it will not be possible to express the result in DCB notation. In particular, the

program will not handle a zero mean. Secondly, it is not possible to express an exact number (having zero uncertainty) in DCB notation.

Example. Suppose that the values of three voltage measurements were 0.496 mV, 0.508 mV, and 0.520 mV. These numbers can be entered into the calculator

- (a) without modification
- (b) or as 4.96, 5.08, and 5.20 respectively, with $\times 10^{-1}$ understood
- (c) or as 49.6, 50.8, and 50.2 respectively, with $\times 10^{-2}$ understood
- (d) or as 496, 508, and 520 respectively, with $\times 10^{-3}$ understood.

In each case, the integerized DCB mantissa displayed by the calculator will be "510" (which is understood to mean 5.10×10^{-1}). The DCB quantity, which you put into your lab notebook can be written in any of the following ways:

- (a) 0.510 mV
- (b) 5.10×10^{-1} mV
- (c) 51.0×10^{-2} mV
- (d) 510×10^{-3} mV

There is one main pitfall: Do not discard the terminal zero!
The following is the annotated program:

	X register	Y register	Z register
Preliminaries			
1 f LBL C			
2 f FIX 0			
3 g \bar{x}	\bar{x}		
Handles the boundary condition where \bar{x} might be < 0 .			
4 ENTER	\bar{x}	\bar{x}	
5 g ABS	$ \bar{x} $	\bar{x}	
6 STO 1			
7 \div	Sign (\bar{x})		
8 STO 8			
Formatting \bar{x} and standard error of the mean.			
9 g s	s		
10 RCL 2	N	s	

11	\sqrt{x}	\sqrt{N}	s
12	\div	Std. error	
13	STO 0		
14	4	4	Std. error
15	$x \neq y$	Std. error	4
16	g TEST 9 ($x \geq y$)		
17	GTO 0		
18	\div	4/std. error	
19	g LOG	log	
20	1	1	log
21	+	1 + log	
22	g INT	exponent	
23	10^x	factor	
24	STO X 1		
25	RCL X 0	std. error	
Finding Range Index			
26	f LBL 0		
27	g LOG	log std. error	
28	2	2	log std. error
29	g LOG	log 2	log std. error
30	\div	base 2 log of std. error	
31	g INT	Range Index	
Rounding			
32	1	1	RI
33	-	Adjusted RI	
34	2	2	ARI
35	$x \neq y$	ARI	2
36	Y^X	AR	
37	STO I		
38	RCL I	\bar{x}	AR
39	$x \neq y$	AR	\bar{x}
40	\div	Shrunk \bar{x}	
41	.		
42	5	.5	Shrunk \bar{x}
43	+	Intermed	
44	g INT	RS \bar{x}	
45	RCL I	AR	RS \bar{x}
46	X	Rounded \bar{x}	

Binary Marker

47 RCL I	AR	Rounded \bar{x}	
48 2	2	AR	Rounded \bar{x}
49 ÷	Marker	Rounded \bar{x}	
50 +	DCB		
51 RCL 8	Sign (\bar{x})	DCB	
52 X	Voila!		
53 g RTN			

HP15C Program for Decoding DCB.

This second program translates DCB into plain English. Again, we will give instructions on the use of the program before we actually list it.

- A. Key in the integerized DCB mantissa (i.e. the mantissa with the decimal point omitted, but understood to be between the first and second digits).
- B. Key in "f D". The "D" is a mnemonic for "Decode", and is the name of the program.
- C. The integerized mantissa of the mean will appear in the display (X register). The Y register will contain the estimated relative uncertainty (estimated standard error of the mean divided by the sample mean).

The resolution of DCB is such that the estimated relative uncertainty in the Y register will not differ from the calculated value of the relative uncertainty by more than a factor of $\sqrt{2} \approx 1.414$ (not to be confused with a confidence interval). If you want to view more than one significant figure of the estimated relative uncertainty, then you will need to key in "f PREFIX" (or fiddle with "f FIX" or "f SCI").

The following is the actual program listing:

```
1 f LBL D
2 g x = 0
```

Avoiding an

3	GTO 2	infinite loop.
4	f CLEAR REG	Initialization.
5	ENTER	Handles any
6	g ABS	DCB quantity
7	÷	even if
8	STO 8	it is less
9	g LST X	than zero.
10	STO 0	
11	f LBL 1	
12	2	Is the
13	÷	number
14	f FRAC	divisible
15	g TEST 0	by 2?
16	GTO 2	If no, branch down.
17	1	Otherwise
18	STO + 1	increment counter
19	g LSTX	and
20	GTO 1	loop back.
21	f LBL 2	
22	2	No. of clean
23	RCL 1	divisions by 2 as
24	YX	exponent of 2.
25	STO 1	
26	CHS	Subtract out the
27	RCL + 0	binary marker.
28	RCL 1	Estimation
29	3	of the
30	2	standard
31	1/x	error
32	\sqrt{x}	of the
33	÷	mean.
34	x \rightleftharpoons y	mean in X register; estimated standard error in Y reg.
35	÷	Estimated relative uncertainty (ERU)
36	g LST X	mean in X register; ERU in Y register
37	RCL 8	
38	X	Mean in X register; estimated standard error in Y reg.
39	f FIX 0	
40	g RTN	

EIGHT CODE

THEORY

Eight Code is the most complicated and most efficient of the data compaction methods in this study, although it is less forgiving and less flexible than DCB. Before giving the Eight Code algorithm, we will discuss the general principles of this technique. Even these may seem bewildering at first. However the BASIC coding program occupies about 1K of memory in an eight bit microcomputer (and about 120 bytes in an HP15C scientific calculator). Therefore it is possible to make practical use of the technique, if one has a good understanding of its purpose.

Eight Code relies heavily on the **modulo** concept. "Q modulo N" means: "Divide the quantity Q by N, and take the remainder." As the name suggests, Eight Code uses "modulo 8".

In Eight Code, **mantissa** is also an important concept. The mantissa of the Avogadro constant ($6.022 \times 10^{23} \text{ mol}^{-1}$) is "6.022". At times, we will find it convenient to think of mantissas as being integers. For example, the integerized mantissa of the Avogadro constant is "6022".

As with Linear Code, and especially with DCB, the numerical value of the coded quantity will be fairly close to that of the actual experimental result. However no information about the estimated measurement uncertainty U_m is available unless one knows the coding method (or has programmed it into his or her scientific calculator). In order to obtain information about U_m from Eight Code, one must look at the coded integerized mantissa, modulo 8, which we will call the **mod value**.

In Linear Code, the final base 10 digit is the uncertainty code. Similarly in DCB, the final binary "1" serves as the marker, or uncertainty location bit. Both the uncertainty code (in Linear Code) and the marker (in DCB) give information only about the estimated measurement uncertainty (U_m). The efficiency and complexity of Eight Code stem from the fact that the mod value has a two-fold purpose; it gives information about U_m and about the measured quantity.

There are eight integral mod values, corresponding to four categories of uncertainty (on the last digit of the coded integerized mantissa). **Class 1**, the category of highest uncertainty, corresponds to a mod value of zero. **Class 2**, the second-highest uncertainty category, corresponds to a mod value of 1. Mod values of two and three correspond to **Class 3**, the third-highest uncertainty category. **Class 4**, the category of lowest uncertainty, corresponds to mod values 4 through 7.

Notice that it is possible to represent Class 4 (or Class 3) by more than one mod value. The apparent redundancy of Eight Code allows the measured quantity itself to have two extra binary digits for Class 4 uncertainties, and one extra for Class 3 uncertainties. The rationale for doing this is the following general principle:

The smaller the relative uncertainty, the larger the number of significant figures.

The Eight Code procedure is as follows:

- 1) Preliminary integerization. (This is exactly the same as Step 1 in the Scientific DCB procedure given in a previous section.) Let **M** and **Um** represent the absolute value of the mean of a series of measurements (of the same parameter), and its estimated measurement uncertainty (standard error of the mean), respectively.
 - (a) If Um of M is less than 4, then multiply both M and Um by 10. Repeat as needed, until the Um becomes ≥ 4 .
 - (b) Set a counter variable J (which is initially zero) equal to the number of times (a) is repeated.
 - (c) If $Um \geq 40$ then divide the base 10 quantity (and Um) by 10. Repeat as needed, until the Um becomes < 40 .
 - (d) Let $J = -1$ multiplied by the number of times (c) is repeated if (a) does not apply.
- 2) Let $AM = 10^J$ times M . Round AM to the nearest multiple of 8. We will call this result **ARM** (the Adjusted and Rounded Mantissa).
- 3) If adjusted $Um \geq 8$, then go to Step 4. Steps 3 to 5 do the actual coding.
 - (a) If $2 \leq AM - ARM$ then add 7 to ARM .
 - (b) If $0 \leq AM - ARM < 2$ then add 6 to ARM .
 - (c) If $-2 \leq AM - ARM < 0$ then add 5 to ARM .
 - (d) If $AM - ARM < -2$ then add 4 to ARM .

- 4) If adjusted $U_m \geq 16$, then go to Step 5.
 - (a) If $0 \leq AM - ARM$ then add 3 to ARM.
 - (b) If $AM - ARM < 0$ then add 2 to ARM.
- 5) If $16 \leq U_m < 25.3$ then add 1 to ARM.
- 6) Bookkeeping. Take ARM and move the decimal point J places to the left (or -J places to the right). Do not discard or add any terminal zeros when moving the decimal point. If necessary, express in a scientific-like notation. (This step is essentially the same as in the DCB variations.)

In the next example, we will work through the Eight Code algorithm, starting with the mean and standard error of the mean for a series of hypothetical chemical measurements. We will also see that Eight Code is not as forgiving as DCB.

Example. Mean = $M = 11.3$; $U_m = 0.5$

- 1) $AM = 113$; adjusted $U_m = 5$; $J = 1$
- 2) $ARM = 112$
- 3) Adjusted $U_m < 8$. Steps 4 and 5 do not apply.
 - (a) does not apply
 - (b) $0 \leq AM - ARM = 113 - 112 = 1 < 2$ Therefore we add 6 to ARM and obtain 118. Substeps (c) and (d) do not apply.
- 6) Therefore the experimental result, expressed in Eight Code is 11.8.

If we had used DCB in the above example, instead of Eight Code, then the coded result would have been "11.4". The ordinary number "11.4" is closer to the original measurement. In general, the penalty for not knowing the code is smaller for DCB, and smallest for Linear Code.

Example. $M = 239$ mV; $U_m = 3$ mV

The experimental result expressed in Eight Code is 239.2 mV.

Now we will work backwards; i.e. we will translate from Eight Code into prose.

The coded integerized mantissa is 2392, and $J = 1$.

The ordinary number 2392 modulo 8 is zero.

Since the mod value is zero, Steps 3 to 5, which assign mod values 1-7, and which cover the uncertainty range ± 4 through ± 25.3 , do not apply.

This means that the uncertainty on the final "2" (in 239.2) is between ± 25.3 and 40, by default.

Therefore the prose value is "239.2, with an estimated measurement uncertainty between ± 2.53 and ± 4 ". (Remember that this is not a confidence interval.)

As an exercise, it might be helpful to work backwards through the first example.

We will see that Eight Code is slightly more cost-effective than Scientific DCB. In comparing the economics of the two techniques, we will emphasize two elements—digital cost and uncertainty resolution.

Step 1 is identical for both techniques. This integerization step determines digital cost. Therefore the digital costs of Eight Code and Scientific DCB are the same.

The uncertainty resolution of Eight Code is slightly better than that of Scientific DCB. In Scientific DCB, the estimated measurement uncertainty (U_m) can be located within a two-fold range, except when the binary marker is in the eights' place, in which case U_m is pinned down within a $40/32 = 1.25$ -fold range.

We could have defined Eight Code such that the specificity for the uncertainty would be identical to that of Scientific DCB, by changing the second magic number in Step 5 of the Eight Code procedure from 25.3 to 32. However it is more efficient to have the non-two-fold range divvied up evenly between two uncertainty categories. (This results in two 1.58-fold ranges, and two two-fold ranges.) Thus Eight Code has a slight edge over Scientific DCB, in terms of efficiency.

All things considered, DCB is a better technique than Eight Code. Eight Code is slightly more efficient, but DCB is more forgiving, more flexible, and simpler than Eight Code. In hindsight, the real significance of Eight Code is as a prototype for future techniques, which are beyond the scope of the present research.

EXPERIMENTAL AND RESULTS SECTION

The efficacy of Eight Code was tested in an experiment, in which an AIM 65 microcomputer was interfaced through an in-house, analog-to-digital converter* to the output of a Perkin-Elmer LC-55 single beam spectrophotometer. As in the DCB experiment, measurements were simulated by twiddling the adjustment knobs.

The BASIC interfacing and coding program was well within the AIM's 4K of memory. The interfacing part of the program was a built upon the skeleton of a program called "DVM", courtesy of Michael Schuyler. (See the Appendix chapter.)

The BCD (not DCB!) signal from the A-to-D converter was outputted to the microcomputer. The BASIC program took a series of measurements, and combined the mean and its standard error into a single Eight Code quantity. The computation time was only a few seconds in each case. On the whole, Eight Code and DCB were equally easy to implement. The experimental results were printed out on computer tapes, and copies of these tapes are listed in the Appendix chapter. The Appendix chapter also contains Eight Code programs for the HP15C scientific calculator.

*The analog-to-digital converter was part of an assembly, called an "AIM box", which was supplied, courtesy of Michael Schuyler.

BIBLIOGRAPHY

1. Skoog, Douglas A.; West, Donald M. "Fundamentals of Analytical Chemistry", 3rd ed.; Holt, Rinehart, and Winston: New York, 1976; 47.
2. Gordon, Roy; Pickering, Miles; and Bisson, Denise J. Chem. Educ. 1984, **61**, 780-781
3. Pinkerton, R.C.; Gleit, Chester E. J. Chem. Educ. 1967, **44**, 232-234
4. Sheppard, W. F. Proceedings of the London Mathematical Society 1898, **29**, 11, 369.
5. Natrella, Mary. G. "Experimental Statistics: National Bureau of Standards Handbook 91", U.S. Government Printing Office: Washington, 1966; Chapter 23.
6. Ibid., 23-2.
7. Skoog and West, op. cit., 79-80
8. Ibid., 47.
9. Ibid., 48.
10. Gordon, Pickering, and Bisson, loc. cit.
11. Sheppard, loc. cit.
12. Eisenhart, Churchill "Selected Techniques of Statistical Analysis for Scientific and Industrial Research and Production and Management Engineering", Eisenhart, Churchill; Hastay, Millard W.; Wallis, W.A. eds., McGraw-Hill Book Company, Inc.: New York, 1947; 192.
13. Hoel, Paul. G. "Introduction to Mathematical Statistics", 3rd ed.; John Wiley and Sons: New York, 1966; 151-155.

APPENDIX

QUESTIONNAIRE

OREGON STATE UNIVERSITY
CHEMISTRY DEPARTMENT
CORVALLIS, OREGON
(503) 754-2081

To: Faculty
From: Stephen Hawkes
Subject: Significant Figure Survey

Date: March 7, 1983

To help us with a paper for J. Chem. Educ. would you please circle what you consider the correct answer to the following question:

Problem: Round the answer to the multiplication
 $1.43 \times 4.098 = 5.86014$ to the correct number of significant figures.

- (a) 6 (b) 5.9 (c) 5.86
(d) 5.860 (e) 5.8601
(f) The question has no unique answer (Explain):

EIGHT CODE PROGRAMS FOR THE HP15C SCIENTIFIC CALCULATOR

The instructions for using these programs are very similar to those for the DCB routines, and do not need to be repeated in great detail.

Eight Code Mantissa Translation Program.

```
1 f LBL D
2 f FIX 0
3 f CLEAR REG
```

Handling boundary condition where the sign is negative

```
4 ENTER
5 g ABS
6 ÷
7 STO 3
8 g LST X
9 STO 0          R0←coded mantissa
```

Calculation raw mod value.

```
10 8
11 ÷
12 f FRAC
13 8
14 X
```

Misc.

```
15 STO-0          R0←rounded value
16 g ABS
17 STO 1          R1←mod value
18 GTO 1          Go to part of program, corresponding to mod value.
```

Mod value = 0; category of highest uncertainty.

```
19 f LBL 0
20 6
21 .
22 5
23 STO 1
24 3
```

25 STO 2

26 GTO 8

Mod value = 1; second-highest uncertainty category.

27 f LBL 1

28 7

29 .

30 5

31 STO 1

32 1

33 STO 2

34 GTO 8

Mod value = 2.

35 f LBL 2

36 2

37 STO-0

38 GTO 9

Mod value = 3

39 f LBL 3

40 2

41 STO+0

Third-highest uncertainty category.

42 f LBL 9

43 7

44 STO 1

45 GTO 8

Mod value = 4

46 f LBL 4

47 3

48 STO-0

49 GTO .0

Mod value = 5

50 f LBL 5

51 1

52 STO-0

53 GTO .0

Mod value = 6

54 f LBL 6

55 1

56 STO+0

57 GTO .0

Mod value = 7

58 f LBL 7

59 3

60 STO+0

Category of lowest uncertainty.

61 f LBL .0

62 5

63 STO 1

Uncertainty Calculation

	X Register	Y Register	Z Register	T Register
64 f LBL 8				
65 2	2			
66 RCL 1	R_1	2		
67 2	2	R_1	2	
68 ÷	$R_1/2$	2		
69 YX	$2R_1/2$			
70 5	5	$2R_1/2$		
71 RCL 2	R_2	5	$2R_1/2$	
72 4	4	R_2	5	$2R_1/2$
73 ÷	$R_2/4$	5	$2R_1/2$	
74 YX	$5R_2/4$	$2R_1/2$		
75 X	Product			
76 RCL 0	$ \bar{x} $	Unc		
77 ÷	$\text{Unc}/ \bar{x} $			
78 g LST X	$ \bar{x} $	Rel unc		
79 RCL 3	Sign	$ \bar{x} $	Unc/\bar{x}	
80 X	\bar{x}	Rel unc		
81 g RTN				

The mean is in the X register, and the estimated relative uncertainty is in the Y register.

Eight Code Formatting Program

```

1 f LBL C      N
Preliminaries
2 g  $\bar{x}$        $\bar{x}$ 
3 STO 1        Register 1 contains the mean.
4 1            1             $\bar{x}$ 
5 STO 1
6 0            0            1             $\bar{x}$ 
7 STO 8

```

Calculate the standard error of the mean and store in Register 0.

```

8 g s          s
9 RCL 2        N            s
10  $\sqrt{x}$        $\sqrt{N}$       s
11 ÷           s/ $\sqrt{N}$ 
12 STO 0
If unc < 4 then scale up.
13 4           4            s/ $\sqrt{N}$ 
14  $x \neq y$       s/ $\sqrt{N}$       4
15 g TEST 9 (x ≥ y)
16 GTO .1
17 ÷           4 $\sqrt{N}/s$ 
18 g LOG       log 4 $\sqrt{N}/s$ 
19 1           1            log 4 $\sqrt{N}/s$ 
20 +          Sum
21 g INT       INT (sum)
22 STO -8     Scale count
23 10x        Scale factor
24 STO X 0
25 STO X 1
26 GTO .2

```

If unc ≥ 40 then scale down.

```

27 f LBL .1    s/ $\sqrt{N}$       4            0
28 4
29 0           40          s/ $\sqrt{N}$       4
30 g TEST 7 (x > y)
31 GTO .2
32 ÷           s/40 $\sqrt{N}$     4            0

```

33	g LOG	$\log s/40\sqrt{N}$	4	0
34	1	1	$\log s/40\sqrt{N}$	
35	+	Sum		
36	g INT	INT (Sum)		
37	CHS	-INT (Sum)		
38	STO-8	Scale count		
39	10^X	Scale factor		
40	STO X 0			
41	STO X 1			

Misc.

42	f LBL .2			
43	g \bar{x}	\bar{x}		
44	ENTER			
45	g ABS			
46	\div			
47	STO 1			
48	g LST X			
49	STO 1	Register 1 contains the mean.		
50	RCL 8	Scale count		
51	CHS	-scale count		
52	10^X	10^{-SC}		

Round to the nearest 8.

53	RCL X 1	$10^{-SC} \bar{x}$		
54	ENTER	$10^{-SC} \bar{x}$	$10^{-SC} \bar{x}$	
55	ENTER	$10^{-SC} \bar{x}$	$10^{-SC} \bar{x}$	$10^{-SC} \bar{x}$
56	8	8	$10^{-SC} \bar{x}$	$10^{-SC} \bar{x}$
57	\div	$10^{-SC} \bar{x}/8$	$10^{-SC} \bar{x}$	
58	.			
59	5	.5	$10^{-SC} \bar{x}/8$	$10^{-SC} \bar{x}$
60	+	Sum	$10^{-SC} \bar{x}$	
61	g INT	Trunc	$10^{-SC} \bar{x}$	
62	8	8	Trunc	$10^{-SC} \bar{x}$
63	X	8 Trunc	$10^{-SC} \bar{x}$	
64	STO 9	Register 9 contains the rounded adjusted \bar{x} .		

How close is the rounding?

65	-	Difference	
66	STO .0	Register .0 contains the rounding loss(+) or gain(-).	

If $Um \geq 8$ then skip ahead to the next higher uncertainty category.

67 RCL 0	Unc	Dif	
68 8	8	Unc	Dif
69 $g \times \leq y$			
70 GTO .3			

Find correct add-on component (mod value) in the smallest uncertainty category.

71 RCL .0	Dif	8	Unc	Dif
72 2	2	Dif	8	Unc
73 GSB A	If yes then use (incrementing) Subroutine A.			
74 CHS	-2	Dif		
75 GSB A				
76 2	4	-2	Dif	
77 STO+9				

If $Unc \geq 16$ then jump ahead to next higher uncertainty category.

78 f LBL .3		
79 RCL 0	Unc	
80 1		
81 6	16	Unc
82 $g \times \leq y$		
83 GTO .4		

Find code for the two smallest uncertainty categories.

84 RCL .0	Dif	16
85 0	0	Dif
86 GSB A		
87 1		
88 STO+9		

Find code for the two largest uncertainty categories.

89 f LBL .4

90 2

91 5

92 .

93 3 25.3

94 RCL 0 Unc 25.3

95 GSB A

96 RCL 8

97 RCL 9

98 RCL X I

99 f FIX 0

100 g RTN

Subroutine A. Increment add-on counter if $x \leq y$.

101 f LBL A n (or Unc) Dif (or 25.3)

102 g TEST 7 Is Dif (or 25.3) < n (or Unc)?

103 g RTN If yes, then return to calling program. Otherwise...

104 I

105 STO+9

106 R↓

107 g RTN

The integerized Eight Code mantissa is in the X register, and the ten's exponent is in the Y register.

COMPUTER TAPES

- Computer tapes for the DCB experiment are listed below. Included are
- 1) the experimental results for two runs
 - 2) the BASIC program for interfacing the digital output of an Orion 701A Ionalyzer with an AIM 65 microcomputer, and for expressing the mean and its standard error of a series of pH measurements as a single DCB number
 - 3) an assembly language subroutine for reproducibly recognizing when practical equilibrium has been attained (on the next page).

```

RUN          9/14/84
UNC0? 05
367E-1      3.652
RUN
UNC0? 07
3651E-1     3.650
LIST
110 POKE 04, 04
120 POKE 05, 15
130 DEF=DEF(0)
140 F=0: I=1: D=10
150 H=15: L=4091
160 D1=F*(I)
170 FOR I=1 TO 3
180 F=DEF*(I+1)
190 N1=INT(F/H)
200 N2=F-H*N1
210 B(I)=D-H*N1+N2
220 NEXT I
230 DAT=.00001*B(1)
    +.001*B(2)+.1*
    B(3)
235 DAT=DAT*25/4
240 INPUT "UNC?": UNC
250 IF UNC=4 GOTO 300
0
260 UNC=D*UNC
270 DAT=D*D*DAT
280 F=F-1

```

```

290 GOTO 250
300 IF UNC=40 GOTO
340
310 F=F+1
320 UN1=UNC/D
330 DAT=DAT/D
340 F=0
350 IF UN1=EXP(40)
+2.1*100 GOTO 380
360 R1=F+1
370 GOTO 150
380 HR=EXP(R1)*LOG(2)
))
390 T=I+T-D*HR
400 R=DAT-I*HR*INT(
T)
410 IF R<=HR GOTO 43
0
420 DAT=D+HR*INT(T)
+3*HR
425 GOTO 440
430 DAT=D*HR*INT(T)
+HR
440 SF=LEN(STR$(DAT)
)-1
450 D#=STR$(DAT)
470 PRINT " "; D#; "E
"; 1-SF
"; 1-SF

```

```

/98 0F01 LDA #00
0F02 00 STA 0000
0F03 01 STA 0001
0F04 02 STA 0002
0F05 03 STA 0003
0F06 04 STA 0004
0F07 05 STA 0005
0F08 06 STA 0006
0F09 07 STA 0007
0F0A 08 STA 0008
0F0B 09 STA 0009
0F0C 0A STA 000A
0F0D 0B STA 000B
0F0E 0C STA 000C
0F0F 0D STA 000D
0F10 0E STA 000E
0F11 0F STA 000F
0F12 10 STA 0010
0F13 11 STA 0011
0F14 12 STA 0012
0F15 13 STA 0013
0F16 14 STA 0014
0F17 15 STA 0015
0F18 16 STA 0016
0F19 17 STA 0017
0F1A 18 STA 0018
0F1B 19 STA 0019
0F1C 1A STA 001A
0F1D 1B STA 001B
0F1E 1C STA 001C
0F1F 1D STA 001D
0F20 1E STA 001E
0F21 1F STA 001F
0F22 20 STA 0020
0F23 21 STA 0021
0F24 22 STA 0022
0F25 23 STA 0023
0F26 24 STA 0024
0F27 25 STA 0025
0F28 26 STA 0026
0F29 27 STA 0027
0F2A 28 STA 0028
0F2B 29 STA 0029
0F2C 2A STA 002A
0F2D 2B STA 002B
0F2E 2C STA 002C
0F2F 2D STA 002D
0F30 2E STA 002E
0F31 2F STA 002F
0F32 30 STA 0030
0F33 31 STA 0031
0F34 32 STA 0032
0F35 33 STA 0033
0F36 34 STA 0034
0F37 35 STA 0035
0F38 36 STA 0036
0F39 37 STA 0037
0F3A 38 STA 0038
0F3B 39 STA 0039
0F3C 3A STA 003A
0F3D 3B STA 003B
0F3E 3C STA 003C
0F3F 3D STA 003D
0F40 3E STA 003E
0F41 3F STA 003F
0F42 40 STA 0040
0F43 41 STA 0041
0F44 42 STA 0042
0F45 43 STA 0043
0F46 44 STA 0044
0F47 45 STA 0045
0F48 46 STA 0046
0F49 47 STA 0047
0F4A 48 STA 0048
0F4B 49 STA 0049
0F4C 4A STA 004A
0F4D 4B STA 004B
0F4E 4C STA 004C
0F4F 4D STA 004D
0F50 4E STA 004E
0F51 4F STA 004F
0F52 50 STA 0050
0F53 51 STA 0051
0F54 52 STA 0052
0F55 53 STA 0053
0F56 54 STA 0054
0F57 55 STA 0055
0F58 56 STA 0056
0F59 57 STA 0057
0F5A 58 STA 0058
0F5B 59 STA 0059
0F5C 5A STA 005A
0F5D 5B STA 005B
0F5E 5C STA 005C
0F5F 5D STA 005D
0F60 5E STA 005E
0F61 5F STA 005F
0F62 60 STA 0060
0F63 61 STA 0061
0F64 62 STA 0062
0F65 63 STA 0063
0F66 64 STA 0064
0F67 65 STA 0065
0F68 66 STA 0066
0F69 67 STA 0067
0F6A 68 STA 0068
0F6B 69 STA 0069
0F6C 6A STA 006A
0F6D 6B STA 006B
0F6E 6C STA 006C
0F6F 6D STA 006D
0F70 6E STA 006E
0F71 6F STA 006F
0F72 70 STA 0070
0F73 71 STA 0071
0F74 72 STA 0072
0F75 73 STA 0073
0F76 74 STA 0074
0F77 75 STA 0075
0F78 76 STA 0076
0F79 77 STA 0077
0F7A 78 STA 0078
0F7B 79 STA 0079
0F7C 7A STA 007A
0F7D 7B STA 007B
0F7E 7C STA 007C
0F7F 7D STA 007D
0F80 7E STA 007E
0F81 7F STA 007F
0F82 80 STA 0080
0F83 81 STA 0081
0F84 82 STA 0082
0F85 83 STA 0083
0F86 84 STA 0084
0F87 85 STA 0085
0F88 86 STA 0086
0F89 87 STA 0087
0F8A 88 STA 0088
0F8B 89 STA 0089
0F8C 8A STA 008A
0F8D 8B STA 008B
0F8E 8C STA 008C
0F8F 8D STA 008D
0F90 8E STA 008E
0F91 8F STA 008F
0F92 90 STA 0090
0F93 91 STA 0091
0F94 92 STA 0092
0F95 93 STA 0093
0F96 94 STA 0094
0F97 95 STA 0095
0F98 96 STA 0096
0F99 97 STA 0097
0F9A 98 STA 0098
0F9B 99 STA 0099
0F9C 9A STA 009A
0F9D 9B STA 009B
0F9E 9C STA 009C
0F9F 9D STA 009D
0FA0 9E STA 009E
0FA1 9F STA 009F
0FA2 A0 STA 00A0
0FA3 A1 STA 00A1
0FA4 A2 STA 00A2
0FA5 A3 STA 00A3
0FA6 A4 STA 00A4
0FA7 A5 STA 00A5
0FA8 A6 STA 00A6
0FA9 A7 STA 00A7
0FAB A8 STA 00A8
0FAC A9 STA 00A9
0FAD AA STA 00AA
0FAE AB STA 00AB
0FAF AC STA 00AC
0FB0 AD STA 00AD
0FB1 AE STA 00AE
0FB2 AF STA 00AF
0FB3 B0 STA 00B0
0FB4 B1 STA 00B1
0FB5 B2 STA 00B2
0FB6 B3 STA 00B3
0FB7 B4 STA 00B4
0FB8 B5 STA 00B5
0FB9 B6 STA 00B6
0FBA B7 STA 00B7
0FBB B8 STA 00B8
0FBC B9 STA 00B9
0FBD BA STA 00BA
0FBE BB STA 00BB
0FBF BC STA 00BC
0FC0 BD STA 00BD
0FC1 BE STA 00BE
0FC2 BF STA 00BF
0FC3 C0 STA 00C0
0FC4 C1 STA 00C1
0FC5 C2 STA 00C2
0FC6 C3 STA 00C3
0FC7 C4 STA 00C4
0FC8 C5 STA 00C5
0FC9 C6 STA 00C6
0FCA C7 STA 00C7
0FCB C8 STA 00C8
0FCC C9 STA 00C9
0FCD CA STA 00CA
0FCE CB STA 00CB
0FCF CC STA 00CC
0FD0 CD STA 00CD
0FD1 CE STA 00CE
0FD2 CF STA 00CF
0FD3 D0 STA 00D0
0FD4 D1 STA 00D1
0FD5 D2 STA 00D2
0FD6 D3 STA 00D3
0FD7 D4 STA 00D4
0FD8 D5 STA 00D5
0FD9 D6 STA 00D6
0FDA D7 STA 00D7
0FDB D8 STA 00D8
0FDC D9 STA 00D9
0FDD DA STA 00DA
0FDE DB STA 00DB
0FDF DC STA 00DC
0FE0 DD STA 00DD
0FE1 DE STA 00DE
0FE2 DF STA 00DF
0FE3 E0 STA 00E0
0FE4 E1 STA 00E1
0FE5 E2 STA 00E2
0FE6 E3 STA 00E3
0FE7 E4 STA 00E4
0FE8 E5 STA 00E5
0FE9 E6 STA 00E6
0FEA E7 STA 00E7
0FEB E8 STA 00E8
0FEC E9 STA 00E9
0FED EA STA 00EA
0FEE EB STA 00EB
0FEF EC STA 00EC
0FF0 ED STA 00ED
0FF1 EE STA 00EE
0FF2 EF STA 00EF
0FF3 F0 STA 00F0
0FF4 F1 STA 00F1
0FF5 F2 STA 00F2
0FF6 F3 STA 00F3
0FF7 F4 STA 00F4
0FF8 F5 STA 00F5
0FF9 F6 STA 00F6
0FFA F7 STA 00F7
0FFB F8 STA 00F8
0FFC F9 STA 00F9
0FFD FA STA 00FA
0FFE FB STA 00FB
0FFF FC STA 00FC

```

```

0F5E A9 LDA #10
0F60 20 BIT 0FFF
0F62 D0 BNE 0F68
0F65 8E STA 0FFF
0F68 CD JMP 0FFF
0F6E D6 BNE 0F76
0F8D 28 LSR 0F77
0F76 4C JMP 0F4E
0F7D 4C JMP 0F6F
0F7F 3A ROR
0F77 8D LDA 0FFC
0F7A 8D STA 0FF9
0F7D 8D LDA 0FFD
0F83 8D STA 0FFA
0F8D 8D LIA 0FFE
0F8E 8D STA 0FFB
0F8F 68 RTS
0F8A A9 LDA #00
0F8C 8D STA 0FFC
0F8F 8D STA 0FFD
0F92 8D STA 0FFE
0F95 A2 LDX #10
0F97 A9 LDA #02
0F99 2C BIT 0000
0F9C F6 BEQ 0F99
0F9E AD LDA 0000
0FA1 F6 BEQ
0FA2 13 LDI
0FA3 8D LDI 0FFC
0FA6 8D STA 0FFC
0FA9 AD LDA 0001
0FAC 8D ADC 0FFD
0FAF 8D STA 0FFD
0FB2 A9 LDA #00
0FB4 5D ROR 0FFE
0FB7 8D STA 0FFE
0FBF D6 LDI
0F86 CA DEL
0F8C D6 BNE 0F97
0F8E 68 RTS

```

The next 1.6 pages are a listing of the BASIC program for interfacing a Perkin-Elmer LC-55 spectrophotometer with an AIM 65 microcomputer, and for expressing the mean and its standard error for a series of measurements into a single Eight Code number. After that, the results for several runs of the Eight Code experiment are listed.

ROCKWELL AIM 65

8/11/84

```

1000:=
1001:=
LIST
1 REM ROCKWELL PROGRAM
2 REM DVM SKELETON
5-610
3 PRINT "DVM/ABSORBANCE="
4 INPUT "PRINT INTERVAL"
5 DIM A(100) T(100)
10 B=40000 F=10 N=2
20 U=16
40 W=250
20 POKEB+1:POKEB+
20,255:POKEB+11,64
25 POKEB+1:POKEB+192:POKE
B+13:POKEB+12,
21 POKEB+4,75
28 FOR I=1 TO N:FOR J=1 TO N
A(POKEB+I)=NEXT
29 D=198/2040/2048
30 INPUT "DATA POINTS" N
32 PRINT "N=" N
35 IF N<192 AND N>16 GOT
040
38 PRINT "THE NUMBER
OF DATA POINTS MUST BE FROM
17 PRINT "1 TO 100"
GOTO 30
40 F=0:GOTO 150
75 IF T(1) THEN PRINT:
"INTERVAL MUST BE AT
LEAST 0.2 SECONDS"
150 PRINT "HIT SPACE
TO BEGIN"
200 FOR I=1 TO N
210 IF I=192 GOTO 218
216 GETA:IFA#C>" "
GOTO 218
218 IF I=1 THEN POKEB+
5,195:POKEB+2,POKE
253,0:POKEB+4,0

```

```

220 J=PEEK(253)+V*P
EEK(254)
240 IF J<1*(1-1)*FG
0T0220
245 T(I)=J
250 POKEB+1,1
260 POKEB+12,192:PO
KEB+12,224
310 L=PEEK(B):POKEB
+1,9:POKEB+
320 A(I)=L+U+H/U-0
330 PRINT "A(I)/D:"
MV"
390 NEXT
440 T=0:B=0
500 FOR I=1 TO N:S=S+A
(I):T=T+A(I)*A(I)
511 NEXT
515 AV=S/N/D
520 PRINT "AV="
AV
525 W=(T/D)/N*AV*A
V)/(N-1)
530 PRINT "SD=":SOR
(W):"SD"
540 GOTO 300
600 DATA 72,173,4,16
0,153,252,208,10,169
,2
610 DATA 113,252,230
,253,208,2,230,254,1
04,64
700 REM CONVERT MV
TO CONC
710 TR=AV/8000
720 STR=FOR(U)/8000
730 AB=-LOG(TR)/LOG
(10)
740 DAB=-LOG(TR+STR
)/LOG(10)
750 SAB=ABS(AB-DAB)
760 DAT=AB+SAT
770 UNC=SAB*SAT
780 PRINT "CONC=":D
AT: "+-":UNC
790 REM ADJUST UNC
800 F=0
810 IF UNC>40 GOTO 860
820 UNC=10+UNC
830 DAT=10+DAT
840 F=F+1
850 GOTO 810
860 IF UNC<40 GOTO 910
870 UNC=UNC/10

```

```

880 DAT=DAT/10
890 F=F+1
900 GOTO860
910 REM CODING
920 DF=DAT-8*INT(DA
T/8+.5)
930 AD=0
940 IFUNC=16*SQR(2.
5)GOTO1140
950 IFUNC=16GOTO980
960 AD=1
970 GOTO1140
980 IFUNC=8GOTO1040
990 IFDF>8GOTO1020
1000 AD=1
1010 GOTO1140
1020 AD=2
1030 GOTO1140
1040 IFDF>2GOTO107
0
1050 AD=4
1060 GOTO1140
1070 IFDF>0GOTO1100
1080 AD=5
1090 GOTO1140
1100 IFDF>1GOTO1130
1110 AD=6
1120 GOTO1140
1130 AD=7
1140 PRINT " " " " "
1150 PRINT " DAT-DF+
AD " "E" "F
1160 PRINT " "IPRIN
T!" "

```

CONCURRENCE= .

1

N= 3

3223.61869 MV

2629.14571 MV

2409.54771 MV

AV MV= 2620.77052

SD= 407.0696 MV

CONC= .451712231 +-

.0585481125

52 E-2

40-35-30

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

CONC/ABSORBANCE=

1
N= 3
4025.11563 MV
3925.64324 MV
4050.25126 MV
AV MV= 4001.67504
SD= 63.6255616 MV
CONC= .300048168 +-
6.05127289E-03
308 E-3 50-50-50

CONC/ABSORBANCE=

2.311
N= 3
7535.17589 MV
7258.29141 MV
6831.65113 MV
AV MV= 7197.7852
SD= 25.16542 MV
CONC= .104672058 +-
.0481565491
15 E-1 95-90-85

500nm

CONC/ABSORBANCE=

1
N= 3 52-50-48
4168.44111 MV
4040.80180 MV
3873.33441 MV
AV MV= 4013.17588
SD= 155.813453 MV
CONC= .107117518 +-
.0164568213
297 E-1

CONC/ABSORBANCE=

1
N= 3 53-50-47
4273.86935 MV
3939.93849 MV
3762.9195 MV
AV MV= 3993.16248
SD= 250.129127 MV
CONC= .301220938 +-
.0255675885
304 E-3

CONC/ABSORBANCE=

1
N= 3 43/50/57
3473.69935 MV
4050.25126 MV
4515.07533 MV
AV MV= 4015.07538
SD= 516.461608 MV
CONC= .105335284 +-
.0527455708
36 E-1

CONC/ABSORBANCE=

1
N= 3 60-50-40
4753.79137 MV
4025.11563 MV
3221.16533 MV
AV MV= 4001.67504
SD= 769.113487 MV
CONC= .300048168 +-
.0762462491
37 E-2

CONC/ABSORBANCE=

1

N= 3

2845.73267 MV

4036.15675 MV

5183.41769 MV

AV MV= 4022.1665

SD= 1168.37413 MV

CONC= .253353676 +-
.110862797

34 E-2

CONC/ABSORBANCE=

1

N= 3

70-50-30

5646.24121 MV

4005.02513 MV

2432.1566 MV

AV MV= 4026.47571

SD= 1668.16844 MV

CONC= .237349237 +-
.145879959

34 E-2

CONC/ABSORBANCE=

1

N= 3

46-50-54

3636.63327 MV

4020.1665 MV

4349.24623 MV

AV MV= 4006

SD= 359.717931 MV

CONC= .301029995 +-
.0373984

304 E-3

CONC/ABSORBANCE=

1

N= 3

56-50-44

4515.07538 MV

4045.22613 MV

3572.66423 MV

AV MV= 4044.38861

SD= 471.1061 MV

CONC= .296237108 +-
.047852459

36 E-2

CONC/ABSORBANCE=

1

N= 3

45-50-50

3648.24121 MV

4037.68844 MV

4329.14573 MV

AV MV= 4005.02513

SD= 341.625397 MV

CONC= .300484742 +-
.0355494787

304 E-3

CONC/ABSORBANCE=

1

N= 3

4145.72664 MV

4027.63813 MV

3997.43744 MV

AV MV= 4056.55142

SD= 78.3474065 MV

CONC= .294860179 +-
.030705375E-03

298 E-3

Unadjusted 51-50-49