THE CASE FOR CRUNCHY METHODS IN PRACTICAL MATHEMATICS

Michael Wood
Department of Accounting and Management Science
Portsmouth Business School
Locksway Road, Milton
Southsea, Hants, PO4 8JF
England
email: michael.wood@port.ac.uk

Abstract
This paper focuses on the distinction between methods which are mathematically "clever", and those which are simply crude, typically repetitive and computer intensive, approaches for "crunching" out answers to problems. Examples of the latter include simulated probability distributions and resampling methods in statistics, and iterative methods for solving equations or optimisation problems. Most of these methods require software support, but this is easily provided by a PC. The paper argues that the crunchier methods often have substantial advantages from the perspectives of user-friendliness, reliability (in the sense that misuse is less likely), educational efficiency and realism. This means that they offer very considerable potential for simplifying the mathematical syllabus underlying many areas of applied mathematics such as management science and statistics: crunchier methods can provide the same, or greater, technical power, flexibility and insight, while requiring only a fraction of the mathematical conceptual background needed by their cleverer brethren.

Introduction
Imagine a girl with 155 pence to spend on chocolate bars costing 37 pence each. If she has a thorough command of arithmetic she will simply divide 155 by 37, obtain 4.2 and realise that she can buy four chocolate bars and be left with some change. If she is not sure about division but understands about multiplication, she might proceed by guessing the answer and then checking by multiplying:

37 x 3 = 111, and
37 x 4 = 148, but
37 x 5 = 185, so

she can buy four, but not five, bars. Alternatively she may add 37 to itself until she reaches a total which is more than 155. It should then be obvious to her that the number of bars she can buy is one less than the number of 37's she has added together:

37+37+37+37=148, but
37+37+37+37+37=185

so she can buy four bars. Finally, she could adopt a simpler strategy still by getting her 155 pence in individual one penny coins, and then counting them out in piles of 37 to see how many complete such piles she has. There are thus (at least) four approaches to this problem: division, guess-multiply-check, repeated addition and counting.

There is often a similar choice with more advanced problems. Algebraic equations may be solved symbolically (sometimes), or by numerical, trial and error methods; definite integrals can be evaluated symbolically (sometimes) or numerically; optimization problems can be tackled by the methods of the calculus (sometimes) or by a search heuristic; probability distributions can be investigated either by mathematical theory (sometimes) or by computer simulation; queues can be modelled by means of probability theory (sometimes) or by simulation; and so on.

The common element in all these examples is the choice between using a crude repetitive method to *crunch* out the answer, and using sophisticated (relatively speaking), or *clever*, mathematics. This paper treats the *crunchiness* of a mathematical method as a general property, and considers the advantages and disadvantages of greater or lesser crunchiness. The conclusion is that crunchier methods have a number of substantial advantages over cleverer methods: these include the fact that they are conceptually more straightforward and they tend to be of more general applicability and require fewer restrictive assumptions (note the frequent occurrence of the qualifying "sometimes" in the previous paragraph).

My perspective in this paper is *not* that of an educationalist. I am not interested in how to ensure that the girl above, and her friends, understand division, but that they have available a useful range of approaches to problems - which may or may not include the concept of division. Similarly, managers wishing to use management science techniques and medical researchers using statistical techniques are not concerned with learning mathematics for its own sake, but in the availability of a practical and reliable cognitive and technological toolkit for approaching their problems. These toolkits include artifacts of various kinds (calculators, computers, etc). Needless to say these artifacts are a crucial, and changing, factor in the situation. At present, the most important of the these artifacts are computer packages (e.g. spreadsheets, simulation packages) and, at a more elementary level, calculators.

The arguments concerning crunchy methods have implications for the design of artifacts to support cognitive processes, and for the mathematics curriculum - particularly for older students who need to use mathematical techniques of various kinds for practical purposes. This paper argues that crunchy methods have the potential to offer very substantial improvements in the user-friendliness, power and reliability of the cognitive and technological toolkits available to these students.

The use of mathematically based techniques for practical purposes by people who are not experts in the techniques, or the mathematics underlying them, poses widely acknowledged problems in areas such as management and engineering, and most of the many fields to which statistics is applied (see, for example, Yilmaz, 1996; Mar Molinero, 1996; Stuart 1995; Romero et al, 1995; Greenfield, 1993; Bailey and Weal, 1993; Altman and Bland, 1991). The symptoms of the problems include dislike, often bordering on fear, of the techniques and the courses which teach them, as well as the consequent under-use and mis-use of the techniques in practical contexts; the remedies suggested usually centre on relating the subject more closely to real world applications. One argument of this paper is that a move to crunchier methods provides a very powerful means of tackling these problems.

Crunchy methods

We must start by defining the term "crunchy". In general, "clever" methods employ a mathematical theorem to deduce the corollaries of assumptions about a situation; crunchy methods use the assumptions directly to work out the corollaries, sometimes with the aid of some common sense or heuristic principles.

To be more precise, we can say that Method A is crunchier than Method B if Method A is

1 More directly linked to the intuitive definition of the situation (*more transparent*); and

2 Likely to involve *more repetitive* steps; and

3 Dependent on fewer, or lower level, technical concepts and theorems, so can be followed by users with a lower level of technical expertise (*lower technical level*)

than Method B.

The choice of the word "crunchy" stems from the phrase "number crunching" and is intended to convey the idea of a crude "crunching" through of stages or possibilities. Several other phrases have a similar meaning in specific contexts: these include brute force methods, iterative heuristics, numerical methods, computer-intensive methods (based on simulation) in statistics (Noreen, 1989; Simon, 1992) and trial and error methods. However none of these really has the right meaning across the full range of situations I have in mind. The closest is probably "brute force", but this does not seem appropriate to crude methods of avoiding division, and there is also no convenient comparative form like "crunchier".

I will illustrate the notion of crunchiness in the context of the chocolate problem. The initial assumptions are that the shopkeeper will want 37 pence for the first bar, and another 37 pence for the second bar, and so on (discounts are not available). The cleverest of the four methods is the division strategy. The guess-multiply-check strategy is crunchier than the division strategy because it is more clearly linked to the idea that the price of four items is four times the price of one (point 1 of the definition), is likely to involve some incorrect guesses so the guess-multiply-check sequence needs repeating (point 2), and does not depend on an understanding of the concept of division - (point 3). A similar argument demonstrates that the repeated addition strategy is crunchier still (by all three criteria), and the counting strategy is the crunchiest of them all (again, by all three criteria).

Similar considerations apply to more advanced problems: for example the estimation of the probability of a family of four children comprising two girls and two boys. This can easily be worked out using the binomial probability distribution which gives an answer of 37.5% (making the usual assumptions of independence and equal probabilities for boys and girls). Alternatively the same answer can be obtained by computer simulation of, say, 10,000 simulated families. Figure 1 shows the result of such a simulation: the proportion of two girl families in this simulated set of families is 37.1%.

Figure 1: Simulation of 10,000 families of four

0 girls	XXXXXXX
1 girl	XXXXXXXXXXXXXXXXXXXXXXXXXXX
2 girls	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
3 girls	XXXXXXXXXXXXXXXXXXXXXXXXXX
4 girls	XXXXXX

(X represents 93 families.)

Proportion of 2 girl families: 37.1%

This latter approach is crunchier because:

1 The simulation is defined directly by the definition of the situation: each simulated family has four children and in each case there is a 50% chance of the child being a girl.

2 It involves simulating the 10,000 families and so involves repeating the same process 10,000 times instead of using the "clever" binomial formula.

3 It involves no more conceptual background than the assumption of statistical independence between the sexes of different children in the same family, and the notion of a probability as a long run proportion. Use of the binomial distribution, on the other hand, requires knowledge of the addition and multiplication rules of probability and the idea of "combinations" if the distribution is worked out from first principles. Alternatively, if the binomial distribution is taken "on trust", the user needs to have mastered the concept of the binomial distribution as a mathematical entity, the assumptions under which it is valid, and the nature of the information necessary to use the binomial equations. The conceptual background needed here is far richer and more extensive than for the simulation method.

Statistical process control (Shewhart) charts entail using probability theory to monitor an ongoing industrial or business process. The conventional methods, based on clever formulae derived from probability distributions such as the binomial, are difficult for the typically non-expert users to understand and interpret meaningfully (Hoerl and Palm, 1992; Wood and Preece, 1992), and, in addition, the probability models often fail to fit closely the patterns found in real processes. A crunchy approach - resampling (Noreen, 1989, Simon, 1992) - has been suggested to make the methods more user friendly (Wood et al, 1999), and to make them mirror reality more closely (Bajgier, 1992; Seppala et al, 1995). Concepts such as the standard deviation, central limit theorem, binomial distribution and the normal distribution are an essential part of understanding how the conventional methods work, but are irrelevant to an appreciation of the crunchy method.

There are many other examples. One is the calculation of the economic order quantity in inventory management (see, for example, Dennis and Dennis, 1991, 452-7). This is the order quantity which minimises the total costs to a business of ordering and of carrying stock. There is a simple formula, based on a number of assumptions, which can be used for this purpose (Dennis and Dennis, 1991, p. 453) - this represents the clever method. Alternatively, formulae for the ordering and carrying costs, and for the total cost, can be set up on a spreadsheet, and then different order quantities can be "tried out" to see which makes the total cost as low as possible. (Most modern spreadsheets have a "solver" or an "optimiser" which enables this process to be automated, although the results are generally not infallible and users would be advised to check - for example - that the suggested optimum is in fact reached from different starting points - see the appendix for an illustration of this.) This approach is crunchier because it follows directly from a simple model of the costs, because it involves a repetitive trial and error process (even if this is performed automatically by the spreadsheet), and because users do not need to understand the formula for the economic order quantity or its rationale (which involves the differential calculus).

A very similar argument applies to evaluating an integral numerically instead of symbolically, solving an equation by trial and error rather than analytically, using bootstrapping and resampling methods instead of probability theory for statistical inference (Noreen, 1989; Simon, 1992), and to many similar examples. The computer Deep Blue which has recently defeated the world chess champion Gary Kasparov provides another illustration of the theme. The strength of the computer is ability to calculate the consequences of more possible moves further ahead than can a human being. Against this, skilled human players have a more extensive and flexible repertoire of clever, but often intuitive, strategies. The crunchy approach adopted by Deep Blue has been shown to be more effective than the cleverest human being.

These examples show that the use of crunchy methods is far from being restricted to mathematical novices. In some cases, the "clever" method may not exist or may not have been invented yet (e.g. some integrals and the roots of some polynomials cannot be expressed as algebraic expressions of a convenient form), so a crunchier approach (e.g. numerical methods to find an integral or the polynomial root) may be the only possibility. In these cases Method B in the above definition would be a hypothetical method. Sometimes a crunchy method may be helpful as a quick approach for exploring a situation even if a cleverer method is feasible - an example of this appears in the appendix.

The notion of crunchiness thus has three dimensions: transparency, repetitiveness, and low technical level. There is no logically conclusive reason why these three dimensions should always go together, although the three examples above make, I think, a plausible case that they often do. Sometimes the heuristics employed for a search procedure may be highly sophisticated - e.g. genetic algorithms, the Simplex method for linear programming, the methods used by spreadsheet solvers and optimizers - so the method may be crunchy in terms of criteria 1 and 2 above, but possibly not on criterion 3. It is also important to note that the definition of each dimension is slightly vague: how, for example, is the technical level to be defined? Similarly, if our guess-multiply-check shopper strikes lucky with the initial guess, repetition may be unnecessary. Strictly we have defined a means of comparing methods, and so a continuum of methods of varying crunchiness, and not absolute definitions of crunchy and clever methods, but it is helpful to speak loosely of the crunchier methods as being "crunchy".

My aim in this paper is to bring these ideas together under one umbrella, and explore the implications for designing practical mathematical curricula and the artifacts necessary to support them.

Properties of crunchier methods

I will start with three positive points, and then go on to one neutral and three negative ones.

1 Crunchy methods are conceptually simpler

The three criteria used to define the crunchiness of methods imply three important senses in which crunchier methods are simpler: they are more transparent in the sense that they are more directly linked to the definition of the problem, they involve repetition of similar steps, and they demand less in the way of technical conceptual background from users. Each of these is likely to make crunchy methods simpler to understand - in the deep or relational (Skemp, 1976) sense - than their clever equivalents.

2 Crunchy methods tend to be more reliable in the sense that inappropriate use - misuse, misinterpretation, or failure to use when appropriate - is less likely. For this reason users may regard them as more trustworthy.

This follows directly from the first point: if the methods are more transparent and depend on less technical background, then users must be less likely to misuse them. (Like the first point, this is "almost" a tautology. If these points were found not to be true of some supposedly crunchy method, then, by definition, the method cannot be as transparent as supposed. The qualification implied by the word "almost" is due to the three dimensional nature of crunchiness - e.g. it is possible that problems due to the repetitive nature of a crunchy method might cancel out the advantages of greater transparency.) This implies that, in practice, the applications of crunchier methods are likely to be more rigorous (than cleverer methods) because the conditions on which they are based are more transparent.

The difficulties experienced by beginners in subjects such as statistics (see Introduction above) illustrate some of the problems due to the use of clever methods. Pfannkuch (1997), for example, says that "it did not seem to occur to some students to use a significance test ... Underlying this aspect seemed to be a lack of understanding of variation in relationship to significance testing." Using a crunchy method such as resampling (Noreen, 1989; Simon, 1992), which involves simulating the variation in question and requires none of the paraphernalia of mathematically defined probability distributions, means that the rationale behind the method and its relationship to practical situations are more transparent. This is not, of course, to claim that the problems of teaching statistics will all be eliminated, but that some of the obstacles can be removed.

There is a counter-argument to this that crunchy methods may sometimes (but by no means always) be less rigorous from a mathematical point of view - this is discussed under Property 6 below.

3 Crunchy methods tend to be more general and so more powerful and more able to model complex phenomena

The essence of a clever method is that assumptions have to be made about the structure of the situation in order to deduce a way of working out the answer. With crunchy methods, some of these assumptions are often implicit in the fact that a step is repeated, but these tend to be less restrictive and easier to override. For example, if the shopkeeper offers a discount for people buying more than two bars of chocolate, this is easier to build into the crunchier methods. Similarly, discounts can easily be built into the spreadsheet method for estimating the economic order quantity (see above), but the standard formula is based on a number of assumptions, one of which is that there are no discounts; the simulation approach to the family problem can easily (depending on the software) be adapted to take account of the possibility of twins and triplets whereas the clever (binomial) formula cannot; and the resampling approach to statistical process control charts avoids the (frequently unrealistic) assumption that distributions are normal (Bajgier, 1992; Seppala et al 1995).

These examples illustrate the way that crunchy methods can avoid (to some extent) the necessity to impose a simple structure on reality because this is necessary for the mathematics. There is a growing feeling in some corners of academia (e.g. in mathematics - Stewart, 1990 especially pp 81-4 - and economics, physics, biology and other areas - Waldrop, 1994) that, contrary to the optimism engendered by the Newtonian world view, God may not after all be a mathematician: the world is more complex and less structured than mathematicians would like it to be. The initial impression that the world follows simple, linear mathematical patterns may be simply a function of mathematicians ignoring anything which does not fit this assumption. To the extent to which this is true, crunchy methods are, by definition, more suited to real problems in a complex world than are clever ones. There is, for example, increasing use of heuristic methods for searching for solutions to complex, "messy" problems in management science (Pidd, 1996, pp. 290-310).

Another aspect of the same point is that crunchy methods may be more general in the sense that they may incorporate several conventional clever methods. The formulae for the binomial distribution will only model this particular distribution; the crunchy method (as implemented on a computer program) on the other hand (Figure 1) will also produce a (simulated) probability distribution for the means of random samples from any empirically specified distribution, and could easily be adapted to draw random samples without replacement - thus simulating the hypergeometric distribution. The same software and essentially the same method can be used to derive bootstrap confidence intervals for the mean (Gunter, 1991), and estimate action lines for mean and range, and median and standard deviation, and many other control charts for quality control purposes (Wood et al, 1999). Similarly, trial and error methods of solving equations work for any equation provided that the user has a means of evaluating the two sides: clever methods on the other hand require that the equation is of a particular type - e.g. there is one clever method for linear equations, another for quadratic equations, another for trigonometric equations of a particular kind, and so on.

In addition to this, crunchy methods are often possible in situations where no clever method has been invented, or even where it has been proven that none can exist (see above for examples).

4 Any method encourages the development of concepts which refer to the answers produced. Crunchy methods, being different from clever methods, may lead to a different set of concepts

Imagine someone who uses the counting strategy for problems like the chocolate one. If this method becomes "interiorized to become a process [so that it is possible for] the individual to think about it as a totality" (Cornu and Dubinsky, 1989), the individual in question is almost bound to develop, implicitly or explicitly, a concept which refers to the answers produced by this counting strategy. Someone tackling the chocolate problem with the operation of division and a calculator, on the other hand, will have the concept of a "quotient" (or "answer produced by division") to describe the answer produced by the method. The counting strategy does not produce quotients - the answer is always an integer - and the concept developed is slightly different (and difficult to describe neatly).

Sometimes the clever and crunchy methods may lead to essentially the same concept for the answer. The binomial simulation in Figure 1 leads to a proportion which is a binomial probability in just the same sense as the answer produced by probability theory. The methods may be different but the answers both refer to the same concept. On the other hand, if the same simulation software is used to simulate other distributions (as described below), the concepts for the answers produced by the crunchy method may be more general than those for the clever method.

These concepts are important. A very plausible theory of mathematical learning (Leron and Dubinsky, 1995; Cornu and Dubinsky, 1989) asserts that the process of interiorizing an "action" or a method so that it can be viewed as "a totality" or a "higher level" object is a vital part of mathematical learning. In the language of cognitive psychology, viewing a method as a whole as a single "chunk" of information (a "maximal familiar substructure") is helpful for higher level thinking about the method (or the results of the method) since human information processing can handle no more than about seven chunks of information at a time (Simon, 1996). In the case of the mathematical methods we have been discussing, a core aspect of the this higher level concept or chunk of information is the nature and interpretation of the answer produced.

Sometimes concepts developed through the clever method may be more useful than those developed by crunchy methods. The notion of a quotient which may be fractional, for example, is an essential prerequisite for understanding the output of the crunchy binomial simulation. On other occasions the greater generality of crunchy methods may lead to their underlying concepts being more useful because of this greater generality.

One point to note here is that, for a variety of fairly obvious reasons, the vocabulary for describing answers from clever methods is likely to be far richer and more established than the corresponding vocabulary for crunchy methods: e.g. I could not think of a suitable term for the answer to the chocolate problem from the counting strategy. It may be helpful to invent terms to label some of these implicit concepts.

5 Crunchy methods tend to be computationally intensive or slow

This is a consequence of the repetitive nature of the methods. How serious a problem it is clearly depends on the balance between the power of any computer support available and the number of repetitions necessary.

6 Crunchy methods may only yield an approximate or tentative or unproven answer

Simulation methods will not yield an exact probability, and guess and check methods will yield an exact answer in the chocolate problem (where the answer has to be an integer) but may not in similar problems where the possible answers lie on a continuum. However, in both cases the method can yield an answer to any desired degree of accuracy - which is all that is required in most practical situations.

With optimisation problems, the typical crunchy search procedure may be fallible if, for example, the search heuristic finds a local optimum rather than the required global optimum. A cleverer method may find the global optimum, and provide a proof that it is in fact optimal. On the other hand, the difficulty is often that there are no viable clever methods so there may be no alternative to a crunchy heuristic approach (Pidd, 1996, chapter 10).

In practice, the potential advantages of the greater accuracy and rigour of some (not all) clever methods compared to their crunchier alternatives, may be nullified if the clever methods are misused or misinterpreted - see Property 2 above.

7 Crunchy methods do not provide a general answer and so may fail to provide insights into the structure of the situation

For example, clever methods for solving polynomial equations (e.g. factorisation) demonstrate that the maximum possible number of solutions is equal to the degree of the polynomial, and the formula for the standard deviation of a binomial distribution indicates the relation between sample size and the spread of the distribution. No such insights would be likely from the use of a search heuristic or simulation method.

The first three of these properties (the positive ones) are to some extent offset by the last three. However it is important to stress the enormous practical advantages of the first three points: the simplicity of crunchy methods, and their potential generality and ability to model complex situations. To take two examples of the first point, simulation methods can be used as a substitute for probability theory (see, for example, Simon, 1992) with the consequent simplification of large areas of statistical theory, and the Solver built into the spreadsheet Excel will solve a very large variety of equations and optimisation problems. The second point is equally important, and linked to the first point in that the general nature of crunchy methods means that fewer such methods are necessary to deal with a given range of problems with a subsequent simplification of the task of learners and users. In addition crunchy methods will reach areas which clever methods cannot: this is illustrated by the use of simulation and heuristic methods in areas as diverse as operational research, meteorology, economics and so on. Clever methods can only work if the universe conforms to the assumptions on which they are based; if it does not, crunching out the answers is the only option.

Computer support for crunchy methods

Clever and crunchy methods can both be implemented with or without machine support - as is illustrated by the chocolate example. However, for more advanced problems, crunchier methods are likely to be more dependent on computer support systems (CSSs) than are cleverer methods. It is easy to use the binomial probability formula without the help of a computer. On the other hand, while it is possible to simulate sufficient families of four to estimate the required probabilities without a computer, it is not really a practical proposition. Many crunchy methods are only practically feasible because of the ready availability of computers. Accordingly, it is helpful to consider the computer systems in question. (I will assume that the artifacts for supporting crunchy and clever methods are computer packages - which seems a reasonably assumption at the present state of evolution of technology.)

A computer support system for a clever method fulfils some or all of three functions:

1 it helps the user to develop or choose the appropriate method; and/or
2 it implements the method; and/or
3 it helps the user to interpret the results of the method.

For example the calculator used for the chocolate problem just supports function 2; on the other hand many statistical packages give the user help with 1 and 3 too. However, at the present state of the art, computers are far more helpful for 2 than they are for 1 and 3. The reasons for this are fairly obvious: one major problem is often the lack of a common frame of reference for communication between user and software system (Wood, 1989) - non-expert users of statistical packages, for example, may lack a clear understanding of essential concepts like "significant", "interaction", "main effect" and even terms like "variable" and "data".

In principle, a computer package to support a crunchy method could support the same three functions. However, the enormous advantage of a crunchier method is that 1 and 3 (the functions which are difficult to support) are much less problematic so support is less necessary - but this is only likely to be true if the user knows what the CSS is doing. If some intuitive procedure is being repeated many times, it is important that the user appreciates exactly what the repeated procedure is. Then, given the facts that the rationale behind this follows directly from the intuitive understanding of the situation, and that the concepts involved are relatively simple, the whole method should be clear so it should be obvious when it is useful and what the answers mean. In short, the problems users typically face using a mathematical CSS are solved.

This means that a crucial feature of a crunchy computer support system is that the method implemented should be transparent to the user. The obvious way to achieve this is to allow the user the option of stepping through the method step by step. Then, when it is clear how it works, the method can be accelerated and the details hidden from view.

How might this work in practice? A crunchy binomial support system allows users to simulate the first few families individually, and see the results put on the histogram - as in Figure 2 below. Similarly the solver or optimiser on most spreadsheets allows users the option of seeing the first few iterations.

Figure 2: First two simulated families of four

1st simulated family (girl=1, boy=0): 1 0 1 1

Number of girls in this family is 3

---

0 girls
1 girl
2 girls
3 girls X
4 girls
(X represents 1 family.)
Proportion of 2 girl families: 0%

2nd simulated family (girl=1, boy=0): 0 0 0 0

Number of girls in this family is 0

---

0 girls X
1 girl
2 girls
3 girls X
4 girls
(X represents 1 family.)
Proportion of 2 girl families: 0%

We may contrast this with a package for implementing a clever method. Here nothing would be gained by stepping through the method a step at a time because the method is not directly linked to intuitions about how the situation works, it is not repetitive and the conceptual background it links to is relatively complex. A Help facility would be a possibility but the help is likely not to be very helpful if users lack an extensive understanding of probability theory. In practice, users of a statistics package such as SPSS tend to treat it as a black box and are often unaware of the methods used by the package to obtain the answers. This is in strong contrast to the way a crunchy CSS can show users the steps of the method and some intermediate results. This is another reason for the extra transparency and reliability of crunchy methods when implemented with suitable software.

Strictly, the CSSs shown in Figures 1 and 2 are unlikely as practical CSSs because they refer specifically to the family problem. (I have changed the output to clarify the argument of this paper for readers not familiar with the subject area.) To be useful, a practical CSS needs to be able to support the solution of a range of different problem types - not just problems about girls in families of four, but about boys in families of any size, about defectives in samples taken for quality control purpose and about scores in multiple choice tests - all of which can be modelled by the binomial distribution.

In fact, Figures 1 and 2 (except words like "family" and "girls") are produced by a simple program whose uses are far wider than the binomial distribution. The program, RESAMPLE, stores a list of numbers (data) and then takes random "resamples" (with replacement) from this list and analyses the mean, sum, standard deviation, median or any percentile of the resamples. The program can be used to produce bootstrap confidence intervals (Kennedy and Schumacher, 1993; Gunter, 1991) for any of these statistics, estimate control chart limits (Wood et al, 1999), simulate various probability distributions, and several other things as well. In effect the program is a crunchy method for estimating the degree of variability between random samples drawn in similar circumstances. As well as the advantages of transparency (see Figure 2), this crunchy method has the advantage that the learner has only to master one software tool to cover a range of different (from a conventional viewpoint) contexts.

Spreadsheets are another useful tool for implementing crunchy methods (see the Appendix for an example; the family simulation in Figure 1 could also be implemented on a spreadsheet), and there are obviously many other possibilities.

The practical detail of the software is clearly important. Software which is difficult to use may hinder users; different interface styles (e.g. menus, commands) may have different strengths and weaknesses; the words used to describe inputs, outputs and operations are obviously important; there is a strong case for people using software with which they are familiar as much as possible. There is, however, no space to pursue these issues here, except to point out their importance.

We turn now to some practical choices. We will consider three general situations. The first is the situation where a CSS must be designed to support a given set of mathematical methods. The second is where the CSS must be designed to support problem solving in a given area for a given group of people with a given level of expertise - there is an extra element of flexibility in that different methods may be considered. The third situation is like the second, except that the expertise of the users is treated as part of the choice: what do people need to know to use the best possible approach to a given type of problem? The first is a short term problem; the second is a medium term problem in that it assumes that users can be weaned away from ingrained habits towards more appropriate methods; and the third is the long term problem in that it assumes that the training and education system can take account of the computer systems which are likely to be available.

Choice 1: support systems, given methods and user education

This the usual situation. The method is seen as given, and the CSS is designed to support this method as well as possible given the expertise of the users. The difficult aspects are likely to be phases 1 and 3 of the process (above): these are why the user's understanding of the method and its interpretation is crucial.

Usually the method to be implemented is on the clever end of the continuum. There may, on occasions be a possibility of utilising some of the user friendliness of the crunchier methods by pretending to the user that the method is in fact crunchier than it is. We may call such methods *virtual crunchy* methods.

As an example, consider the case of a computer systems for multiple regression such as those provided by the tools in many spreadsheets. In fact, multiple regression, is based on some clever mathematics and does not use an iterative method. This leaves the uneducated user (relatively speaking) with the problem of understanding precisely what multiple regression is and when it is appropriate to use it.

However it would be possible for a multiple regression CSS to start with a guess for the coefficients of the least squares model, show the user how the squares are computed, and then move on to another set of coefficients with a slightly lower sum of squares. In this way the user would get the idea of the technique as a way of searching for a least squares solution, but then, when the button is pressed to speed up the process and find the best answer, the standard algorithm would be used. The idea of searching for a best answer can be used as a metaphor to explain to the user what is happening. The justification for this is simply that the two methods are mathematically equivalent.

(In fact, regression problems can easily be solved on a spreadsheet by using the Solver or Optimiser to minimise the value of a function corresponding to the sum of squares. The real crunchy method does work in this instance. Its disadvantage over the built in regression formulae is that it takes time and some skill with spreadsheet formulae to set up; its advantages are its transparency and the fact that the method can very easily be adapted to models other than the simple linear one assumed by the built-in formulae.)

Choice 2: methods and support system, given user education

If we view the method as adjustable to suit the needs of the users and the situation, then a much wider range of possibilities opens up. Clearly, given the advantages of crunchier methods, these methods would often be the preferred ones for the computer to support, although there are counter arguments as discussed above. Each case would need to be treated on its own merits, taking account of the level of expertise of potential users.

So, for example, a suitable CSS for the very young child in the shop might use the crunchiest approach - counting pennies; similarly for the family example, the simulation method (using whatever software is convenient) would probably be more appropriate than the binomial distribution. On the other hand for users with a higher level of expertise, the cleverer methods may be more appropriate.

There may be situations where there is a justifiable fear that the provision of computer packages supporting crunchy methods may remove the incentive for users to master the more advanced concepts required by the clever methods. This is the reason why I suspect that few would support the use of computer-based counting software in primary schools. However, I do not think this argument applies to many more advanced problems for the reasons discussed in the next section.

These conclusions may be complicated by convention and expectation. In practice problems are often phrased in terms of the method or model and not the requirements of the situation; academics may want the results of an analysis of variance, and production managers may want the economic order quantity as *defined* by the standard formula (Dennis and Dennis, 1991, p. 453). In each case custom or conventional advice decrees the appropriate piece of mathematics to be used, and the situation is in effect, a choice of type 1 above. In the longer term, it may be possible to focus on the actual requirements of the situation and so move towards the more powerful choice where the method can be freely chosen.

We have mentioned above that one probable advantage of the crunchier methods is their greater generality. This factor is likely to be extremely important for CSSs simply because a CSS is of little use if users do not know of it, and in rough terms what it will do - and acquiring this expertise inevitably takes time. General methods are likely to lead to a reduction in the time users need to spend familiarising themselves with what is available, and also mean that the process of choosing an appropriate approach to a given problem is likely to be easier simply because there are fewer possibilities to choose from.

Choice 3: methods and support systems and user education: implications for the curriculum

From a long term perspective we can ask what conceptual background education needs to foster given the availability of CSSs implementing crunchy methods. Is the binomial probability distribution a sensible part of the standard curriculum of elementary statistics given the availability of simulation methods (which can be implemented on any suitable software including a spreadsheet)? Is division a sensible part of the primary school mathematics curriculum despite the crunchier methods which can be used?

My answer to the first of these questions would be no, and to the second yes - and I would suspect that I am not alone in this judgment. There are three important differences between the two situations: division is much more commonly used than the binomial distribution, people are likely to want to use it when they have no CSS to hand, and it leads on to many other ideas which non-mathematicians would expect to master, whereas the binomial distribution does lead on to further ideas (e.g. the normal distribution) but exactly what and how would usually be considered to be of interest to mathematical specialists alone.

In rough terms, if we view mathematical methods as forming a hierarchy in which the lower levels are prerequisites for an understanding of the higher levels, the general conclusion is that there is likely to be sense in using crunchy methods for the top level of the hierarchy since this does not lead on to further ideas. This conclusion holds for both the novice and the expert mathematician.

It may also be reasonably to use crunchy methods in parts of the lower levels of this hierarchy, if the concepts developed by the crunchy methods (see Property 4 above) are adequate for the higher level developments. For example, the simulation of the binomial distribution leads to the concept of a binomial probability distribution (but without the probability formulae). This gives users a label and a concept which can be basis of further theory. The normal distribution can now be viewed as a limiting case of this binomial distribution. The basic pattern of the normal distribution, and the fact that this pattern is similar to many empirical distributions, is quite clear from the simulations; the only thing missing is the formula.

The implication of this conclusion for education in mathematical methods is simply that many clever methods may not be worth learning. Each clever method would have to be considered on its merits, but my judgment is that this strategy of replacing clever methods (especially those at the highest level of the cognitive hierarchy) by a few crunchy principles would lead to a very substantial reduction in the technical content of mathematics education. However, the increased transparency of the crunchy methods may lead to *more* powerful and realistic approaches to problem solving.

An objection

When I showed an earlier version of this paper to a colleague his reaction was indignation based on the assumption that these ideas would, if taken seriously, "lower standards". Students would no longer need to grapple with "proper" mathematics such as methods for solving polynomials and mathematical probability distributions.

There were, I think, two points of difference between us. First we differed on the nature of "proper" mathematics. From my perspective the ability to formulate models was far more important than the ability to find a formula, and if some topics were discarded from the curriculum this was just a part of the inevitable change that the progress of technology brings to a rational curriculum (see, for example, Wood et al, 1997). In the words of Stewart (1990, p. 82):

> "Formula? Who cares about formulas? Those are the surface of mathematics, not the essence!"

The second difference between us was perhaps even more fundamental. My perspective is that we want to help students develop as powerful and reliable a framework as possible *with as little pain and effort as possible*. My colleague's implicit assumption was that we wanted to develop as much understanding as possible of a given curriculum. For a given problem, from the first perspective, it is generally sensible to encourage the adoption of the *easiest* method, but from the second it may be more sensible to encourage learners to use the *most difficult* method since they will then learn more. From my perspective the use of methods which are too difficult for learners to grasp easily with the time and resources available is silly as it is likely to lead to doctors failing to grasp the basics of statistics (Altman and Bland, 1991) and production managers failing to grasp essential statistical principles of quality control. From my colleague's perspective the use of these methods is necessary to "preserve standards" in academia, but not, unfortunately, in real life.

The fact that a method is easy does not mean that it is a bad method. Crunchy methods should *not* be regarded as lacking in rigour; on the contrary, their transparency means that users can see exactly what is going on and check the plausibility of any assumptions made.

Some related arguments

There are a number of other arguments about mathematics, education and computers, which I will mention here very briefly to clarify their relationship with the argument of the present paper.

Computers and calculators can perform clever methods without help from users; therefore users can concentrate on applications and interpretation, and do not need to concern themselves with technical details.

This is regarded by some as a good thing, and by others as a bad thing. It is, however, quite different from the argument of the present paper, which concerns crunchy methods which are thoroughly understood.

Crunchy methods like simulation or trial and error methods are useful for helping students develop the insights which will allow a deeper understanding of clever methods

This is doubtless true, but the argument of this paper is that it is often sensible to treat crunchy methods as an end in their own right, not as the means to another end.

Computer programming is helpful for learning mathematics

Encouraging students to program mathematical activities and processes has been found to be an effective tool for making them explicit and so helping students learn mathematics (Leron and Dubinsky, 1995; Cornu and Dubinsky, 1989; Papert, 1993). This principle could be applied to helping people understand the principles behind both crunchy and clever methods, although it may apply more naturally to crunchy methods with their typically repetitive algorithms.

However, the argument in this paper is not about learning mathematics, but about which mathematical methods are the most appropriate; about the content of the curriculum, not the methods. The computer support systems discussed above are primarily for *doing* mathematics, not for *learning* mathematics (although they may be helpful here too).

Conclusions

I have defined the crunchiness of a mathematical method in terms of:

* its transparency, and
* its repetitiveness, and
* the level of conceptual background required for its use.

Some methods may be crunchy according to some, but not all, of these criteria.

Crunchy methods are contrasted with clever ones. Crunchier methods are likely to be:

* conceptually simpler, and
* more general and so more powerful and more able to model complex phenomena.

This means that they are often preferable from the perspectives of user-friendliness, reliability and realism.

On the other hand crunchy methods:

* tend to be computationally intensive and slow, and
* may only yield an approximate or tentative answer on some occasions, and
* may in some situations be less useful for building further concepts and techniques.

Sometimes these disadvantages are important but often they are of little consequence.

It is fairly easy to devise methods which are crunchier than many conventional clever ones. Examples include simulation, resampling and bootstrapping for problems in probability and statistics; numerical methods for solving equations, optimisation and integration problems; simulation for modelling queues; and so on.

Crunchy methods are ideally suited to computer packages for supporting mathematical reasoning - particularly for non-expert users. To take advantage of the transparency of crunchy methods it is important that users can step through a method slowly to see how it works. Crunchy methods should then have the potential to enable novice users to develop useful methods for real problems and interpret their results with genuine, reliable and consistent intelligence. This contrasts strongly with the present situation where mathematical and statistical methods are frequently misunderstood, misused and their results misinterpreted.

In practice, these opportunities have been taken up very little, although there is a limited recognition of the advantages of resampling and bootstrap methods in statistics. Usually, computersupported mathematical reasoning follows the established methods. In part this is probably because users' understanding of some of the methods is inadequate to enable them to distinguish the method from the problem it solves; the problem is the method and possibilities for improvement are unrecognised. The difficulties are exacerbated by the expectations of examiners and curricula.

The final section of the paper argues that the crunchier methods offer an opportunity to simplify the mathematics curriculum very considerably. Many of the clever theorems of mathematics are simply not necessary (for practical purposes: I am not referring to the study of pure mathematics) if you have a computer to crunch out the answers. Much of the standard curriculum of mathematics applied to statistics, management science and similar disciplines is unnecessary. In its place would go a few crunchy methods implemented by suitable computer packages - e.g. one for simulating univariate probability distributions, one for optimising numerical functions, and so on - and a more thorough understanding of general principles and "lower level" concepts.

I will finish by raising a few questions prompted by these considerations. Is it possible to find a crunchier alternative to *any* mathematical method? Does the idea just apply to mathematical methods? Can we compile a short list of crunchy methods which subsume all commonly used clever ones? As is often the case, I suspect that the answers to these questions depend on a careful definition of the terms in which they are phrased.

References

Altman, D. G., & Bland, M. J. (1991). Improving doctors' understanding of statistics. J. R. Statist. Soc. A, 154(2), 223- 267.

Bailey, T., & Weal, S. (1993). Introducing undergraduates to the spirit of OR while imparting substantive skills. J. Opl Res. Soc., 44(9), 897-908.

Bajgier, S. M. (1992). The use of bootstrapping to construct limits on control charts. Proceedings of the Decision Science Institute, 1611-1613.

Cornu, B., & Dubinsky, E. (1989). Using a cognitive theory to design educational software. Education and Computing, 5, 73-80.

Dennis, T. L., & Dennis, L. B. (1991). Management Science. St. Paul: West Publishing Company.

Greenfield, T. (1993). Communicating statistics. J. R. Statist. Soc. A, 156(2), 287-297.

Gunter, B. (1991, December). Bootstrapping: how to make something from almost nothing and get statistically valid answers - Part 1: Brave New World. Quality Progress, 97-103.

Hoerl, R. W., & Palm, A. C. (1992). Discussion: integrating SPC and APC. Technometrics, 34(3), 268-272.

Kennedy, K., & Schumacher, P. (1993). The introduction of bootstrapping techniques for finding confidence intervals using simulation. Journal of Computers in Mathematics and Science Teaching, 12(1), 83-92.

Leron, U., & Dubinsky, E. (1995, March). An abstract algebra story. American Mathematical Monthly, 102, 227-242.

Mar Molinero, C. (1996, October). Data envelopment analysis - a non-mathematical introduction. OR Insight, 9(4), 22-28.

Noreen, E. W. (1989). Computer intensive methods for testing hypotheses. Chichester: Wiley.

Papert, S. (1993). Mindstorms: children, computers and powerful ideas (2nd edn). Hemel Hempstead: Harvester Wheatsheaf.

Pfannkuch, M. (1997). Statistical thinking: one statistician's perspective. in Aotearoa Conference People in Mathematics Education Proceedings, (pp. 406-413). Hamilton: MERGA Inc.
Pidd, M. (1996). Tool for thinking: modelling in management science. Chichester: Wiley.
Romero, R., Ferrer, A., Capilla, C., Zunica, L., Balasch, S., Serra, V., & Alcover, R. (1995). Teaching statistics to engineers: an innovative pedagogical experience. Journal of statistics education (http://www.stat.ncsu.edu/info/jse), 3(1).
Seppala, T., Moskowitz, H., Plante, R., & Tang, J. (1995, April). Statistical process control via the subgroup bootstrap. Journal of Quality Technology, 27(2), 139-153.
Simon, J. L. (1992). Resampling: the new statistics. Arlington, VA: Resampling Stats, Inc.
Simon, H. A. (1996). The sciences of the artificial. Cambridge, Massachusetts: MIT Press.
Skemp, R. R. (1976). Relational understanding and instrumental understanding. Mathematics Teaching, 77, pp 20-26.
Stewart, I. (1990). Does God play dice? London: Penguin.
Stuart, M. (1995). Changing the teaching of statistics. The Statistician, 44(1), 45-54.
Waldrop, M. M. (1994). Complexity: the emerging science at the edge of order and chaos. London: Penguin.
Wood, M. (1989). Expert systems, expert tutors and training in elementary statistics. in N. Shadbolt (ed), Research and Development in Expert Systems, VI, (pp. 195-206). : Cambridge University Press.
Wood, M., & Preece, D. (1992). Using quality measures: practice, problems and possibilities. International Journal of Quality and Reliability Management, 9(7), 42-53.
Wood, M., Cahill, P., & Hicks, J. (1997). Computer packages as cognitive paradigms: implications for the education of accountants. Journal of Accounting Education, 15(1), 53-69.
Wood, M., Kaye, M., & Capon, N. (1999). The use of resampling for estimating control chart limits. Journal of the Operational Research Society (50), 651-659.
Yilmaz, M. R. (1996). The challenge of teaching statistics to non-specialists. Journal of statistics education (http:// www.stat.ncsu.edu/info/jse), 4(1).

Appendix - Three methods for exploring a problem

I was recently writing a paper which puts forward a numerical model based on this equation:

$$c = (1-(1-p)^n)^v$$

c and *p* represent quantities which are determined by assumptions which can be intuitively linked to the situation being modelled (market research) - typical values would be $c = 0.8$ and $p = 0.1$. The purpose of the model is to estimate a suitable value of *n* - the size of a sample. Unfortunately *v* is unknown and cannot, in principle, be determined.

If the range of values of *n* corresponding to all "reasonable" values of *v* was "reasonably narrow", then I decided that the model would be still useful. Such a reasonable range of values for *v* included all integers from 5 to 1000. What is the corresponding range of values of *n*? There are four obvious methods for solving this.

The crunchiest approach to this question (Method 1) is to set up a spreadsheet and use trail and error:

p	**v**	**n**	**c**
0.1	5	50	0.97

The first three cells contain numerical values: the first two given by the constraints of the problem, and the third (50 for *n*) is a first guess for an appropriate value for *n*. The fourth cell contains the formula above to calculate the corresponding value of *c*. I then simply changed the vale of *n* until the value of *c* was 0.8. This process took four trials and was quick and straightforward. The answer was that *n* is 30. I then changed *v* to 1000 and repeated the process to find that *n* now has to be 80. (I decided this was a suitably narrow range.)

Method 2 was the same but using the spreadsheet (Excel) Solver to find the value of *n* corresponding to a target value of 0.8 for *c*. This arrived at the same answers but took slightly longer due to the time taken to set up the parameters of the Solver. When I tried again, starting from an initial value for *n* of 1, the computer failed to find a solution - which is one of the problems with heuristic methods.

Method 3 involves manipulating the equation and rewriting it as

$$n = log(1-c^{1/v})/log(1-p)$$

This formula can now be entered in the cell for *n*, and 0.8 entered in the cell for *c* which gives the values of *n* directly without trial and error.

Method 4 would be to use the last equation to try to deduce, in general terms, how sensitive *n* is to changes in *v*. In practice I could not see any easy way to do this, and as Method 3 had told me all I needed to know, I abandoned Method 4.

The four methods are clearly in order of increasing cleverness and decreasing crunchiness. Method 1 requires no ability to manipulate equations, and also seems the most trustworthy in that it is direct and gives very little scope for errors. It is also a perfectly rigorous method and can give an answer to any required degree of accuracy. It seems the obvious method for anyone who is not good at manipulating equations.

As Method 1 was so quick Method 2 had few advantages in this situation. With more variables, Method 2 may be preferable to Method 1 - but the Solver can never be regarded as fully reliable.

Method 3 was my preferred method. It enabled me to build up a table of values of *n* corresponding to the different values of the other variables. On the other hand it is no more rigorous than Method 1, and seemed less trustworthy in that I was not confident I had not made an error in manipulating the equation.

The main drawback of method 4 was that I was not clever enough to do it.