

How to get Meaningless Answers in Scientific Computation (and what to do about it)

L. FOX, FIMA

Oxford University Computing Laboratory



1. Introduction

THE phrase "scientific computation" essentially means the numerical solution of problems of a scientific nature, as distinct from other computer applications such as "payroll," textual analysis, and so on. I am therefore talking to scientists and engineers, the people confronted with such problems.

Now those of us who try to teach numerical mathematics in universities or other centres of learning are frequently discouraged by the discovery that those who have problems to solve just will not learn how to solve them. Introduce computer programming to scientists and they flock to learn. Introduce numerical mathematics and they disappear in droves.

The undergraduate mathematics books they read will rarely include any serious numerical studies, and the research literature they read will also be lacking in this topic. Their teachers, both at undergraduate and graduate level, will rarely have more than a smattering of rather old numerical knowledge, and I have even seen very recent books, with titles such as "numerical methods for engineers," which will contain, say, 40 pages on the matrix eigenvalue problem without a single mention of the names of Rutishauser, Givens, Householder or Wilkinson. Scientists, moreover, rarely come in contact with the numerical analysts, although the latter are frequently to blame because (a) they do not go looking for serious practical problems and (b) they insist on publishing in journals that scientists are unlikely to read and in the sort of language that scientists cannot understand.

Here are serious problems of (i) education and (ii) communication. I shall return in my summary to make a few more relevant comments, but I freely admit that I have no real answer to these problems. What I want to do in the main part of this paper is to give some very simple illustrations of what can go wrong and (occasionally) why it goes wrong, with a view to indicating the real importance of at least trying to find a solution to these problems. For computing is a very expensive business, and we should be looking at all costs for economical and accurate methods. I have little doubt that about 80 per cent. of all the results printed from the computer are in error to a much greater extent than the user would believe, and that the time taken is too large by a factor that is almost impossible to estimate.

I have given this material in lectures at various places, and even I have been impressed at the general reactions. They included remarks like "My problems are never like this," "You have deliberately picked out pathological problems," and sentiments of horrified surprise amounting sometimes to panic that none of their results

could ever possibly have any meaning at all. Well (a) these *are* "your problems," (b) all the examples (although perhaps slightly simplified) have turned up in practice and (c) far from trying to spread panic I am trying to indicate that there is something to learn, that it is interesting, worth learning and not really very difficult except perhaps at a very advanced stage.

In what follows I give, in Part A, some illustrations of "how to get rubbish." Most of these cases have appeared in previous literature, and this is indicated where relevant. In Part B I try to indicate the sort of thing that is needed to overcome these problems, with a conclusion in Part C that varies from the deepest pessimism to a few gleams of hope.

A. How to get rubbish

2. Your problem might be ill-conditioned (suffer from inherent instability)

Example 2.1 (reference 1, p. 75)

Consider the linear equations

$$\left. \begin{aligned} \frac{1}{2}x_1 + \frac{1}{3}x_2 + \frac{1}{4}x_3 + \frac{1}{5}x_4 &= b_1 + \varepsilon_1 \\ \frac{1}{3}x_1 + \frac{1}{4}x_2 + \frac{1}{5}x_3 + \frac{1}{6}x_4 &= b_2 + \varepsilon_2 \\ \frac{1}{4}x_1 + \frac{1}{5}x_2 + \frac{1}{6}x_3 + \frac{1}{7}x_4 &= b_3 + \varepsilon_3 \\ \frac{1}{5}x_1 + \frac{1}{6}x_2 + \frac{1}{7}x_3 + \frac{1}{8}x_4 &= b_4 + \varepsilon_4 \end{aligned} \right\},$$

where the coefficients are exact but the right-hand sides b_r have possible uncertainties ε_r , where $|\varepsilon_r| \leq 10^{-4}$.

Now the worst uncertainty in the solutions that the ε_r can produce is

$|\delta x_1| = 0.46$, $|\delta x_2| = 3.28$, $|\delta x_3| = 6.34$, $|\delta x_4| = 3.61$, correctly rounded to two decimal places. Then if $b_1 = -b_2 = b_3 = -b_4 = 1$, we can show that

$$x_1 = 4620 \pm 0.46, x_2 = -32820 \pm 3.28,$$

$$x_3 = 63420 \pm 6.34, x_4 = -36120 \pm 3.61,$$

and this is not an ill-conditioned situation in relative terms. But there are some b_r , of comparable size, for which the resulting problem is ill-conditioned for the determination of the x_r . For example, with

$$b_1 = 1, b_2 = 0.7227, b_3 = 0.5697, b_4 = 0.4714,$$

we find

$$x_1 = 1.162 \pm 0.46, x_2 = -0.234 \pm 3.28,$$

$$x_3 = 2.436 \pm 6.34, x_4 = -0.560 \pm 3.61,$$

and not a single significant figure in the solution is "meaningful."

It is important, however, to note that the uncertainties in the x_r are not independent, and that the determination of a number $y = \sum c_r x_r$ might not be ill-conditioned, even in the second case. For example, with $c_1 = c_2 = c_3 = c_4 = 1$, we find

$$y = 2.804 \pm 0.09,$$

the uncertainty in the figure 2.804 being no more than about 3 per cent.

Example 2.2 (see also reference 1, p. 79)

When the data are uncertain, as in the last example, I call the problem "physical," and there is clearly a limit to the number of figures in the "solution" worth quoting. In a "mathematical" problem the data are exact and any requested accuracy in the solution is *meaningful*. As we may not be able to store the data exactly, however, we may have difficulty in obtaining an accurate solution to a mathematical problem if the corresponding physical problem is ill-conditioned. For the problem of Example 2.1 we have seen that with

$b_1 = 1, b_2 = 0.7227, b_3 = 0.5697, b_4 = 0.4714$, the exact solution is

$x_1 = 1.162, x_2 = -0.234, x_3 = 2.436, x_4 = -0.560$. Now suppose that our computer can store only four decimal digits, and therefore starts with a somewhat perturbed matrix with, for example, $\frac{1}{3}$ taken as 0.3333 and $\frac{1}{7}$ as 0.1429. The correctly rounded solution of this problem (which the computer could not obtain but tries to obtain) is

$x_1 = 1.097, x_2 = 0.187, x_3 = 1.680, x_4 = -0.153$, which we observe to be quite different from that of the original problem. We should have to start with about 8 correct figures, and make no subsequent arithmetic errors, to obtain results that agree to about four figures with the exact solution of the given problem.

Example 2.3 (reference 1, p. 30)

Here is a case of ill-conditioning relating to a recurrence system.

If $y_{r+1} = 0.75 - 0.75(r+4)y_r$, with $y_0 = 0.1957 \pm 0.0001$, that is with y_0 lying somewhere in the interval $0.1956 \leq y_0 \leq 0.1958$, then all we can say for y_7 , for example, is that it lies in the interval

$$10.99 \geq y_7 \geq -4.78,$$

a conclusion that will rarely be of significant value.

Example 2.4 (reference 2, p. 43)

The final example is an ill-conditioned situation relevant to finding zeros of polynomials. The polynomial

$$p_{20}(x) = (x-1)(x-2) \dots (x-20) \\ = x^{20} - 210x^{19} + \dots + 20!$$

has the integers 1, 2, ..., 20 as its zeros. Suppose that we change just one coefficient slightly, that of x^{19} from -210 to $-(210 + 2^{-23})$. This is a small absolute change, about 10^{-7} , and a *very small* relative change which many computers could not make! What effect does it have on the zeros? It turns out that the small zeros are changed by only small amounts, but the pair which were 16 and 17 now change to the complex pair

$$x_{16}, x_{17} = 16.73 \dots \pm i 2.81 \dots$$

The change is both substantial and alarming!

3. Your method might be unstable

(suffer from induced instability)

So far we have talked about problems that are inherently unstable (ill-conditioned); the uncertainties in the solution depend only on the *problem*. We turn now to illustrations of poor (unstable) *methods*, which will give poor results even for well-conditioned problems.

Example 3.1 (reference 1, p. 39)

Consider the boundary-value recurrence problem

$y_{r+1} - 10.1 y_r + y_{r-1} = -1.35r, y_0 = 0, y_5 = \frac{5}{6}$, of which the exact solution is $y_r = \frac{1}{6}r$.

In the so-called "shooting method" we first find $y_r^{(1)}$ from

$$y_{r+1}^{(1)} - 10.1 y_r^{(1)} + y_{r-1}^{(1)} = -1.35r, \\ y_0^{(1)} = 0, y_1^{(1)} = 0,$$

and $y_r^{(2)}$ from

$$y_{r+1}^{(2)} - 10.1 y_r^{(2)} + y_{r-1}^{(2)} = 0, \\ y_0^{(2)} = 0, y_1^{(2)} = 1.$$

We then observe that $y_r^{(1)} + \lambda y_r^{(2)}$ satisfies for any λ the given recurrence relation and the given condition at $r = 0$, and that it satisfies the condition at $r = 5$ if λ is chosen so that

$$y_5^{(1)} + \lambda y_5^{(2)} = \frac{5}{6}.$$

With four-digit floating-point arithmetic, the results of this computation produce the surprisingly inaccurate values

r	0	1	2	3	4	5
y_r	0	0.1666	0.3330	0.4900	0.7000	1.000

Not even the computed y_5 is correct, on which we lavished special attention!

The *problem*, incidentally, is very *well-conditioned*, but our *method* is *unstable*.

Example 3.2 (reference 3, p. 321)

Consider the eigenvalue-vector problem defined by the linear equations

$$(10 - \lambda)x_1 + x_2 = 0 \\ x_1 + (9 - \lambda)x_2 + x_3 = 0$$

$$x_{19} + (-9 - \lambda)x_{20} + x_{21} = 0 \\ x_{20} + (-10 - \lambda)x_{21} = 0$$

The largest eigenvalue is $\lambda = 10.74619 \ 41829 \dots$, and there are many methods for determining it quite accurately. An apparently attractive way of finding the corresponding eigenvector is to put this λ into the equations, take $x_1 = 1$ (which *normalises* the eigenvector), compute $x_2, x_3, \dots, x_{21}$ directly from the first, second, ..., and penultimate equations, with the knowledge that if all the arithmetic is exact then the last equation is automatically satisfied.

If we take the very good approximation $\bar{\lambda} = 10.74619420$, and make no further mistakes in the arithmetic, we produce a vector of which some of the components are

r	1	4	7	12	15	21
x_r	1	10^{-1}	10^{-4}	0.4	10^3	10^{10}

The correct vector has the quite different shape

r	1	4	7	12		21
x_r	1	10^{-1}	10^{-3}	10^{-8}		10^{-20}

Again the method is unstable, and again this problem is quite well-conditioned.

The similar method working backwards, taking $x_{21} = 1$ and omitting the very first equation, does in fact work perfectly well, but this cannot be guaranteed in general!

The last two examples were algebraic, but we also have instabilities in the approximation of "continuous" problems by "discrete" methods, as the following three examples show.

Example 3.3 (reference 1, p. 229)

Consider the simultaneous differential equations

$$y' = -40.2y + 19.6z + \frac{\cos x}{1+x} - \frac{\sin x}{(1+x)^2} + \frac{40.2 \sin x}{1+x}$$

$$z' = 19.6y - 10.8z - \frac{19.6 \sin x}{1+x}$$

With $y(0) = 3, z(0) = 1$, the exact solution is

$$y = e^{-x} + 2e^{-50x} + \frac{\sin x}{1+x}, z = 2e^{-x} - e^{-50x}.$$

One would expect that for an x for which e^{-50x} is negligible one could use, in an approximate step-by-step method, a reasonably large interval conditioned only by the necessity for good representation of the important terms e^{-x} and $\sin x$. For $x > 0.6$ the third differences of the solution are indeed negligible to three decimals at interval $\delta x = h = 0.1$, and the local truncation error of a fourth-order Runge-Kutta method is also negligible. But the usual method fails completely, producing the results

Runge-	x	0.6	0.7	0.8	0.9	1.0
Kutta	y	0.902	0.755	-0.809	-21.9	-310.7
method	z	1.098	1.083	1.727	12.18	156.5

compared with the correct values

correct	y	0.902	0.877	0.849	0.819	0.789
	z	1.098	0.993	0.899	0.812	0.736

Example 3.4 (reference 1, p. 231)

The equation

$$y' = -y^2/(x+1), \text{ with } y(0) = 1,$$

has the harmless-looking solution $y = \{ \log(1+x) + 1 \}^{-1}$. In fact at interval 0.1, for $x > 0.3$, the fourth differences are almost negligible in a four-decimal solution. The formula

$$11y_{r+1} + 27y_r - 27y_{r-1} - 11y_{r-2} = 3h(y'_{r+1} + 9y'_r + 9y'_{r-1} + y'_{r-2}),$$

which is extremely accurate and can be used (after a separate start) in a step-by-step process, produces results with the four-figure errors

x	0.3	0.4	0.5	0.6	0.7	0.8
$(h = 0.1)$	0	0	-2	+6	-19	+59
$(h = 0.05)$	1	7	76	765		

Strikingly, the smaller the interval the greater the error!

Example 3.5 (reference 1, p. 217)

The solution of $y'' + y = 0, y(0) = 0, y'(0) = 1$, is $y = \sin x$. The formula $y_{r+1} - (2-h^2)y_r + y_{r-1} = 0$ has local error $O(h^4)$, and with correct $y(0)$ and $y'(0.1)$ it produces at interval 0.1 results which as far as $x = 1$ have errors of only one or two units in the fourth decimal place.

The formula

$y_{r+2} - 16y_{r+1} + (30 - 12h^2)y_r - 16y_{r-1} + y_{r-2} = 0$ has a very small local error of $O(h^6)$, but starting with the correct values at $x = 0, 0.1, 0.2$ and 0.3 it makes increasingly large errors and at $x = 0.7$ produces as an approximation to $\sin x$ the remarkable value of 3.9330 (!).

In both the last two examples the praiseworthy desire to reduce the local truncation error produced catastrophic global results.

The theme of induced instability is extended, with variations, in the next few sections. All the examples illustrate induced instability, but I have tried to separate out some possible reasons for the selection of these inaccurate methods.

4. You expect too much "analysis" from the computer

The next point I want to make is that it is dangerous to depend on the computer to tell you when to stop in an infinite sequence of operations. This mistake is perhaps stimulated by computer programming manuals, many readers imagining that coding examples always include good numerical analysis!

Example 4.1 (reference 1, p. 10)

In the summation of the series

$$S = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots,$$

it is perfectly correct to stop when the next term is smaller than the tolerable error, because the truncation error is smaller than the first neglected term.

But this "stopping" decision would fail catastrophically for the apparently similar-looking series

$$S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots,$$

whose sum is infinite!

Example 4.2 (see also reference 1, p. 97)

Consider the linear equations

$$\begin{cases} 0.5x_1 - 0.4x_2 = 0.32 \\ -0.4x_1 + 0.5x_2 = -0.22 \end{cases},$$

whose exact solution is $x_1 = 0.8, x_2 = 0.2$, and try for solution the iterative method

$$x^{(r+1)} = \begin{bmatrix} 0.5 & 0.4 \\ 0.4 & 0.5 \end{bmatrix} x^{(r)} + \begin{bmatrix} 0.32 \\ -0.22 \end{bmatrix}.$$

Starting with the initial guess $x_1^{(0)} = 0.7, x_2^{(0)} = 0.1$, we find after some steps the results (rounded to five decimals)

$x^{(6)}$	0.74686	0.14686
$x^{(7)}$	0.75217	0.15217
$x^{(8)}$	0.75695	0.15695

Now the unskilled, seeking a result correct to two decimal places, might well ask the machine to stop iterating and print out the answer $x^{(7)}$, just because the next correction is less than 0.005. The resulting error is quite large, and in terms of summing an infinite series (which is what we are doing in disguised form), the truncation error is correspondingly quite large.

(The corresponding example in reference 1 is not quite satisfactory, as the problem is rather ill-conditioned. The problem discussed here is quite well-conditioned.)

5. Your intuition fails you

Next we examine some cases in which what one might expect to happen fails to happen.

Example 5.1 (reference 1, pages 134 and 175)

One would reasonably expect that if we are given a function $f(x)$, in $-1 \leq x \leq 1$, and if we compute polynomial approximations $p_n(x)$ (of degree n), which agree exactly with $f(x)$ at $n+1$ equidistant points in $-1 \leq x \leq 1$, then these polynomials are increasingly good approximations at all points in $-1 \leq x \leq 1$. This fails for the function $f(x) = (1+x^2)^{-1}$, the polynomial approximations (except, of course, at the matching points) becoming poorer and poorer in the approximate range $0.7 \leq |x| < 1$.

Integration of such polynomials, producing the so-called Newton-Cotes integration formula, may also fail. For example, successive formulae of orders 2, 4, 6, 8, 10 produce the remarkable results (rounded to three decimals)

$\int_{-4}^4 (1+x^2)^{-1} dx = 5.490, 2.278, 3.329, 1.941, 3.596,$
which oscillate with increasing amplitude about the true
value $2 \tan^{-1} 4 = 2.6516 \dots$

Example 5.2 (reference 1, p. 190)

Successive trapezoidal-rule approximations for this
last integral produce the rounded values

h	4	2	1	$\frac{1}{2}$
$T(h)$	4.2352	2.9176	2.6588	2.6505

One might reasonably expect that the true value is
smaller than 2.6505, as the first four values decrease
monotonically. But $T(\frac{1}{4}) = 2.6514$, and the correct value
is 2.6516 $\dots$

6. You accept consistency too easily

Another source of error is the acceptance of consistent
results, mistaking consistency (at too early a stage) for
accuracy.

Example 6.1 (reference 4, p. 199)

For the integral

$$I = \frac{1}{2} \int_{-1}^1 (0.92 \cosh x - \cos x) dx = 0.2397141 \dots,$$

successive Simpson-rule estimates, with $h = 1$ and $h = \frac{1}{2}$,
are

$$S(1) = 0.239778, S(\frac{1}{2}) = 0.239777.$$

These effectively agree to six figures, although the last
two are incorrect.

Example 6.2 (reference 4, p. 199)

For the integral

$$I = \int_{-1}^1 (x^4 + x^2 + 0.9)^{-1} dx = 1.582233 \dots,$$

the three and four-point Gauss integration formulae
produce the approximations

$$G_3 = 1.585026, G_4 = 1.585060.$$

These effectively agree to four decimal places, but only
the first two are correct.

In both these examples the machine "stopping"
criterion would give quite wrong answers.

7. A successful method may fail in slightly different circumstances

Next we illustrate the fact that it is dangerous to use
a method which is successful in some circumstances but
which might fail in other and apparently not very
different situations.

Example 7.1 (reference 1, p. 150)

The "automatic" Neville interpolation method (for
finding $f(x)$ for a particular x), applied for the com-
putation at $x = \frac{1}{2}$ of the function $(1+x^2)^{-1}$, tabulated
at integer values of x , produces the array

x_j	$0.5 - x_j$	$f(x_j)$
0	0.5	1.000
		0.750
1	-0.5	0.500
		0.500
		0.800
-1	1.5	0.500
		0.575
		0.856
		0.350
		0.575
		0.814
2	-1.5	0.200
		0.575
		0.603
		0.849
		0.200
		0.462
		0.603
-2	2.5	0.200

The idea is that the values along the leading diagonal
will converge to the true value (here 0.8). There is no
sign of convergence. The method fails, but so do all
methods at this interval!

Note, incidentally, some examples of consistent
inaccuracy along the next sloping line(!), which is the
Neville result for $f(\frac{1}{2})$ without the use of the point $x = 0$.

Example 7.2 (reference 1, p. 152)

The books say that Neville's method can also be used
for inverse interpolation (the computation of x for which
 $f(x)$ has a specific value) just by interchanging the rôles
of x and $f(x)$. For the zero near $x = 0.5500$ of the
quartic polynomial $\sqrt{80}x^4 + 0.2x - 0.7071$ the Neville
array looks like

$f(x_j)$	$0-f(x_j)$	x_j			
-0.3981	0.3981	0.4			
			0.4821		
0.5721	-0.5721	0.6		0.5898	
			0.4132		0.5726
-0.6528	0.6528	0.2		0.4378	-0.2572
			0.3039		1.2165
3.1165	-3.1165	0.8		2.1793	

There is again no sign of convergence, but here the fault
is with Neville's method. There are quite good alter-
native methods, using interval 0.2 in x , for performing
accurate inverse interpolation with this function.

Example 7.3 (reference 1, pages 179 and 184)

Next we consider the method of Romberg integration,
which effectively says that the approximation $T(h)$
produced by the trapezoidal rule at interval h is related
to the true value I through the formula

$$I - T(h) = Ah^2 + Bh^4 + Ch^6 + \dots,$$

where A, B and C are constants. By taking $h = 1, \frac{1}{2}, \frac{1}{4} \dots$,
we can combine results easily to eliminate successive
 $A, B, C, \dots$ and hence obtain very good results with a
small amount of extra computation.

A successful Romberg table, for the integral of Ex-
ample (6.1), would appear as

$T(2)$	0.879333			
		0.239777		
$T(1)$	0.399666		0.239778	
		0.239778		0.239714
$T(\frac{1}{2})$	0.279750		0.239715	
		0.239719		0.239714
$T(\frac{1}{4})$	0.249727		0.239714	
		0.239714		
$T(\frac{1}{8})$	0.242217			

(Note the *inaccurate* consistency in the second column,
mentioned earlier, and the later accurate consistency.)

For the integral $-\int_0^1 x^{\frac{1}{2}} \ln x dx = 4/9$, the Romberg
array appears as

$T(1)$	0			
		0.3267		
$T(\frac{1}{2})$	0.2450		0.4004	
		0.3958		0.4268
$T(\frac{1}{4})$	0.3581		0.4267	0.4376
		0.4248		0.4376
$T(\frac{1}{8})$	0.4081		0.4374	
		0.4366		
$T(\frac{1}{16})$	0.4295			

If it is converging down the main diagonal the conver-
gence is very slow. In fact the Romberg method here

fails because, due to the infinity of the first derivative of the integrand at $x = 0$, the stated form of the error $I - T(h)$ is not here true.

Example 7.4 (reference 1, pages 12 and 13)

For a more striking example, consider the integral equation

$$f(x) = g(x) + \int_0^1 k(x, y) f(y) dy.$$

By expressing the integral, say by the trapezoidal rule, as the linear combination $h \sum a_r k(x, y_r) f(y_r)$, and then giving x in turn the values of y_r in this combination, we obtain a set of linear algebraic equations for the determination of the $f(x_r)$. As h is taken smaller and smaller the results get better and better if $k(x, y)$ is sufficiently well-behaved.

Now suppose the term $f(x)$ on the left of the equation is replaced by zero. Computationally the problem has apparently changed very little, in the sense that we can produce the linear equations and solve them as before.

This fails catastrophically! For example, a solution of the equation

$$0 = \frac{1}{3} \{ x^3 - (1+x^2)^{3/2} \} + \int_0^1 (x^2 + y^2)^{1/2} f(y) dy$$

is $f(x) = x$. Using Simpson's rule, at successive intervals $\frac{1}{4}, \frac{1}{8}$, we obtain the remarkable results

$f(0)$	$f(\frac{1}{8})$	$f(\frac{1}{4})$	$f(\frac{3}{8})$	
0.0448		0.2124		
0.0216	0.1098	0.2360	0.5117	
$f(\frac{1}{2})$	$f(\frac{5}{8})$	$f(\frac{3}{4})$	$f(\frac{7}{8})$	$f(1)$
0.5992		0.7185		1.0329
-0.3507	1.3618	-0.7614	1.2992	0.6004

The *mathematical* properties of these two problems are in fact quite different.

8. Your test examples may be too special

In testing a method it is important that the chosen examples really provide a searching test. The following examples, the second of which was sent to me a year or so ago by a somewhat agitated worker, fail in this respect.

Example 8.1

The eigenvalue problem

$$y'' + \lambda y = 0, \quad y(0) = 0, \quad y(1) = 0,$$

has the eigenvalues $\lambda_n = n^2 \pi^2$ and corresponding eigenfunctions $y_n = \sin n\pi x$, $n = 1, 2, \dots$. If we use finite differences at interval h , with the approximation

$$h^2 y''_r = y_{r+1} - 2y_r + y_{r-1},$$

we approximate the problem by a linear algebraic eigenvalue problem. With interval $h = k^{-1}$ there are $k - 1$ approximate eigenfunctions, and at any interval these approximations agree exactly with $y_n = \sin n\pi x$, $n = 1, 2, \dots, k - 1$. The approximate eigenvalues do not agree with $\lambda_n = n^2 \pi^2$.

We cannot infer from this remarkable result that the same phenomena will happen for other linear differential eigenvalue problems!

Example 8.2

The hyperbolic equation

$$\frac{\partial^2 f}{\partial y^2} = \frac{\partial^2 f}{\partial x^2}, \quad 0 \leq x \leq 1, \quad y > 0,$$

with initial conditions

$$f(x, 0) = \sin 2\pi x, \quad \frac{\partial f}{\partial y}(x, 0) = 0,$$

and periodic boundary conditions

$$f(0, y) = f(1, y),$$

has the solution

$$f(x, y) = \sin 2\pi x \cos 2\pi y.$$

With the notation $f_{r+n, s+m}$ specifying $f(x_r + n\delta x, y_s + m\delta y)$, we have either the *explicit* scheme

$$\frac{f_{r, s+1} - 2f_{r, s} + f_{r, s-1}}{(\delta y)^2} = \frac{f_{r+1, s} - 2f_{r, s} + f_{r-1, s}}{(\delta x)^2}$$

or the *implicit* scheme

$$\frac{f_{r, s+1} - 2f_{r, s} + f_{r, s-1}}{(\delta y)^2} = \frac{\delta_x^2}{(\delta x)^2} \{ \theta f_{r, s+1} + (1-2\theta) f_{r, s} + \theta f_{r, s-1} \},$$

where δ_x^2 is the operation of forming the second difference in the x direction. With suitable starting methods either of these can be used to produce values along the " $s + 1$ " line in the y direction.

The explicit scheme is stable for $\delta y \leq \delta x$, and the implicit scheme is unconditionally stable if $\theta \leq \frac{1}{4}$. The truncation error of both formulae is of the same order in powers of δx and δy .

My correspondent was extremely surprised to find that with quite large values of $\delta y = \delta x$ the results of the explicit method were very good, while with the implicit method we need very small δx and δy (even if $\delta x = \delta y$) to obtain good accuracy.

Normally, of course, the implicit method is preferable, but for this particular example the explicit method, with $\delta y = \delta x$, is such that the local truncation error is identically zero, with complete cancellation of the contributions from the x and y directions. The individual terms are quite large unless δy and δx are quite small, and this is why the implicit method, for which cancellation does not take place unless $\theta = 0$ (which reduces to the explicit method), needs small intervals for good accuracy.

B. What to do about it

9. Avoiding inherent instability (ill-conditioning)

We can sometimes avoid inherent instability, in either a mathematical or physical problem, by a suitable *reformulation* of the problem. In Example 2.3, for illustration, suppose the recurrence relation had been deduced from the integrals

$$y_r = \int_0^1 x^{r+3} e^{(4/3)(x-1)} dx, \quad r = 0, 1, 2, \dots,$$

with initial condition $y_0 = \int_0^1 e^{(4/3)(x-1)} x^3 dx = \frac{1}{128} (243e^{-4/3} - 39) = 0.1957 \dots$. To obtain good results for this mathematical problem, say for $r = 6$ and above, we should have to use a very accurate starting value of y_0 , since as we have seen the results are very sensitive to the (rounding) error in the computer-stored value of y_0 .

On the other hand, the recurrence relation with the condition $y_r \rightarrow 0$, as $r \rightarrow \infty$, is another and theoretically *exactly equivalent* formulation. But now if we take $y_n = 0$, for large n , and recur backwards with

$$y_r = \frac{0.75 - y_{r+1}}{0.75(r+4)},$$

we get very good answers, relatively insensitive, for $r < n$, to small errors in y_n . This is demonstrated in the following results obtained with this backward recurrence

(reference 1, p. 31).

y_6	0.1700	0.2200	0
y_5	0.0859	0.0785	0.1111
y_4	0.1107	0.1119	0.1065
y_3	0.1218	0.1215	0.1226
y_2	0.1396	0.1397	0.1394
y_1	0.1628	0.1628	0.1628
y_0	0.1957	0.1957	0.1957

Again, the linear equations of the type mentioned in Example 2.1 may have been produced as a result of an attempt to find a "least squares" polynomial approxi-

mation $y = \sum_{r=0}^n a_r x^r$, when y is specified (measured) at

equal intervals in a finite range $0 \leq x \leq 1$. To achieve reasonable results, for even small integers n , we have to compute the normal equations very accurately and solve them accurately. If we seek the approximation in the

form $y = \sum_{r=0}^n c_r \phi_r(x)$, where $\phi_r(x)$ is a polynomial of

degree r deliberately chosen so that the corresponding normal matrix is diagonal, then the results are good with no particular care with the arithmetic (reference 1, p. 117).

Finally, Example 2.4 illustrates the fact that polynomials are often very ill-conditioned with respect to the determination of their zeros, and it therefore pays, where possible, not to formulate a given problem in this way. For the determination of the eigenvalues of a matrix A , for example, we do *not*, in any practical case, expand $\det(A - \lambda I)$ as a polynomial and then compute its zeros.

If we cannot reformulate the problem then our method *must* provide at least some information about its degree of ill-conditioning.

10. Avoiding induced instability

One can here say very little except "do not use unstable methods." Commenting briefly on our illustrations, the methods of Examples 3.1 and 3.2 fail because we are solving boundary-value problems by initial-value methods, and the corresponding initial-value problems are inherently unstable. Example 3.3 fails essentially because we truncate a Taylor's series too early with the selected interval and with this selected method: it will succeed if the interval is sufficiently small. Examples 3.4 and 3.5 fail because, in approximating continuous problems by discrete problems, we introduce "spurious" solutions which swamp the required solution. The methods fail at all intervals.

In the examples of section 4 we are again truncating a series too quickly and failing to determine its rate of convergence. For the series of type 4.1 we need some mathematical analysis, and in 4.2 we can find not only the rate by examining successive approximations but can do something useful about it, accelerating the rate of convergence.

In the examples of section 5 the phenomena are concerned with the more difficult concept that the function concerned, although well-behaved for real x , has adjacent singularities in the complex plane.

In section 6 the temptation to accept inaccurate consistency could be avoided by using finite-difference methods, which have tended (in my view incorrectly and dangerously) to disappear from automatic computation. The same remarks apply to the first two examples of section 7. In Example 7.3 the observable misbehaviour of the integrand at one end of the range of integration indicates that the "standard" methods must be examined and analysed carefully. Example 7.4 is not at all obvious, and is not easy to explain in a few words.

Examples 8.1 and 8.2 indicate that when something surprising happens (especially if it is pleasant!) one should try to find out the reasons. The cases may be particular and not general. Here the proper use of finite-difference equations again explains the phenomena.

C. But who does it?

The reader, be he unskilled in numerical methods, may remark testily that the present writer has a degree in mathematics and 30 years' study of and experience in numerical methods. How, he will ask, is the beginner to acquire this "know-how"? And a very good question it is, with no easy answer.

There is, in my view, no doubt that the man with problems to solve must learn *something* about numerical methods. The merest tyro would look askance at the term-by-term addition of the first n integers instead of the use of the obvious formula $s = \frac{1}{2}n(n+1)$. Extending this, I believe that all scientists and engineers with numerical problems, who want to do their own numerical work, should read and understand enough to be able to analyse and appreciate most of the examples in this paper. With this knowledge they are equipped to perform competently with everyday problems, and to know for what more difficult problems they might usefully seek advice from "experts."

This surely is not too much to ask. Reference 1, for example (with apologies for the apparent advertisement), which has all this and more, is not very long (250 pages) and involves no very advanced mathematics. This, or something like it, is perhaps best studied at the late undergraduate level and time, and this sort of mathematics should appear somewhere in the syllabus of all scientific and engineering departments, even if some more-classical mathematics has to be displaced in consequence.

Armed with this knowledge, the scientist is also well-equipped to use the library of computer programs and routines available in many computer centres. It is good to see that these libraries are being improved, both in quantity, quality and documentation and annotation. But even the understanding of the documentation, and the making of the right program choice for a particular problem, requires the kind of knowledge I have mentioned. And, of course, few problems are soluble solely with the use of existing programs.

In addition to producing library programs, computer service teams must be able to provide everyday bread-and-butter advice to clients, and there should be real experts to help with the more difficult problems. One can find scores of people able and willing to help with the programming and coding, but those who can help even with everyday numerical problems are rare birds indeed. Rare, also, are the experts. All I can suggest here, with respect to the first class, is that we need more mathe-

maticians and scientists who are prepared to take the relevant courses (for example 1-year MSc courses) and then to do a service job. For the second, we need more people to study numerical analysis in depth, at a research level, and particularly with respect to practical problems rather than abstract "model" situations.

It is hoped that this sort of thing will be discussed in the IMA Conference on "The teaching of numerical analysis" (to be held at the University of Sussex, Brighton, January 3rd to 5th, 1972). The IMA and other organisations do and should continue to organise meetings of an expository nature, particularly for those who are involved with numerical computing but who are beyond the stage of formal university-type course work at undergraduate or early postgraduate stage. In this respect suggestions for topics from the relevant workers would be very useful indeed.

Finally, I used earlier a qualifying phrase about those "who want to do their own numerical work." Most people employ plumbers, electricians and glaziers to do all but the most trivial work of the relevant nature in their own houses. The ideal situation, I suppose, is that in a similar way almost all computing problems should be handed over to experts, who should be involved even (perhaps *especially*) in the problem formulation stage. In fact probably the most successful computing groups with which I have been involved were the war-time Admiralty Computing Service at Bath, and its successor the Mathematics Division in its early days at the National Physical Laboratory. Problems were submitted from

all over the country, the modern digital computer had not yet appeared, and for these problems the *methods* were decided by experts and the *arithmetic* (handle-turning variety) was performed by very competent juniors.

The high-speed computer has revolutionised this situation, but in a most unsatisfactory way. It is true that far more computing is being done, and there are not enough experts to cope with all but a small fraction of it. But it seems to be the general impression that this unsatisfactory situation is after all in some way inevitable and therefore satisfactory, that either the computer can do *both* the work of the expert and also of his assistant, or that the former is largely redundant because the machine, with its speed, size and versatility, will produce good results even if the methods are not as good as they might be.

I have tried to show in this paper that this conclusion is false, and that we ought, somehow or other, to bring back the old expertise into modern scientific computation.

References

1. Fox, L., and Mayers, D. F., "Computing methods for scientists and engineers," Clarendon Press, Oxford, 1968.
2. Wilkinson, J. H., "Rounding errors in algebraic processes," HMSO, London, 1963.
3. Wilkinson, J.H., "The algebraic eigenvalue problem," Clarendon Press, Oxford, 1965.
4. Clenshaw, C. W., and Curtis, A. R., "A method for numerical integration in an automatic computer," *Num. Maths*, 1960, **2**, 197-205.