4. J. Stevens and Y.-W. Stevens, in *Hormonally Responsive Tumors*, V. P. Hollander, Ed. (Academic Press, Orlando, FL, 1985), pp. 156–182.
5. E. B. Thompson, J. M. Harmon, M. R. Norman, T. J. Schmidt, in *Hormones and Cancer*, S. Iacobelli *et al.*, Eds. (Raven, New York, 1980), pp. 89–98.
6. N. Kaiser, R. J. Milholland, F. Rosen, *Cancer Res.* **34**, 621 (1974).
7. M. L. Nicolson and D. A. Young, *ibid.* **38**, 3673 (1978).
8. M. Lippman and R. Barr, *J. Immunol.* **118**, 1977 (1977).
9. K. A. Smith, G. R. Crabtree, S. J. Kennedy, A. U. Munck, *Nature (London)* **267**, 523 (1977).
10. E. B. Thompson, *Cancer Treat. Rep.* **63**, 189 (1979).
11. G. Crabtree, K. Smith, A. Munck, *Cancer Res.* **38**, 4268 (1978).
12. D. Garman and M. S. Center, *Biochem. Biophys. Res. Commun.* **105**, 157 (1982).
13. E. J. Peck, J. Burgner, J. Clark, *Biochemistry* **12**, 4596 (1973).
14. A. Allera and G. S. Rao, *Adv. Exp. Med. Biol.* **196**, 53 (1986).
15. D. Bression, M. Michard, M. L. Dafniet, P. Pagesy, F. Paillon, *Endocrinology* **119**, 1048 (1986).
16. M. Huet-Minkowiski, J. C. Gasson, S. Bourgeois, *Cancer Res.* **41**, 4540 (1981).
17. D. J. Gruol, N. F. Campbell, S. Bourgeois, *J. Biol. Chem.* **261**, 4909 (1986).
18. C. M. Szego and R. J. Pietras, in *Biochemical Action of Hormones*, G. Litwack, Ed. (Academic Press, New York, 1981), vol. 8, chap. 6, pp. 307–463.
19. M. E. Fant, R. D. Harbison, R. W. Harrison, *J. Biol. Chem.* **254**, 6218 (1979).
20. T. Suyemitsu and H. Terrayame, *Endocrinology* **96**, 1499 (1975).
21. R. J. Pietras and C. M. Szego, *J. Steroid Biochem.* **11**, 1477 (1979).
22. G. S. Johnson and C. J. Jaworski, *Mol. Pharmacol.* **23**, 648 (1986).
23. M. L. Elks, V. C. Manganiello, M. Vaughan, *J. Biol. Chem.* **258**, 8582 (1983).
24. S. E. Sadler and J. L. Maller, *ibid.* **257**, 355 (1982).
25. S. S. Simons, Jr., M. Pons, D. F. Johnson, *J. Org. Chem.* **45**, 3084 (1980).
26. E. Lazarides and K. Weber, *Proc. Natl. Acad. Sci. U.S.A.* **71**, 2268 (1974).
27. B. Gametchu and R. W. Harrison, *Endocrinology* **114**, 274 (1984).
28. W. J. King and G. L. Greene, *Nature (London)* **307**, 745 (1984).
29. T. Antakley and H. J. Eisen, *Endocrinology* **115**, 1984 (1984).
30. M. Perrot-Applanat, F. Logeat, M. T. Groyer-Pickard, E. Milgrem, *ibid.* **116**, 1473 (1985).
31. H. S. Pouslen, *Eur. J. Cancer* **17**, 1495 (1981).
32. W. A. Knight, R. B. Livingston, E. J. Gregory, W. L. McGuire, *Cancer Res.* **37**, 4669 (1977).
33. B. Gametchu *et al.*, in preparation.
34. R. R. Porter, *Biochem. J.* **46**, 479 (1950).
35. V. B. Singh and V. K. Moudgil, *J. Biol. Chem.* **260**, 3684 (1985).
36. V. Buonassisi, G. Sato, A. I. Cohen, *Proc. Natl. Acad. Sci. U.S.A.* **48**, 1184 (1962).
37. W. B. Pratt and L. Arranow, *J. Biol. Chem.* **241**, 5233 (1966).
38. E. V. Jensen, M. Numata, P. I. Brecher, E. R. DeSombre, *Biochem. Soc. Symp.* **32**, 133 (1971).
39. K. Fuxe *et al.*, *Endocrinology* **117**, 1803 (1985).
40. M. G. Govindan, *Exp. Cell Res.* **127**, 293 (1980).
41. A. C. Towle and P. Y. Sze, *J. Steroid Biochem.* **18**, 135 (1983).
42. B. Gametchu, unpublished observations.
43. M. Huet-Minkowiski, J. C. Gasson, S. Bourgeois, in *Drug and Hormone Resistance in Neoplasia: Basic Concepts*, N. Bruchovsky and J. H. Goldine, Eds. (CRC Press, Boca Raton, FL, 1982), vol. 1, pp. 79–94.
44. J. Stevens, Y-W. Stevens, H. Haubenstock, in *Biochemical Action of Hormones*, G. Litwack, Ed. (Academic Press, New York, 1983), vol. 10, pp. 383–446.
45. J. C. Gasson and S. Bourgeois, *J. Cell Biol.* **96**, 409 (1983).
46. Supported in part by PHS grant CA17701 (National Cancer Institute). I thank C. Watson, A. Kurosky, P. Weigel, and D. Tindall for critically reviewing this manuscript.

29 August 1986; accepted 20 January 1987

Technical Comments

Solving Linear Equations

For some time I have been disturbed by the reporting of applied mathematics in *Science*. However, the article "Solving linear systems faster" by Gina Kolata (*1*) contains so many misleading statements about a field I know well that I feel obliged to respond.

The thesis of the article is contained in its heading: "A new method [developed by Victor Pan and John Reif] that exploits parallel computations promises to have a real impact on practical problems." At the very least this statement is a gross exaggeration, and to support it the article presents a distorted view of algorithms and computing practices in the real world, not to mention of Pan and Reif's work. To give my comments focus, I will begin with a summary of the article.

1) The computer solution of linear systems is a problem with important applications in many areas—weather prediction for one. There are two classes of methods for solving such systems: direct and iterative. Of the former, the most commonly used method, called Gaussian elimination, has the drawback that it is inherently sequential and therefore not amenable to parallel implementation. Moreover, Gaussian elimination is unstable in the sense that rounding error can cause it to produce inaccurate answers. Consequently, most people use an iterative method to improve the solution obtained by Gaussian elimination.

2) Iterative methods are "stable, efficient, and amenable to parallel processing." They start with an approximate inverse of the matrix of the system and converge to the true inverse, after which the system can be solved by multiplying the right-hand side of the system of the inverse. The chief drawback to these methods is that they require an approximate inverse to start the iteration.

3) The first such iterative method was published by Schultz in 1933 and has been rediscovered several times since—by Pan and Reif among others. However, Pan and Reif also discovered how to obtain a starting approximation, thus "solving a problem that had been stumping researchers for 50 years." Pan and Reif also observed that by using about n^3 processors (which in theory can be reduced to about $n^{2.5}$) they could parallelize the algorithm so that it would be faster than previously proposed stable algorithms. Moreover, their method can be made more efficient for the very important class of sparse systems, in which most of the coefficients are zero. Although current parallel computers have too few processors to realize the full potential of the algorithm, bigger ones are in the offing, and according to Ronald Rivest of the Massachusetts Institute of Technology "the possibilities look very exciting."

This, I believe, is a fair summary of what is in the article. I have stuck to the words and quotations in the article itself and tried to avoid interpolating material from Pan and Reif's paper.

To put the matter of applications in perspective, it should be noted that the equations of weather forecasting, like those of many other applications, are not linear algebraic equations but nonlinear partial differential equations. Although the solution of linear systems sometimes plays an important role in solving such equations, it is only part of the total computation. For this reason speedups in the solution of linear systems often give disappointingly small speedups in applications.

The statement that Gaussian elimination is inherently sequential is incorrect. Indeed Pan and Reif themselves point out that by the use of $O(n^2)$ processors Gaussian elimination can be reduced from an $O(n^3)$ algorithm to an $O(n)$ algorithm. Perhaps more important, given the small size of current parallel processors, is the fact that with only n processors Gaussian elimination can run in $O(n^2)$ time. All of this is well known to researchers in parallel computations.

The question of the stability of Gaussian elimination requires some background. Numerical analysis grade linear systems according to their difficulty of solution. At one end of this continuum are well-conditioned problems whose solutions are insensitive to small perturbations in the coefficients; as one moves away from this end one encounters problems that are increasingly sensitive, the so-called ill-conditioned problems. No method can be expected to solve ill-conditioned problems to a given precision without using higher precision somewhere in the computations. Gaussian elimination does about as well as an algorithm can be expected to do. It solves well-conditioned problems accurately, and for all problems, whatever their condition, it produces a solution that almost exactly satisfies the equations (although it may be very inaccurate). This means that if Gaussian elimination fails to give an accurate solution, there is something wrong with the equations, which must be reexamined. For this reason, among others, people with real applications rarely use iterative methods to refine the solutions computed by Gaussian elimination.

There are many types of iterative methods, not one, as the article seems to say. Of these many types, the article confounds two: methods of iterative refinement, which are occasionally used to touch up solutions computed by direct methods, and iterative methods for computing inverses. Moreover, it omits the most widely used class of methods: those that work directly with the system and do not require approximate inverses (although the inverse of an approximation can be used to speed some of them up). All these methods vary in efficiency and potential for parallelization. None of them will

solve ill-conditioned problems accurately without recourse to higher precision arithmetic.

Algorithms that compute a matrix inverse, whether iteratively or directly, are unsatisfactory for two reasons. First, they are a step backward from Gaussian elimination. Not only are they less efficient, but solutions obtained by inverting and multiplying may fail to satisfy the original equations. Second, the inverses of large sparse matrices are not usually sparse, and their computation increases storage costs dramatically. For iterative inversion methods the problem of obtaining a starting approximation is only an additional complication to an otherwise unpromising approach.

Given the drawbacks of iterative matrix inversion, it is not surprising that Schultz's method has languished (*2*). Nor has the problem of finding a starting approximation stumped researchers; it was solved 20 years ago by Ben-Israel (*3*). The fact that it made no splash suggests the status of the problem in the numerical community.

The parallel implementation of Schultz's algorithm proposed by Pan and Reif requires about n^3 processors. The article does not alert the reader to the consequences of this fact. Let us suppose, very generously, that we can place 100 processors with the capability of doing floating-point arithmetic on a board 8 inches square and that we can place the boards an inch apart, so that 100 processors occupy a volume of 1/27 cubic feet. Then the billion processors required to solve a system of order 1000 (a moderate system by today's standards) would occupy a 72-foot cube, the size of a-not-inconsiderable mansion. When the theoretical number $n^{2.5}$ is used in place of n^3 the cube shrinks to 23 feet, still a monstrously large computer. None of this makes any concession to practicalities like heat dissipation and communication costs, which will increase the size of the computer and decrease its efficiency.

Given the unsuitability of the parallel version of Schultz's method for solving problems of moderate size, how can it be said that "Pan and Reif made their method much more efficient for certain commonly occurring problems in which the matrices have lots of zeros. . . "? The answer is that Pan and Reif actually propose two algorithms: the iterative method described in the article and a direct method, based on a technique known as nested dissection, which is closely related to Gaussian elimination. The two algorithms are connected by the fact that the first is used to speed up the second. Nowhere in the article is this made clear.

Does the combined method promise to have a real impact on how linear equations are solved? I think not—for the following reasons. Typically the method of nested dissection might be applied to a system of linear equations associated with a two- or three-dimensional mesh. The order of the system is equal to the number of points in the mesh. The first step in the method is to find a set of points, called a separator, whose removal will divide the mesh into two pieces. The equations corresponding to these points are eliminated from the system and the process is applied recursively to the two submeshes. The elimination of the equations can be accomplished in various ways. Pan and Reif propose to do it by using their first algorithm to solve systems of the same order as the number of points in the separator. There is more to the algorithm than this (for example, the eliminations are done in the order opposite that described above), but the description will suffice for what follows.

Now consider a 32×32 mesh, which corresponds to a system of order about 1000. The separators have 32 points, and accordingly Pan and Reif will require about 32,000 processors to solve a system of order 32. This is overkill.

However, by today's standards a 32×32 mesh is a toy problem. A more realistic problem might come from a three-dimensional mesh of dimensions $10 \times 100 \times 100$. Here the separators are of order 1000, and we are back to the billion processors mentioned above. The difficulty is that n^3 is such a swiftly growing function of n that by the time we reach realistic problems the number of processors has become unrealistic.

It is important to keep things in perspective. By itself, the method of nested dissection is a prime candidate for parallel implementation, and in fact a number of people besides Pan and Reif are working along these lines. Moreover, by coupling nested dissection with Schultz's method, Pan and Reif have obtained some important theoretical results on the complexity of the solution of linear systems. But for the reasons outlined above, the combined algorithm is unlikely to have any impact on the way linear equations are solved.

G. W. STEWART
Department of Computer Science, Institute for Physical Science and Technology, University of Maryland, College Park, MD 20742

REFERENCES AND NOTES

1. G. Kolata, *Science* **228**, 1297 (1985).
2. For a complete analysis of the algorithm, its convergence, and its behavior in the presence of rounding error, see T. Sodorstrom and G. W. Stewart, *SIAM J. Numer. Anal.* **11**, 61 (1974).
3. A. Ben-Israel, *Math. Comput.* **20**, 439 (1966).

8 August 1985; accepted 20 March 1986

Response: Stewart makes three main points.

1) He criticizes what he feels is a general lack of accuracy in Kolata's science reporting.

2) He (correctly) states that there was a prior paper of Ben-Israel that had previously given a method for initializing the Newton iteration method for matrix inversion.

3) He argues (we believe incorrectly) that the parallel algorithms given in our paper for solving linear systems cannot be practical because of large processor bounds.

Our reply to point 1 is that, indeed, Kolata made some misleading statements in her article, particularly in not differentiating between the parallel Newton iteration algorithm for dense matrices (which we view as being only of theoretical interest for large linear systems) and our parallel nested dissection algorithm for sparse matrices (which we have found to be practical for large sparse linear systems).

Also, Kolata gave weather prediction as an example of the application of large linear systems, when, as Stewart correctly points out, weather prediction involves the solution of nonlinear systems. But these nonlinear systems can be linearized by stepping in time. Thus the solution of the resulting large linear systems is a significant component of weather prediction, but not the only one. However, the point Kolata was making was that the solution of large linear systems is an important problem. This is not a controversial statement—it is widely accepted by the scientific community. The solution of large linear systems is central in the engineering sciences (for example, in structural analysis and circuit analysis), in applied physics (for example, in the solution of partial differential equations), and in many optimization problems that occur in economics.

Stewart is correct about Ben-Israel's prior work, but the fault is clearly ours, not Kolata's. Although we were not aware of his work when we presented our paper in 1985, we have long since revised the paper to give Ben-Israel credit for this contribution.

Kolata's reporting, particularly in the areas of mathematics and computer science, has been credible. She has generally succeeded in the difficult task of describing complex technical ideas in simple terms that can be understood and appreciated by the general readership of *Science*. Technical errors, no matter how limited, detract from the general goals of science reporting. Unfortunately, Stewart's comment, which lists what he regards as Kolata's errors, actually contains a number of what we view as errors.

Many in the scientific computing community might disagree with Stewart's statement that iterative methods are not commonly used to improve the accuracy of the solu-