

BOOK REVIEW

APPEARED IN BULLETIN OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 32, Number 1, January 1995, Pages 127-131
©1995 American Mathematical Society
0273-0979/95 \$1.00 + \$.25 per page

The computational complexity of differential and integral equations, by Arthur G. Werschulz. Oxford University Press, 1991, ix + 331 pp., \$55.00. ISBN 0-19-853589-9

Computational mathematics is generally thought to consist of the following sub-fields:

- (1) numerical solution of differential and integral equations,
- (2) numerical linear algebra,
- (3) computational geometry, and
- (4) discrete algorithms and complexity theory.

There has been significant interactions between most of these areas with one notable exception, namely, the first and the last. The most complete overview of all four subject areas has been in the highly influential and brilliantly lucid book on applied mathematics by Gilbert Strang [1]. Even there the full power of complexity theory did not emerge until the seventh chapter, where discrete algorithms were considered. In earlier chapters where differential equations were involved, complexity theory took the form of simple operation counts.

In the book under review, Werschulz makes an attempt to go well beyond this and apply the full force of complexity theory to algorithms designed for the approximation solution of differential and integral equations. Before commenting on the success or failure of this enterprise, it should be noted that the book is well written. It contains very few typos and technical errors, and those that are present are easily corrected. In addition, Werschulz has a sound command of both areas. The book has an appendix on Sobolev Spaces, a subject crucial for finite element approximations, that is complete and to the point. His survey of complexity theory is equally clear and well done.

The book starts out by discussing the general ideas in terms of the simple two-point boundary value problem

$$(1) \quad (Lu)(x) = -\frac{d^2u}{dx^2}(x) + u(x) = f(x) \quad \text{for } x \in [0, 1],$$

$$(2) \quad \frac{du(0)}{dx} = \frac{du(1)}{dx} = 0.$$

Here f is the given data and is assumed to lie in the unit ball F of square integrable functions; i.e.,

$$(3) \quad F = \{f : \|f\|_{L^2} \leq 1\}.$$

The next step is to formulate (1)–(2) in terms of the Sobolev space $H^1(0, 1)$, which consists of $L^2(0, 1)$ functions whose derivatives are in $L_2(0, 1)$. The variational principle takes the following form. Find u in $H^1(0, 1)$ for which

$$(4) \quad \int_0^1 \left[\frac{du}{dx} \frac{dv}{dx} + uv \right] dx = \int_0^1 f v dx$$

holds for all v in $H^1[0, 1]$. The solution u is given by $u = Sf$, where S is a linear operator from F to $H^1(0, 1)$. In a typical finite element approximation one selects a grid consisting of n points in $[0, 1]$ and uses, for example, the space S_n of continuous piecewise linear functions in (4) instead of $H^1[0, 1]$. If v_1, \dots, v_n is a basis for S_n , then the approximate solution u_n in S_n is characterized by the following:

$$(5) \quad \int_0^1 \left[\frac{du_n}{dx} \frac{dv_i}{dx} + u_n v_i \right] dx = \int_0^1 f v_i dx, \quad 1 \leq i \leq n.$$

This is equivalent to a nonsingular set of n algebraic equations in n unknowns.

The starting point for the complexity theory of this or any other algorithm for approximating Sf is an information operator

$$(6) \quad Nf = \begin{bmatrix} \langle f, v_1 \rangle \\ \vdots \\ \langle f, v_n \rangle \end{bmatrix},$$

where $\{\langle \cdot, v_j \rangle\}_{j=1}^n$ are independent continuous linear functionals on $L^2[0, 1]$. This in essence is the basic information available to any algorithm. The latter is defined as the linear operator

$$(7) \quad \varphi: N(F) \rightarrow H^1[0, 1],$$

and the error associated with this algorithm is defined by

$$(8) \quad e(\varphi, N) = \sup_{f \in F} \|Sf - \varphi(Nf)\|_{H^1[0, 1]}.$$

Note that in the finite element example we have $\langle f, v_i \rangle = \int_0^1 f v_i dx$ for $f \in F$ and $u_n = \varphi(Nf)$. The radius and diameter associated with this error are

$$(9) \quad r(N) = \inf_{\varphi} e(\varphi, N)$$

and

$$(10) \quad d(N) = \sup\{\|Sh\| : h \in \ker N \wedge F\},$$

respectively, and it is easy to see that

$$r(N) \leq d(N) \leq 2r(N).$$

In this context it is easily shown that the best algorithm φ^* is obtained by projection, i.e.,

$$(11) \quad \varphi^*(Nf) = \sum \langle f, v_j \rangle S v_j$$

satisfies

$$(12) \quad r(N) = e(\varphi^*, N).$$

One of the most interesting and, indeed, controversial issues to arise in this theory concerns adaptive algorithms. The use of such algorithms, for example, to construct grids for finite-element and finite-difference approximation is widespread [3, 4]. Many feel that the introduction of adaptive grid algorithms has been the single most important advance in numerical analysis in the last decade. Werschulz and his theory are in direct conflict with these trends. Indeed, his results show "... as far as optimal error is concerned, there is no need to consider algorithms using adaptive information,..."

To be more precise, the model used by Werschulz for adaptive methods is one where the information $y_j = \langle f, v_j \rangle$ depends not only on the data f but also on previous information y_1, \dots, y_{j-1} . In addition, the dimension n of the matrix operator (6) also depends on f . We can write this as

$$(13) \quad N^a(f) = \left\{ \begin{array}{c} \lambda_1(f) \\ \lambda_2(f; y_1) \\ \vdots \\ \lambda_n(f; y_1, \dots, y_{n-1}) \end{array} \right\}$$

where $y_j = \lambda_j(f; y_1, \dots, y_{j-1})$ and $\{\lambda_j(\cdot; y_1, \dots, y_{j-1}): 1 \leq j \leq n\}$ are $n = n(f)$ independent continuous linear functionals on F . The nonadaptive version is

$$(14) \quad N^{\text{non}}(f) = \left\{ \begin{array}{c} \lambda_1(f) \\ \lambda_2(f; 0) \\ \vdots \\ \lambda_n(f; 0, \dots, 0) \end{array} \right\},$$

and Werschulz shows that for a wide class of linear problems (Theorem 4.4.2.1, page 39)

$$(15) \quad d(N^{\text{non}}) \leq d(N^a).$$

In most cases $r(N^{\text{non}}) = \frac{1}{2}d(n^{\text{non}})$, which means

$$(16) \quad r(N^{\text{non}}) \leq r(N^a).$$

Much has been written about this type of result and its striking conflict with modern computational practice. After exploring the meaning of the result for elliptic partial differential equations (Chapter 5), Werschulz focuses on linearity as a key issue, suggesting that all problems are in reality nonlinear and that this may be the reason for the practical success of adaptive methods. For example, for boundary value problems like

$$(17) \quad Lu = -\text{div}[p \text{ grad } u] + qu = f \quad \text{in } \Omega$$

and

$$(18) \quad \alpha \frac{\partial u}{\partial n} + \beta u = 0 \quad \text{on } \partial\Omega,$$

the dependence of the solution u on the data f is linear, but its dependence on the coefficients p, q, α, β is nonlinear.

I personally feel that linearity is not the issue, and indeed a version of his Theorem 4.4.2.1 may in fact also be true for a reasonable class of nonlinear problems. If one accepts this, then complexity theory itself emerges as the culprit. Buried in the infs and sups that define the error radius and diameter is the presumption

of information not generally available in practical situations. An all-knowing numerical modeler may not need adaptive grids. In other contexts, however, where there is a great deal of uncertainty about the nature of the solution u and possible singularities, adaptively defined grids provide the single best scheme for getting accurate results.

In addition to the above information/approximation theoretic structures, Werschulz also considers computational costs. He assumes, not unreasonably, that the cost of evaluating Nf is linear in its dimension; i.e.,

$$(19) \quad \text{cost}(N, f) = cn,$$

for some number c . To this is added the cost of the algorithm φ , giving the total computational cost of

$$\text{cost}(\varphi, N) = \sup_{f \in F} \{cn + \text{cost}(\rho, Nf)\}.$$

The ε -complexity of the problem is defined by

$$\text{comp}(\varepsilon) = \inf\{\text{cost}(\varphi, N) : e(\varphi, N) \leq \varepsilon\}.$$

In this context Werschulz derives some interesting results. For example, he shows that the finite-element scheme (5) is within a constant factor from being optimal. More precisely, if $\varphi^{\text{FEM}}(Nf) = u_n$ is defined by (5) and n is sufficiently large so that

$$e(\varphi^{\text{FEM}}, N) \leq \varepsilon,$$

then

$$(20) \quad \overline{\lim}_{\varepsilon \rightarrow 0} \frac{\text{cost}(\varphi^{\text{FEM}}, N)}{\text{comp}(\varepsilon)} \leq 2.1 \left(\frac{c+8}{c} \right).$$

This result is consistent with the fact that finite-element methods are projections and bear a close relationship with the optimal algorithm φ^* defined by (11).

Adaptivity is also considered from the point of view of complexity. Again, it is found that it is in a suitable sense suboptimal. Indeed, the precise results for the boundary value problem (1)–(2) are quite counterintuitive. For example, if the data f is continuous in $(0, 1)$ except for one point of discontinuity, then the adaptive complexity is

$$\text{comp}^a(\varepsilon) = 0(\log \varepsilon^{-1}) \quad \text{as } \varepsilon \rightarrow 0,$$

while the nonadaptive case has

$$\text{comp}^{\text{NON}}(\varepsilon) = 0(\varepsilon^{-1}) \quad \text{as } \varepsilon \rightarrow 0$$

(Corollary 5.7.1, page 138). Here adaptivity wins by an order of magnitude. On the other hand, if f has more than one point of discontinuity, a situation for which adaptive grids are thought to be the most useful, one has

$$\text{comp}^a(\varepsilon) = 0(\varepsilon^{-1}), \quad \text{comp}^{\text{NON}}(\varepsilon) = 0(\varepsilon^{-1}) \quad \text{as } \varepsilon \rightarrow 0;$$

i.e., adaptivity is no better than the nonadaptive case.

The above material is developed in the first five chapters, and in Chapter 6 extensions to other problems are given. Included are elliptic systems, Fredholm equations of the second kind, ill-posed problems, and initial value problems for ordinary differential equations.

Because of the infs and sups involved in $\text{comp}(\varepsilon)$, this type of analysis is often called worst-case analysis. In the final two chapters (Chapters 7 and 8) a probabilistic analysis is treated. This approach typically gives more realistic results. For example, in the worst-case setting the solution u to (1)–(2) is in $H^1[0, 1]$ but no smoother. In this context, the convergence of most numerical schemes including finite elements is in question. This leads to the questionable conclusion that such problems are intractable. This disappears in the average case setting, and Werschulz does a good job of developing this material. The issue concerning adaptivity, on the other hand, does not change, and the general conclusion is the same as the deterministic case (see pages 241–242).

Clearly, a lot of work has gone into this book, and its breadth is very impressive. Nevertheless, the overall impact of the work will depend on how well the theme “. . . adaptive information is no more powerful than nonadaptive information . . . ” holds up in light of computational practice.

REFERENCES

1. G. Strang, *Introduction to applied mathematics*, Wellesley-Cambridge Press, Wellesley, MA, 1986.
2. J. L. Lions and E. Magenes, *Nonhomogeneous boundary value problems and applications*, Vol. 1, Springer-Verlag, Berlin, 1972.
3. I. Babuska, O. C. Zenkiewicz, T. Gago, and E. R. de A. Olivira, *Accuracy estimates and adaptive refinements in finite element computations*, Wiley, New York, 1986.
4. J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *Numerical grid generation*, North-Holland, Amsterdam, 1985.

GEORGE J. FIX
UNIVERSITY OF TEXAS
E-mail address: GJFix@utamat.uta.edu