GENERALIZED COMPACT MULTI-GRID

VICTOR PAN
Mathematics and Computer Science Department
Lehman College, CUNY, Bronx, NY 10468, U.S.A.
JOHN REIF
Computer Science Department, Duke University
Durham, NC 27706, U.S.A.

Abstract—Extending our recent work, based on the ideas of the multi-grid iteration, we decrease the storage space for a smooth solution of a nonlinear PDE and, furthermore, for any smooth function on a multi-dimensional grid and on discretization sets other than grids.

A new approach to the numerical evaluation and storage of the solutions to a large class of linear partial differential equations (PDE's) discretized over a d-dimensional grid G was recently proposed in [1]. The method (which we call *Compact Multi-grid*, since it follows the framework of the multi-grid iterative process) enables us, in particular, to decrease, by roughly the factor of $\log N$, the time and the precision of computing and the storage (memory) space where N is the number of points of the grid G and where the Boolean (bit-) complexity measure is assumed. In practice, N is usually large, so that the improvement is significant.

In the present paper we again follow the multi-grid iteration scheme but focus on the economization of the memory space. We set a refined and more general framework for this method and arrive at more general results. Our generalized compact multi-grid method enables us equally well to decrease the storage space for the solutions to linear and nonlinear PDE's, and only a routine smoothness assumption is needed for this. Furthermore, the method applies to the compression of the space for the storage of all smooth and many nonsmooth solutions to PDE's on multi-dimensional grids and on sets of a more general class in a Euclidean space, which includes many practical cases not covered in [1].

Next, we will formalize our results.

Let a, b, c, d and g denote five fixed positive constants, d integer, G_i denotes the d-dimensional grid of $|G_i| = N_i = 2^{di}$ points, $G_i = \{(j_1 2^{-i}, j_2 2^{-i}, \dots, j_d 2^{-i}), j_k = 0, 1, \dots, 2^i - 1; k = 1, \dots, d\}$, $i = 0, 1, \dots, n$, so that the projection of G_i into each coordinate edge consists of 2^i equally spaced points of the half-open unit interval $\{t, 0 \le t < 1\}$, and the grids $G_0, G_1, \dots, G_n = G$ recursively refine each other. Suppose that we need to store the approximations $u^*(\mathbf{x})$ to a smooth function $u(\mathbf{x})$ given on the finest of these grids, $G = G_n$, within the absolute error bound $\Delta = b/N^c$ and normalized so that $|u^*(\mathbf{x})| \le 1$, for $\mathbf{x} \in G$, where $N = N_n = |G| = |G_n|$. We will assume that $u^*(\mathbf{x})$ satisfies the following (Hölder's type) *smoothness requirement* on G:

$$|u^*(\mathbf{x}) - u^*(\mathbf{y})| \le a(\|\mathbf{x} - \mathbf{y}\|_s)^g \tag{1}$$

where, say, $s = 1, 2$, or ∞, and g is a fixed positive constant, so that the fixed point binary representation (and the storage) of each value $u^*(\mathbf{x})$ on G requires $\lceil c \log N - \log b \rceil$ bits of memory,

V. Pan is supported by NSF Grant CCR-9020690 and by PSC CUNY Award #662478. J. Reif is supported by DARPA/ISTO Contracts N00014-88-K-0458, DARPA N00014-91-J-1985, N00014-91-C-0114, NASA subcontract 550-63 of prime contract NAS5-30428, US-Israel Binational NSF Grant 88-00282/2, and NSF Grant NSF-IRI-91-00681.

Preprint of paper appearing in Computers Math Applications, Volume 25, Number 9, p.3-5, May 1993.

which is $O(\log N)$ as $N \to \infty$. It seems that the order of $N \log N$ bits are needed for the storage of the N values of $u^*(\mathbf{x})$ at all the N points of G, but actually we have the following result:

PROPOSITION 1. *$O(N)$ bits of memory suffice in order to represent the values of all the approximations to the function $u^*(\mathbf{x})$ on a multi-dimensional grid of N points, provided that (1) holds.*

REMARK. In the approach of [1], where $u_i(\mathbf{x})$ denote the discretizations over the grids G_i of the solution $u(\mathbf{x})$ to a PDE, it is assumed that $\log_2 |u(\mathbf{x}) - u_i(\mathbf{x})| \leq \alpha - \beta i$ for two fixed constants α and β. This is in the spirit of solving PDE's by the multi-grid methods (see e.g., [2–11]). Now, however, we will show that even the weaker assumption (1) is sufficient to make our compact scheme work.

To arrive at Proposition 1, let $\ell(v)$ denote the number of bits in the fraction of the floating point representation of a binary number v, and define the auxiliary functions $u_i^*(\mathbf{x})$ on G_i that minimize $\ell(u_i^*(\mathbf{x}))$ subject to $|u_i^*(\mathbf{x}) - u^*(\mathbf{x})| \leq a\,2^{-gi}$ for $\mathbf{x} \in G_i$ and for $i = 0, 1, \dots, n-1$, that is, $u_i^*(\mathbf{x})$ is obtained by rounding off $u^*(\mathbf{x})$ to $\ell(u_i^*(\mathbf{x})) = \lceil g\,i - \log_2 a \rceil$ bits, thus ignoring the bits of $u^*(\mathbf{x})$ that represent the values less than $2^{-\ell(u_i^*(\mathbf{x}))-k}$.

Denote

$$e_i(\mathbf{x}) = u_i^*(\mathbf{x}) - u_{i-1}^*(\mathbf{x}), \qquad \text{for } \mathbf{x} \in G_{i-1},\ i = 1, \dots, n. \tag{2}$$

Further, for every point $\mathbf{y}$ of $G_i - G_{i-1}$, fix some (say, northwestern on a 2-dimensional grid) nearest neighbor $\mathbf{x} = \mathbf{x}_i(\mathbf{y})$ on G_{i-1} such that

$$\|\mathbf{y} - \mathbf{x}\| = 2^{-i}, \tag{3}$$

and define

$$e_i(\mathbf{y}) = u_i^*(\mathbf{y}) - u_{i-1}^*(\mathbf{x}). \tag{4}$$

Equation (2) and the definiton of $u_i^*(\mathbf{x})$ imply that

$$|e_i(\mathbf{x})| \leq a(2^{-gi} + 2^{-g(i-1)}) = a\,2^{-g(i-1)}(1 + 2^{-g}). \tag{5}$$

Since the bits of $e_i(\mathbf{x})$ corresponding to the values less than $2^{-\lceil g\,i - \log_2 a \rceil}$ are ignored, we obtain that $\ell(e_i(\mathbf{x})) \leq \lceil g\,i - \log_2 a \rceil - \lfloor g\,i - g - \log_2(a(1 + 2^{-g})) \rfloor$, and therefore,

$$\ell(e_i(\mathbf{x})) \leq 2g + 3. \tag{6}$$

Furthermore, apply (2) and (4) and deduce that $e_i(\mathbf{y}) = (u_i^*(\mathbf{y}) - u_i^*(\mathbf{x})) + e_i(\mathbf{x})$. Now apply (1), (3) and (5) and deduce that $|e_i(\mathbf{y})| \leq a\,2^{-g(i-1)}(1 + 2^{-2g})$, and therefore,

$$\ell(e_i(\mathbf{y})) \leq 3g + 3, \qquad \text{if } \mathbf{y} \in G_i. \tag{7}$$

Now, according to the compact multi-grid storage scheme, we first store the value $u^*(\mathbf{x})$ at the point $\mathbf{x}$ of G_0 [by using $\lceil c \log N - \log b \rceil = O(\log N)$ memory bits] and then, recursively for $i = 1, \dots, n$, store the values $e_i^*(\mathbf{y})$ at all the N_i points $\mathbf{y}$ of G_i [by using at most $(3g+3)N_i - N_{i-1}$ memory bits due to (6) and (7)]. The overall storage of at most $(3g+3)\sum_{i=1}^{n} N_i - \sum_{i=0}^{n-1} N_i = (3g+3)\sum_{i=1}^{n} 2^{di} - \sum_{i=0}^{n-1} 2^{di} < 2(3g+3)N = O(N)$ bits suffice to store $e_i^*(\mathbf{y})$ for all $\mathbf{y} \in G_i$ and fo $i = 1, \dots, n$.

This is a compact representation of $u^*(\mathbf{x})$ on G, since if we need, we may recover $u^*(\mathbf{x})$ for any point $\mathbf{x}$ of G by using the saved values $u^*(\mathbf{x})$ on G_0 and $e_i^*(\mathbf{x})$ on G_i for $i = 1, \dots, n$ and by recursively applying (2) and (4) for $i = 1, \dots, n$. For each $\mathbf{x}$ of G, this recovery takes at most $O(\log n)$ bit-operations; furthermore, in many applications, we may store the function $u^*(\mathbf{x})$ in the above compressed form and only very rarely need to decompress it (see [1], the end of Section 1.4).

The above approach and the results of Proposition 1 can be extended in the two following directions.

1. Instead of the above functions $u_i^*(\mathbf{x})$, obtained by the truncation of $u(\mathbf{x})$, we may use any auxiliary functions $u_i(\mathbf{x})$ on the grids G_i for $i = 0, 1, \dots, n$, such that $u^*(\mathbf{x}) = u_n(\mathbf{x})$ on G, and

$$u_i(\mathbf{x}) = P_i\, u_{i-1}(\mathbf{x}) + c_i(\mathbf{x})\ , \qquad \mathbf{x} \in G_i,$$

where P_i is a prolongation operator and $P_i\, u_{i-1}(\mathbf{x})$ is a prolongation of $u_{i-1}(\mathbf{y})$ from G_{i-1} to G_i obtained by means of interpolation (typically, by averaging) of the values of $u_i(\mathbf{y})$ taken at a certain set of points $\mathbf{y}$ of G_{i-1} that lie near $\mathbf{x}$, and where for each $\mathbf{x} \in G_i$ and for each i, the fraction of the floating point binary representation of $e_i(\mathbf{x})$ contains at most g bits for a fixed constant g. Then Proposition 1 is extended as long as the prolongation operators P_i enforce that

$$\ell\big(e_i(\mathbf{x})\big) = O(1),$$

for all i.

2. Proposition 1 can be extended to the case where the grids $G_0, \dots, G_n$ are replaced by any rapidly expanding sets $S_0, \dots, S_n$ (such that $|S_n| = N$, $|S_0| = O(1)$, and for every point $\mathbf{y}$ of S_i there is its neighbor $\mathbf{x}$ of S_{i-1} such that $\|u_i(\mathbf{y}) - u_{i-1}(\mathbf{x})\| < \gamma^{-i}$, and furthermore, $|S_i| \geq \Theta |S_{i-1}|$, for two constants $\gamma > 1$ and $\Theta > 1$, and $S_0 \subset S_1 \subset \cdots \subset S_n = S$). Note that this extension enables us to treat many nonsmooth functions $u(\mathbf{x})$ too, since we may increase the density of S where $u(\mathbf{x})$ is not smooth.

REFERENCES

1. V. Pan and J. Reif, Compact multi-grid, *SIAM J. on Scient. Statist. Comput.* **13** (1), 119–127 (1992).
2. A. Brandt, Multi-level adaptive technique (MLAT) for fast numerical solutions to boundary value problems, *Proc. 3rd Int. Conf. Numerical Methods in Fluid Mechanics (Paris, France) 1972*, Vol. 18, pp. 82–89, Springer-Verlag, Berlin, W. Germany, (1972).
3. A. Brandt, Multi-level adaptive solutions to boundary value problems, *Mathematics of Computation* **31**, 333–390 (1977).
4. A. Brandt, *Multi-grid Techniques: 1984 Guide, with Applications to Fluid Dynamics*, Available as GMD Studien Nr. 85, GMD-AIW Postfach 1240, D-5205, St. Augustin 1, W. Germany, (1984).
5. R. Bank and T. Dupont, An optimal order process for solving finite element equations, *Mathematics of Computation* **36**, 35–51 (1981).
6. C.C. Douglas and J. Douglas, Jr., A unified convergence theory for abstract multi-grid or multi-level algorithms, serial and parallel, *SIAM J. on Numerical Analysis* (to appear).
7. W. Hackbusch, On the convergence of multi-grid iteration applied to finite element equations, Report 77-8, Universität zu Köhn, (1977).
8. W. Hackbusch, Convergence of multi-grid iterations applied to difference equation, *Mathematics of Computation* **34**, 425–440 (1980).
9. W. Hackbusch, *Multi-Grid Methods and Applications*, Springer, Berlin, (1985).
10. W. Hackbusch and U. Trottengerg (Eds.), Multi-grid methods, *Springer's Lecture Notes in Math.*, Vol. 960, (1982).
11. S. McCormick, Editor, *Multi-Grid Methods*, SIAM, Philadelphia, (1987).