PHYSICAL REVIEW E, VOLUME 65, 035701(R)

Rational approximation with multidimensional scattered data

Xu-Guang Hu,* Tak-San Ho, and Herschel Rabitz
Department of Chemistry, Princeton University, Princeton, New Jersey 08544-1009
(Received 31 August 2001; published 6 March 2002)

Accurate and efficient rational approximation schemes are presented for interpolating multidimensional scattered data with a novel weighted least-squares procedure including domain decomposition. Two particular representations of the method are formulated and the corresponding algorithms are implemented. Numerical tests on three- and six-dimensional model systems are carried out, demonstrating high efficiency and accuracy. This work was motivated by the need for multidimensional function approximation using irregular grids when solving quantum fluid dynamics equations, and the method should have broader physical applications.

DOI: 10.1103/PhysRevE.65.035701 PACS number(s): 02.60.Gf, 02.60.Ed, 02.70.−c

Many mathematical modeling problems require multidimensional scattered data interpolation. A particular case of interest arises in the fluid dynamical formulation of quantum mechanics which has recently been explored for the simulation of molecular systems [1–3]. Within the Lagrangian description of quantum fluid dynamics (QFD), a major numerical challenge is the need for accurate and efficient function and derivative approximation schemes working with an irregular grid distribution.

It is generally difficult to properly treat multidimensional interpolation and approximation problems, especially, involving scattered data. Computationally, issues of accuracy, efficiency and stability play critical roles in these problems [4,5]. The approximation of a function by rational expressions has provided an attractive means for a variety of numerical applications. One prime advantage of rational approximations is their ability to handle functions with poles, which is particularly relevant in solving the QFD equations and many other applications. Rational approximations can be determined in a number of ways [6,7], but often with practical limitations, especially, in multidimensional situations. For example, Padé approximants require an explicit series expansion of the function to be approximated, and numerical values of the corresponding high order function derivatives are usually not readily available. Moreover, although the coefficients of the desired rational approximation can be determined by interpolating any set of tabulated data points, the procedure is usually feasible only for one-dimensional problems. Finally, the related continued fractions are yet to be explored for multidimensional applications. Below, a robust least-squares approach in conjunction with a domain decomposition technique is introduced for generating multidimensional rational approximations using scattered data. This approach is intended to circumvent many practical difficulties with the existing methods.

A rational approximation of an N-dimensional function $f(\mathbf{r})$ in a Cartesian coordinate frame $\mathbf{r}=\{\xi_1,\xi_2,\ldots,\xi_N\}$ can be expressed as follows

$$g(\mathbf{r})=\left(b_0+\sum_{k=1}^{m} b_k\phi_k(\mathbf{r})\right)\Bigg/\left(1-\sum_{k=1}^{n} b_{m+k}\phi_{m+k}(\mathbf{r})\right), \tag{1}$$

where $\{\phi_k(\mathbf{r})\}_{k=1}^{m+n}$ is a set of independent multidimensional monomials whose linear combinations render multidimensional polynomials with no common factors in the numerator and denominator. We propose the following algorithm, within the framework of the weighted linear least-squares procedure, to determine the unknown coefficients in Eq. (1), especially, with the availability of only scattered data. First, rearrange Eq. (1) into

$$g(\mathbf{r})=b_0+\sum_{k=1}^{m} b_k\phi_k(\mathbf{r})+g(\mathbf{r})\sum_{k=1}^{n} b_{m+k}\phi_{m+k}(\mathbf{r}), \tag{2}$$

and then replace $g(\mathbf{r})$ on the right-hand side of the above equation by $f(\mathbf{r})$ to yield the relation

$$g(\mathbf{r})=b_0+\sum_{k=1}^{m} b_k\phi_k(\mathbf{r})+f(\mathbf{r})\sum_{k=1}^{n} b_{m+k}\phi_{m+k}(\mathbf{r}), \tag{3}$$

which can be considered as a better approximation to $f(\mathbf{r})$ than Eq. (1). The replacement of $g(\mathbf{r})$ by $f(\mathbf{r})$ is essential in the development of the method. Consequently, the right-hand side of Eq. (3) now becomes linear in terms of the unknown coefficients, assuming that the values of $f(\mathbf{r})$ are known on an irregular grid. Given a set of data $\{\mathbf{r}_i, f(\mathbf{r}_i)\}_{i=1}^{M}$ such that $M\geqslant m+n+1$, and an appropriate weight $w(\mathbf{r}_i)$, minimization of the quadratic error functional

$$J[\{b_k\}]=\sum_{i=1}^{M}[f(\mathbf{r}_i)-g(\mathbf{r}_i)]^2 w(\mathbf{r}_i), \tag{4}$$

yields a set of $m+n+1$ linear algebraic equations for the unknown coefficients $\{b_k\}_{k=0}^{m+n}$:

$$\sum_{j=0}^{m+n}\alpha_{kj}b_k=\beta_k, \quad k=0,1,2,\ldots,(m+n) \tag{5}$$

with

$$\alpha_{kj}=\sum_{i=1}^{M} B_k(\mathbf{r}_i)B_j(\mathbf{r}_i)w(\mathbf{r}_i), \quad \beta_k=\sum_{i=1}^{M} f(\mathbf{r}_i)B_k(\mathbf{r}_i)w(\mathbf{r}_i), \tag{6}$$

where $\{B_j(\mathbf{r})\}_{j=0}^{m+n}$ is a new basis set composed of $\{1,\phi_p(\mathbf{r}),f(\mathbf{r})\phi_q(\mathbf{r})\}_{p=1,2,\ldots,m}^{q=m+1,m+2,\ldots,m+n}$. In general, the accuracy of the rational approximation will increase as additional higher order monomials are included in Eq. (1), how-

*Present address: Life Science Department, Accelrys Ltd., 9685 Scranton Road, San Diego, CA 92121.

ever the efficiency of solving Eq. (5) will decrease due to the increased number of unknown coefficients. In the following treatment, assuming that the function $f(\mathbf{r})$ is *locally smooth*, then good accuracy and efficiency can be achieved by the domain decomposition technique, i.e., dividing the underlying domain into a set of *overlapping* subdomains, without invoking high order monomials in Eq. (1).

With a properly chosen weight function $w(\mathbf{r},\mathbf{r}_c)$, a local rational approximation $g(\mathbf{r},\mathbf{r}_c)$ to $f(\mathbf{r})$ over a subdomain $\Omega(\mathbf{r}_c)$ centered at $\mathbf{r}_c=\{\xi_{1c},\xi_{2c},\ldots,\xi_{Nc}\}$, is then carried out following the same procedures described in Eqs. (1)–(5). This process is repeated using a sequence of properly chosen $\Omega(\mathbf{r}_c)$'s to cover the whole domain. In the following treatment, two local rational approximations (LRA) using only low-order monomials are presented.

The first LRA, hereafter referred to as (2,1) LRA, is based on employing all monomials up to second-order in the numerator and only first-order monomials in the denominator to produce

$$g_{2,1}(\mathbf{r},\mathbf{r}_c)=\frac{b_0(\mathbf{r}_c)+\sum_{k=1}^{N} b_k(\mathbf{r}_c)\tilde{\xi}_k+\sum_{k=1}^{N'} b_{N+k}(\mathbf{r}_c)(\tilde{\xi}\tilde{\xi}')_k}{1-\sum_{k=1}^{N} b_{N+N'+k}(\mathbf{r}_c)\tilde{\xi}_k}, \tag{7}$$

where $\tilde{\xi}_k=\xi_k-\xi_{kc}$ and $N'=N(N+1)/2$ with N being the dimensionality of $\mathbf{r}$. Here, the second-order monomials are denoted as $(\tilde{\xi}\tilde{\xi}')_k$ and the total number N_b of unknown coefficients is $N+(N+1)(N+2)/2$. Although Eq. (7) in principle can include higher order monomials, numerical results indicate that the current formulation, with suitable domain decompositions, is efficient and accurate for rendering good accuracy.

The second LRA is implemented in two *partial correction* steps for attaining better numerical efficiency, i.e., in terms of smaller computer memory storage, and is related to continued fractions. The scheme starts, i.e., at the zeroth step, with a (1,1) LRA

$$g_{1,1}^{(0)}(\mathbf{r},\mathbf{r}_c)=\frac{b_0^{(0)}(\mathbf{r}_c)+\sum_{k=1}^{N} b_k^{(0)}(\mathbf{r}_c)\tilde{\xi}_k}{1-\sum_{k=1}^{N} b_{N+k}^{(0)}(\mathbf{r}_c)\tilde{\xi}_k}, \tag{8}$$

around the local center $\mathbf{r}_c$. It is noted that, in this step, only $N_b^{(0)}=2N+1$ unknown coefficients $b_k^{(0)}$ are involved over the subdomain $\Omega(\mathbf{r}_c)$ utilizing the data set $\{\mathbf{r}_i,f(\mathbf{r}_i)\}_{i=1}^{L}$ with $L\geqslant 2N+1$.

In the first correction step, to improve the accuracy of Eq. (8), a partial correction, involving the (1,1) LRA and all second-order *diagonal* monomials in the numerator, is introduced via the following *ansatz*

$$g_{2,1}^{(1)}(\mathbf{r},\mathbf{r}_c)=\frac{g_{1,1}^{(0)}(\mathbf{r},\mathbf{r}_c)+b_0^{(1)}(\mathbf{r}_c)+\sum_{k=1}^{N} b_k^{(1)}(\mathbf{r}_c)\tilde{\xi}_k^2}{1-\sum_{k=1}^{N} b_{N+k}^{(1)}(\mathbf{r}_c)\tilde{\xi}_k}, \tag{9}$$

where the number $N_b^{(1)}$ of unknown coefficients $b_k^{(1)}$ remains as $2N+1$.

In the second step, further corrections are performed by replacing the second-order diagonal terms $\tilde{\xi}^2$ in the numerator with the remaining second-order *cross* terms. For an N-dimensional problem, there are altogether $N_c=N(N-1)/2$ second-order cross monomials $\{(\tilde{\xi}\tilde{\xi}')_k\}_{k=1}^{N_c}$. To attain even more numerical efficiency, this step is implemented in a sequence of small steps. Specifically, when the dimensionality N is an odd number, the N_c cross terms are divided into $n_s=(N-1)/2$ subgroups, each with N terms. Likewise, when N is an even number, they are divided into $n_s=N/2$ subgroups, each with $N-1$ terms. In each small step, only a small number, either N or $N-1$, of cross terms are included for the correction. At the $(n-1)$th small step with $n>1$, following the same procedures as described in the first correction step, we can easily, in terms of the first $n-1$ subgroups of second-order cross terms, arrive at the expression

$$g_{2,1}^{(n)}(\mathbf{r},\mathbf{r}_c)=\frac{g_{2,1}^{(n-1)}(\mathbf{r},\mathbf{r}_c)+b_0^{(n)}(\mathbf{r}_c)+\sum_{k=1}^{N'} b_k^{(n)}(\mathbf{r}_c)(\tilde{\xi}\tilde{\xi}')_k}{1-\sum_{k=1}^{N} b_{N'+k}^{(n)}(\mathbf{r}_c)\tilde{\xi}_k},$$

$$n=2,\ldots,n_s+1, \tag{10}$$

where $N'=N$ when N is an odd number and $N'=N-1$ when N is an even number. We remark that the vector β [cf. Eq. (6)] consists of the quantity $f(\mathbf{r}_i)-g_{2,1}^{(n-1)}(\mathbf{r}_i,\mathbf{r}_c)$, instead of just $f(\mathbf{r}_i)$, in the calculations leading to $g_{2,1}^{(n)}(\mathbf{r},\mathbf{r}_c)$. Moreover, the same set of local data $\{\mathbf{r}_i,f(\mathbf{r}_i)\}_{i=1}^{L}$ for the subdomain $\Omega(\mathbf{r}_c)$ is utilized throughout the second correction step. The number $N_b^{(n)}$ of unknown coefficients $b_k^{(n)}$ with $n>1$ in each step is $2N$ when N is even and $2N+1$ when N is odd. Finally, each $g_{2,1}^{(n)}(\mathbf{r},\mathbf{r}_c)$ with $n>1$ takes on a closed form of *upward* continued fractions of low-order polynomials after expressing all its preceeding counterparts explicitly in terms of monomials. The complete continued fraction LRA $g_{2,1}^{(n_s+1)}(\mathbf{r},\mathbf{r}_c)$ is hereafter denoted as (1/2/2,1) LRA. Our numerical results in this paper reveal that the accuracy of the (1/2/2,1) LRA is nearly independent of the chosen ordering of the correction sequence. The main advantage of the (1/2/2,1) LRA is that the number $N_b^{(n)}$ of the unknown coefficients $b_k^{(n)}$, with $n=0,1,2,\ldots,n_s+1$, in each correction step, including the zeroth one, scales linearly with the spatial dimensionality N while the number N_b for (2,1) LRA scales with N^2.

The choice of weight function $w(\mathbf{r},\mathbf{r}_c)$ in the weighted least-squares procedure, cf. Eqs. (4), (5), and (6), may have significant effect on the accuracy and stability of the proposed LRA methods. Three commonly used weight functions are Gaussians, cubic splines, and quartic splines [8]. However, our numerical results in some $N=3$ and 6 model systems (see below) showed that the LRA methods based on these *a priori* weight functions are less accurate and stable even compared to their counterparts based on a simple constant weight. In the following a robust procedure is proposed

TABLE I. Errors for functions and their derivatives.

	(2,1) LRA			(1/2/2,1) LRA			RBF [a]		
	$E(f)$[b]	$E(d_1)$	$E(d_2)$	$E(f)$	$E(d_1)$	$E(d_2)$	$E(f)$	$E(d_1)$	$E(d_2)$
three-dimensional case									
P_2	exact	exact	exact	3.1[−4]	4.3[−3]	3.0[−2]	6.5[−6]	4.4[−4]	1.1[−2][c]
P_3	4.3[−4]	3.4[−3]	5.0[−2]	4.7[−4]	6.1[−3]	4.5[−2]	1.0[−3]	4.9[−2]	1.4[−1]
P_4	5.4[−4]	3.9[−3]	5.1[−2]	8.4[−4]	8.6[−6]	9.1[−2]	2.7[−3]	1.2[−2]	9.4[−1]
F	4.6[−4]	5.0[−3]	5.5[−2]	2.6[−4]	4.0[−3]	7.0[−2]	8.3[−4]	2.7[−2]	1.2[−1]
six-dimensional case									
P_2	exact	exact	exact	1.3[−4]	2.5[−3]	2.8[−2]	3.7[−6]	8.8[−4]	7.3[−3]
P_3	3.0[−4]	4.1[−3]	2.3[−2]	7.7[−4]	6.8[−3]	8.1[−2]	1.9[−3]	3.2[−2]	1.0[−1]
P_4	4.4[−4]	3.9[−3]	3.6[−2]	8.9[−4]	9.3[−3]	8.4[−2]	2.2[−3]	3.9[−2]	8.3[−1]
F	2.0[−4]	1.9[−3]	1.4[−2]	5.7[−4]	6.3[−3]	7.8[−2]	7.8[−4]	1.3[−2]	3.1[−1]

[a] $\alpha=6$ is used.

[b] For $N=3$, $-1\leqslant P_n\leqslant 1$ and $-2\leqslant F\leqslant 2$. For $N=6$, $-1.5\leqslant P_n$, $F\leqslant 1.5$.

[c] $[-k]$ means 10^{-k} throughout the table.

to incorporate *a posteriori* weights that are intended to provide good accuracy and stability to the methods: First, we solve the linear algebraic equations for the unknown coefficients b_k (or $b_k^{(n)}$) over each subdomain $\Omega(\mathbf{r}_c)$ with a constant weight $w(\mathbf{r})=1$. Then, we calculate the deviations $d_i(\mathbf{r}_c)=|f(\mathbf{r}_i)-g(\mathbf{r}_i,\mathbf{r}_c)|, i=1,2,\ldots,L$ to obtain the *a posteriori* weight of the form

$$w(\mathbf{r}_i,\mathbf{r}_c)=1-d_i(\mathbf{r}_c)/d_{max}(\mathbf{r}_c), \tag{11}$$

where $d_{max}(\mathbf{r}_c)$ is the largest deviation out of the L deviations $d_i(\mathbf{r}_c)$. Finally, we once again solve the linear algebraic equations for the same set of unknown coefficients using the weights based on Eq. (11). The test results show that this procedure can significantly improve the overall accuracy and stability.

Numerical tests of the LRA schemes are carried out on three-dimensional (3D) and six-dimensional (6D) functions defined in the associated hypercubes $[-1,1]^3$ and $[-1,1]^6$, respectively. These examples, given functional values over irregular grids, demonstrate the quality of the LRA schemes for approximating multidimensional functions. Specifically, we have considered three polynomials $P_2(\mathbf{r})$, $P_3(\mathbf{r})$, $P_4(\mathbf{r})$ (of orders 2, 3, and 4) and a composite nonpolynomial $F(\mathbf{r})$ (of three elementary functions) which, respectively, have the following expressions:

$$P_2(\mathbf{r})=a_0+\sum_{i=1}^{N} a_i\xi_i+\sum_{i\leqslant j}^{N} b_{ij}\xi_i\xi_j,$$

$$P_3(\mathbf{r})=P_2(\mathbf{r})+\sum_{i,j=1}^{N} c_{ij}\xi_i^2\xi_j+\sum_{i<j<k}^{N} d_{ijk}\xi_i\xi_j\xi_k,$$

$$P_4(\mathbf{r})=P_3(\mathbf{r})+\sum_{i,j=1}^{N} e_{ij}\xi_i^3\xi_j+\sum_{i<j}^{N} f_{ij}\xi_i^2\xi_j^2+\sum_{i\neq j,k;j<k}^{N} g_{ijk}\xi_i^2\xi_j\xi_k+\sum_{i<j<k<l}^{N} h_{ijkl}\xi_i\xi_j\xi_k\xi_l,$$

$$F(\mathbf{r})=\sin\left(\sum_{i=1}^{N} a_i\xi_i\right)\exp\left(\frac{1}{N}\sum_{i=1}^{N} a_i\xi_i\right)+\cos\left(\sum_{i=1}^{N} a_i\xi_i\right)\times\exp\left(-\frac{1}{N}\sum_{i=1}^{N} a_i\xi_i\right)+\ln\left[1+\left(\sum_{i=1}^{N} a_i\xi_i\right)^2\right]. \tag{12}$$

The expansion coefficients for each function in Eq. (12) are random numbers assuming alternating + and − signs such that the function is properly bounded over the corresponding domain (see footnotes in Table I). These random numbers have been generated, using the DES generator [6], in the interval $[1,1.5]$ for 3D functions and in the interval $[0,1]$ for 6D functions.

In the following calculations, input data (i.e., function values) for each test function were sampled over irregular grids. For simplicity, these irregular grids have been obtained based on random numbers again generated by the DES generator [6]. Specifically, 13 and 11 random numbers were, respectively, generated along each coordinate for the 3D cube and for the 6D hypercube, thus yielding $13^3=2197$ irregular points in the former and $11^6=1\,771\,561$ in the latter. The average distances between any two adjacent data points are 0.4 and 0.5 for the 3D and 6D cases, respectively.

Based on the sampled irregular grids, a domain decomposition procedure has been implemented as follows: The overlapping subdomains, each associated with a randomly picked local center, are obtained in succession to cover the whole domain under consideration. Each subdomain, say, corresponding to a local center $\mathbf{r}_c$, contains a small number (here 27 in the 3D case and 729 in the 6D case, including the center $\mathbf{r}_c$) of the nearest neighboring (NN) grid points, which are indexed in ascending orders in terms of distances from $\mathbf{r}_c$ and stored accordingly. Moreover, the local centers are kept apart from one another at appropriate distances. To this end, a half or nearly a half (i.e., the first 13 in the 3D case and first 363 in the 6D case) of the NN grid points associated with every local center are excluded from being the local center of any other subdomains. The ratios of the total number M of data points to the number of subdomains are approximately

10 in the 3D case and 140 in the 6D case. Consequently, the only output data needed to be stored for later evaluations of the functions in the LRA schemes are the local center $\mathbf{r}_c$, the size of each subdomain $\Omega(\mathbf{r}_c)$, and the corresponding coefficients b_k (or $b_k^{(n)}$). To evaluate the approximate function value $f(\mathbf{r})$ at an arbitrary point $\mathbf{r}$ in the domain, one simply recalls the stored output data assigned to a particular subdomain with the center $\mathbf{r}_c$ closest to the point $\mathbf{r}$.

Table I gives the average absolute errors of the test functions and their derivatives computed using the two LRA schemes. These results are calculated using 4913 (in the 3D) and 7 529 536 (in the 6D case) randomly picked positions. These positions are not on the original grids used to generate LRA's. For comparison, a spline-based radial basis function (RBF) interpolation was also carried out using multiquadrics (MQ) $\sqrt{\alpha^2+r^2}$. Interpolation based on the RBF's has received considerable attention recently because of its effectiveness for handling multivariate scattered data [9]. Here, the same data sets have been utilized for the MQ-RBF interpolation, with the optimized parameter α taking on values between 4 and 10. $E(f)$, $E(d_1)$ and $E(d_2)$ in Table I designate the errors for each function, and its first and second derivatives, respectively. The derivatives were explicitly evaluated using the two LRA schemes with no special treatment exploited for the computations on boundary points. It is found that both the (2,1)- and (1/2/2,1)-LRA schemes can accurately approximate multidimensional polynomials, of at least up to fourth order, and an arbitrary nonpolynomial. Moreover, the (1/2/2,1) LRA scheme is generally as good as the (2,1) LRA in accuracy, except for the second-order polynomial $P_2(\mathbf{r})$ for which the latter is exact. As expected, the errors in the derivatives tend to rise in going from $E(d_1)$ to $E(d_2)$. Similar test calculations have also been carried out using input data arranged on evenly spaced regular grids. In these cases the LRA results (not shown here) are of the same quality as those depicted in Table I. On the other hand, improved results were obtained with the RBF method using data sampled on evenly spaced regular grids due to the increase of numerical stability, typical of spline interpolation methods. In the calculations, the CPU time required by the RBF method is larger by about a factor of 4 for the 3D case and 15 for the 6D case over that required by both LRA schemes.

It is important to point out, from the test examples above, that (1) the accuracy of the LRA schemes is generally independent of how the input data are distributed and (2) the LRA schemes require far fewer data points, e.g., than the RBF method, to attain the same level of accuracy. These observations are relevant, particularly when extending the applications to problems of high dimensions. Moreover, it is found that the efficiency of the present method is closely related to two main components in the implementation: (1) domain decomposition, which involves a searching procedure for distances between the position to be evaluated and the local center of a subdomain and (2) the linear least-squares procedure for determining the coefficients in the rational approximation. In the present calculations, the searching algorithm scales roughly as $O(M^{1.5})$, whereas the noniterative least-squares procedure scales approximately linearly, i.e., $O(M)$, where M is the total number of scattered data points. However, these scalings exhibited some dependence on the total number of overlapping subdomains. Specifically, the scaling in the searching procedure is in favor of a small number of subdomains, however, the opposite is true in the least-squares procedure. In practice, the number of subdomains will need to be properly chosen for the domain associated with the given M to achieve a good balance between efficiency and accuracy.

In summary, the rational approximation methods developed in the paper possess many attractive features important to the solution of a variety of problems, including the solution of multidimensional QFD equations. The nonlinear rational approximations were implemented in terms of a noniterative, robust least-squares algorithm. Moreover, the method is meshless and is therefore amendable to an arbitrary data distribution, which characterizes many data driven problems. Finally, the LRA schemes are efficient and accurate for multidimensional interpolations due, on the one hand, to the adoption of the moving least-squares technique in conjunction with proper domain decomposition, and on the other hand, through the use of only first- and second-order monomials throughout the implementation. In particular, the (1/2/2,1) LRA involves only linear algebraic equations of small size, scaling as the dimensionality N of the problem, and requires only low memory for solving these equations. The present method was shown to be capable of generating accurate approximations to arbitrarily chosen functions of up to six-dimensions from scattered data. Application to the solution of the QFD equations is underway.

The authors acknowledge support from the Department of Energy.

[1] F.S. Mayer *et al.*, J. Chem. Phys. **111**, 2423 (1999), and references therein.

[2] X.-G. Hu *et al.*, Phys. Rev. E **61**, 5967 (2000), and references therein.

[3] C.L. Lopreore and R.E. Wyatt, Phys. Rev. Lett. **82**, 5190 (1999); J. Suñé and X. Oriols, *ibid.* comment and reply, **85**, 894 (2000); **85**, 895 (2000).

[4] *Advances in Multivariate Approximation*, edited by W. Haussmann *et al.* (Wiley-VCH, New York, 1999).

[5] *Multivariate Approximation and Splines*, edited by G. Nürnberger *et al.* (Birkhäuser, Boston, 1997).

[6] W. H. Press *et al.*, *Numerical Recipes in C*, 2nd ed. (Cambridge University Press, New York, 1992).

[7] *Nonlinear Numerical Methods and Rational Approximation II*, edited by A. Cuyt (Kluwer Academic, Dordrecht, 1994).

[8] T. Belytschko *et al.*, Comput. Methods Appl. Mech. Eng. **139**, 3 (1996).

[9] M. D. Buhmann, in *Multivariate Approximation: From CAGD to Wavelets*, edited by K. Jetter and F. Utreras (World Scientific Publishing, Singapore, 1993), pp. 35–75.