Collocation and optimization initialization

E. J. Kansa[1] & L. Ling[2]
[1]Convergent Solutions, USA
[2]Hong Kong Baptist University, Hong Kong

Abstract

Integrated volumetric methods such as finite elements and their “meshless” variations are typically smoother than the strong form finite difference and radial basis function collocation methods. Numerical methods decrease their convergence rates with successively higher orders of differentiation along with improved conditioning. In contrast, increasing order of integration increases the convergence rate at the expense of poorer conditioning. In the study presented, a two-dimensional Poisson equation with exponential dependency is solved. The solution of the point collocation problem becomes the initial estimate for an integrated volumetric minimization process. Global, rather than local integration, is used since there is no need to construct any meshes for integration as done in the “meshless” finite element analogs. The root mean square (RMS) errors are compared. By pushing the shape parameter to very large values, using extended precision, the RMS errors show that spatial refinement benefits are relatively small compared to pushing shape parameters to increasing larger values. The improved Greedy Algorithm was used to optimize the set of data and evaluation centers for various shape parameters. Finally, extended arithmetic precision is used to push the range of the shape parameters.
Keywords: meshless radial basis functions, multiquadric, strong and weak formulation, partial differential equations, global minimization.

1 Introduction

The objective is to find efficient methods for the numerical solution of partial differential equations (PDEs), integral equations (IEs), and integro-differential equations (IDEs). Assume that a dependent variable, $\mathrm{U}(\overrightarrow{x}, t)$ is an unknown piece-wise continuous function. Over the interior, let $\mathcal{L}$, be a linear or nonlinear

WIT Transactions on Modelling and Simulation, Vol 57, © 2014 WIT Press
www.witpress.com, ISSN 1743-355X (on-line)
doi:10.2495/BE370051

hyperbolic, elliptic, or elliptic differential operator with the associated forcing function, $\mathbf{f}(\overrightarrow{x}, t)$. Let $\wp$ be well-posed boundary operators, and let $\mathbf{g}(\overrightarrow{x}, t)$ be the associated forcing boundary condition. The set of well-posed interior and boundary conditions are:

$$\mathcal{L}\mathbf{U} = \mathbf{f}, \text{ over } \Omega \backslash \partial\Omega, \tag{1}$$

$$\wp\mathbf{U} = \mathbf{g}, \text{ on } \partial\Omega, \tag{2}$$

The finite element method (FEM) uses trial polynomial-based discretization method in which the interior operator, $\mathcal{L}$, and boundary operator, $\wp$, are locally integrated and multiplied by a test function. One appealing aspect of FEM is the fact that integration increases the convergence rate and Gauss theorem:

$$\int_V (\nabla \cdot \overrightarrow{A}) dV = \int_S (\overrightarrow{A} \cdot \overrightarrow{n}) dS \tag{3}$$

reduces orders of differentiation. Local integration requires connectivity rules over different polyhedra and the integration schemes are defined over either Voronoi or Delaunay tessellations, requiring overhead. The dependent variables may not be continuous, but multi-valued at a center. Even though the low order polynomial basis functions are integrated, FEM still has only polynomial convergence, rather than the exponential convergence of C^∞ radial basis functions.

The test problem is the Poisson equation over a unit 2D square:

$$\nabla^2 U(x_1, x_2) = (a^2+b^2)\exp(ax_1+bx_2) \text{ over } \Omega, \tag{4}$$

with the following Dirichelet conditions on

$$\begin{aligned} U(x_1 = 0, x_2) &= \exp(ax_1) \text{ on } \partial\Omega(x_2 = 0) \\ U(x_1, x_2) &= \exp(bx_2) \text{ on } \partial\Omega(\ x_1 = 0), \end{aligned} \tag{5}$$

and Neumann conditions on the opposite boundaries:

$$\begin{aligned} \partial U/\partial x_1 &= a \cdot \exp(ax_1+b) \text{ on } \partial\Omega(x_2 = 1) \\ \partial U/\partial x_2 &= b \cdot \exp(a+bx_2) \text{ on } \partial\Omega(x_1 = 1). \end{aligned} \tag{6}$$

The exact solution is:

$$U(x_1, x_2), = \exp(ax_1+bx_2). \tag{7}$$

In Kansa [1], any dependent variable, $U(\overrightarrow{x},t)$ is expanded in terms of the N radial basis functions, ϕ_j, and N_p polynomial terms, $p_k(\overrightarrow{x})$, in terms of $N+N_p$

expansion coefficients, α_j.

$$\mathbf{U}(\overrightarrow{x},t) = \sum_{j=1}^{N} \phi_j(\overrightarrow{x})\alpha_j(t) + \sum_{j=N+1}^{N+N_p} p_j(\overrightarrow{x})\alpha_j(t) \tag{8}$$

subject to the constraint conditions

$$\sum_{j=1}^{N} p_k\overrightarrow{(x)}\alpha_j(t) = 0,\ k = N+1, \cdots N+N_p. \tag{9}$$

The approximate function, $U(\overrightarrow{x},t)$, is expanded in terms of C^∞ radial basis functions (RBFs) as

$$\phi_j = (1+(r_j/c_j)^2)^\beta, \text{ (generalized MQ)}, \ r_j = \|\overrightarrow{x}-\overrightarrow{\mathbf{y_j}}\|, \tag{10}$$

where c_j is the local shape parameter, $\beta \geq -1/2$,$\{\overrightarrow{\mathbf{y}}\}$ is the set of data centers, and $\{\overrightarrow{\mathbf{x}}\}$ is the set of evaluation points; $\{\overrightarrow{\mathbf{x}}\}$ and $\{\overrightarrow{y}\}$ can be distinct or the same. Note that c_j^2 plays the roles of a wavelet dilation parameter. Madych [2] has provided theoretical guidance for optimizing the interpolation procedure. They showed that C^∞ radial basis functions, such as MQ, converge exponentially as:

$$\text{convergence rate } \sim O(\lambda^\gamma), \text{where}\gamma = (c/h), \text{ and } 0 < \lambda < 1, \text{ h is the fill distance.} \tag{11}$$

Madych also showed differentiation reduces the rate of convergence; however, if the parameter, γ, is rather large, the derivatives can be very good approximates. It does not make sense, on a CPU count, to use uniform very fine discretization over the entire domain, Ω, unless the length scale is small everywhere. Some insight of the location of fine length scales can be gained from the effect of the source functions, $\mathbf{f}(\overrightarrow{x},t)$ and $\mathbf{g}(\overrightarrow{x},t)$. The set of adjustable parameters, $textQ = \{\overrightarrow{\mathbf{x}}, \overrightarrow{\mathbf{y}}, textc_j^2, \overrightarrow{\alpha}\}$ determine how well the MQ expansion approximates the true solution. The usual procedure in solving PDEs, IEs, and IDEs is to choose either a shape parameter or a distribution, $\{c_j^2\}$, a set of data centers, $\{\overrightarrow{\mathbf{y}}\}$ and evaluation points, $\{\overrightarrow{\mathbf{x}}\}$, then find a set of expansion coefficients by solving a set of N×N coupled linear or nonlinear equations. The process of finding the expansion coefficients, $\{\overrightarrow{\alpha}\}$, is unfortunately not always stable; furthermore, the set, $\{\overrightarrow{\alpha}\}$, depends upon the choices of subset, $\{\overrightarrow{\mathbf{x}}, \overrightarrow{\mathbf{y}}, c_j^2\}$. Using either a very fine discretization everywhere in the domain may have a firm theoretical foundation, and likewise for using very large shape parameters, but both lead to ill-conditioning problem on finite precision computers, no matter how many digits of accuracy are available. On real world computers, compromises must be made to obtain the objective of the most accurate results in the most efficient manner possible.

The search for the optimal method to solve PDEs, IEs, and IDEs is evolving. Given a double precision computer and a choice of a shape parameter, the Greedy Algorithm developed by Ling and Schaback [3] finds the optimal set of data and evaluation centers from a very large initial sample of such centers. Huang *et al.*

[4] and Cheng [5] demonstrated that when using extended precision arithmetic, a very coarse distribution of data and evaluation centers coupled with large shape parameters yields extremely accurate numerical solutions very computationally efficiently. Intuitively, this seems wrong because the CPU time for multiple precision is much longer than for double precision. The reason why intuition is incorrect is that although the CPU time per data and evaluation center is larger with extended precision, the very fact that the total number of points for specific target accuracy is orders of magnitude smaller, hence the total cost is considerably reduced.

A general unifying approach is presented by Galperin and Zheng [6, 7] to solve PDEs, IEs, and IDEs as a global minimization problem in which the interior operator, acting upon the C^{∞} RBFs is volume integrated over $\Omega\backslash\partial\Omega$ and the boundary operator, $\wp$, acting upon the C^{∞} RBFs is surface integrated over each $\partial\Omega_k$. The functional, $\mathcal{F}$, to be minimized, over the parameter set Q is:

$$\min_{\mathrm{q}\in\mathrm{Q}} \mathcal{F} = \varpi \int_{\Omega\backslash\partial\Omega} (\mathcal{L}\mathbf{U}\text{-}\mathbf{f})\mathrm{d}\overrightarrow{\mathbf{x}} + (1-\varpi)\int_{\partial\Omega} (\wp\mathbf{U}\text{-}\mathbf{g})\mathrm{d}\overrightarrow{\mathbf{x}} \ \leq \eta \tag{12}$$

where $\mathbf{Q}$ is the set of free parameters, $\mathbf{Q} = \{\overrightarrow{\mathbf{x}}, \overrightarrow{\mathbf{y}}, \mathrm{c}_j^2, \alpha\}$, $\varpi < 1$ and η is a prescribed error criterion. The integrations performed were global, rather than using local tesselation methods. Only the integrated basis functions at the integration endpoints are required. For example, define $\psi(\mathbf{y})$ to be:

$$\psi(\overrightarrow{\mathbf{y}}_{\mathbf{j}}) = \int_{\Omega\backslash\partial\Omega} \phi(\overrightarrow{\mathrm{x}}\text{-}\overrightarrow{\mathbf{y}}_{\mathrm{j}})\mathrm{d}\overrightarrow{\mathbf{x}} \tag{13}$$

and similarly over the boundaries, $\partial\Omega_k$. Then, the integral, $\int_{\Omega\backslash\partial\Omega}(\mathcal{L}\mathbf{U}\text{-}\mathbf{f})\overrightarrow{d\mathbf{x}}$**,** expanded as:

$$\int_{\Omega\backslash\partial\Omega} (\mathcal{L}\mathbf{U}\text{-}\mathbf{f})\mathrm{d}\overrightarrow{\mathbf{x}} = \sum \int_{\Omega\backslash\partial\Omega} (\mathcal{L}\phi(\overrightarrow{\mathrm{x}}\text{-}\overrightarrow{\mathbf{y}}_{\mathbf{j}})\mathrm{d}\overrightarrow{\mathbf{x}})\alpha_{\mathrm{j}} \text{ - } \int_{\Omega\backslash\partial\Omega} \mathbf{f}\mathrm{d}\overrightarrow{\mathbf{x}} \tag{14}$$

If one wishes to have a finite element analog, then local integration over the tesselation can yield N equations in N unknowns.

Although the Galperin and Zheng procedure eliminates the need to solve sets of linear or nonlinear equations, global minimization requires a huge number of evaluations of the functional, $\mathcal{F}$. Fast reliable global minimization algorithms capable of dealing with at least 4N parameters are not perfected. One way to minimize the number of functional evaluations is reduce the total number of free parameters. The problem is that the set of optimal shape parameters, data and evaluation centers, and expansion coefficients are inter-related. The usual procedure is to choose a set of N $\{\overrightarrow{x}\}$ and $\{\overrightarrow{\mathbf{y}}\}$ and either a shape parameter or a distribution of shape parameters, then solve an N×N set of equations to find the expansion coefficients. To overcome the limitations of double precision, the MATLAB compatible multi-precision package was obtained from www.advanpix.com.

To improve the convergence rate, we want γ to become very large; this can be accomplished in two ways: (1) The h-scheme (spatial refinement) increases the number, N_h, of data centers, but a relatively small c. (2) The c-scheme increases c, but requires a significantly smaller of data centers, N_c, where $N_c \ll N_h$; the c-scheme requires $O(N_c^3) \ll O(N_h^3)$ operations. For purposes of efficiency, the c-scheme is superior and preferable over the h-scheme. The disadvantage of either the h-scheme,or the c-scheme is that as $\gamma \rightarrow \infty$, the system of linear equations becomes very ill-conditioned and subject to extreme round-off errors on computers with limited arithmetic precision.

2 Numerical results

The test problem was chosen over a unit square domain, and the exponential parameters for the forcing function, f, were $a = 2$ and $b = 2$, respectively. At $x_1 = x_2 = 1$, $U = \exp(5) = 148.4132$. To minimize the number of data centers, a geometric progression of increasingly finer discretization was used, and the origin was appended in the construction of the tensor product mesh. Although many computer experiments were made, the best input parameters will be summarized here. In the paper of Fedoseyev *et al.* [8], the computational domain, Ω was extended slightly beyond the boundaries, $\partial\Omega_k$ given by the line segments of the unit square: (0,1) (0,0), (0,1) (1,1) (0,1), The domain extended 0.020 beyond the unit square in the x_1 and x_2 directions. The parameters on the boundaries, see Wertz *et al.* [9], were increased by a factor of 30. Consequently, numerical integration such as Gauss–Legendre (GL) needs to be performed, by first interpolating this integrand onto the appropriate GL zeros. then perform the numerical integrations.

The shape parameters distribution is a power law two parameter recipe for N terms:

$$c_j^2 = c_{\text{min}}^2 (c_{\text{max}}^2 / c_{\text{min}}^2)^{(j-1)/N-1)} \qquad (15)$$

While the power law recipe seems to work for monotonic functions, the wavelet relations. From previous experiments and 440 digits of accuracy, $c_{\text{min}}^2 = 1.6e5$ and $c_{\text{max}}^2 = 1.6e0$. It is seriously doubtful that such a simple two parameter recipe is general, and this be the subject of more study. The order of the GL integration was 25 and the points and weights were generated with extended precision arithmetic. Since most scientists and engineers are familiar with finite difference, element, or volume methods, the customary procedure is to use the brute force of the computer and use extremely fine meshing. This approach is fine for 2 and 3-dimensional problems, but impractical for 6D Boltzmann equations. Since the maximum value of U occurs at $x_1 = x_2 = 1$, the geometric progression algorithm produced successively finer discretization in the x_1 and x_2 directions. The first point was fixed at 0.16 for x_1 and 0.18 for x_2. Afterwards, the minimum values of x_1 and x_2 was -0.02 and the maximum values of x_1 and x_2 was 1.025 were appended to the tensor product 2D mesh. The boundary locus line segments: (0,0), (0,1), (1,1), (1,0) were also added.

Table 1: RMS errors of the Galperin–Zhang formulation using Gauss–Legendre (GL) and exact integration.

N data centers	RMS error-GL integration	RMS error-exact integration
12	0.0723	0.0016
30	0.0636	0.0014
42	0.0586	0.0012
56	0.0554	0.0010
72	0.0539	0.0009
90	0.0526	0.0008
110	0.0512	0.0007
240	0.0386	0.0005

The set of expansion coefficients, $\{\alpha\}$, was obtained by solving the point collocation for the PDE and boundary conditions. The root mean square (RMS) errors of these expansion coefficients relative to the exact solution ranged from 1e-34 to 3e-38 with the total number of points ranging from 12 to 240. To answer the question whether the weak formulation yields accurate results, the 2D integration of the ∇^2 was integrated using the Gauss theorem for both Gauss–Legendre integration and exact integration; likewise, the boundary conditions were integrated exactly over one dimension.

The following table shows the results of increasing the number data centers for the Galperin–Zhang weak formulation using GL integration and exact 2D integration. During previous testing, and using an increasing number of digits of precision, the values of $\mathrm{c}^2_{\mathrm{min}}$ and $\mathrm{c}^2_{\mathrm{max}}$ were pushed., fixing the set, $\{\mathrm{c}_j^2\}$, using the set $\{\alpha\}$ from the solution of the point collocation scheme, and fixing the starting points of the geometric progression for the tensor product meshes.

Note that increasing the number of data centers, h-refinement, while fixing the set, $\{\mathrm{c}_j^2\}$ does show successively smaller RMS errors, but it is questionable whether the extra effort is warranted. The point of this exercise is to provide a good starting point for the global optimization process in which all free parameters are varied to find very deep global minima instead of a local minimum.

3 Discussion

The papers of Galperin and Zheng [6, 7] offer a general framework for the numerical solution of PDEs, IEs, and IDEs. The test solutions are expanded in terms of C RBFs, then operated upon by the domain and boundary operators that are volume and surface integrated, respectively. A functional is constructed from these test expansions that is constrained to be less than or equal to an error bound. The functional is minimized by global optimization methods that are unfortunately not very robust given a large number of parameters. Presently, a decoupling process is required in which the set of data and evaluation centers are input parameters as well as either a uniform or variable shape parameter distribution, then the expansion coefficients are solved as a set of possible ill-conditioned equation systems requiring extended arithmetic precision. Unless some breakthrough in the global minimization algorithm is developed, parameter splitting methods will need to be continued to be used. Optimizing the minimization procedure will require further research and development.

References

[1] Kansa, E.J.: Multiquadrics – A scattered data approximation scheme with applications to computational fluid dynamics: II. Solutions to parabolic, hyperbolic, and elliptic partial differential equations. *Comput. Math. Appl.* **19**(6-8), pp. 147–161, 1990.

[2] Madych, W.R.: Miscellaneous error bounds for multiquadric and related interpolators. *Comput. Math. Applic.* **24**(12), pp. 121–138, 1992.

[3] Ling, L & Schaback, R.: Stable and convergent unsymmetric meshless collocation methods. *SIAM Journal on Numerical Analysis* **46**(3), pp. 1097–1115, 2008.

[4] Huang, C.S., Lee, C.F. & Cheng A.H.D.: Error estimate,optimal shape parameter,and high precision computation of multiquadric collocation method. *Eng. Anal. Bound. Elem.* **31**, pp. 615–623, 2007.

[5] Cheng A.H.D.: Multiquadric and its shape parameter—a numerical investigation of error estimate, condition number, and round-off errors of arbitrary precision computation. *Eng. Anal. Bound, Elem.* **36**, pp. 220–239, 2012.

[6] Galperin, E.A. & Zheng, Q.: Solution and control of PDE via global optimization methods. *Comput. Math. Applic.* **25**(10/11), pp. 103–118, 1993.

[7] Galperin, E.A. & Zheng, Q.: Application of global optimization to implicit solution of partial differential equations. *Comput. Math. Applic.* **25**(10/11), pp. 119–124, 1993.

[8] Fedoseyev, A.I., Friedman, M.J. & Kansa, E.J.: Improved multiquadric method for elliptic partial differential equations via PDE collocation on the boundary. *Comput. Math. Appl.* **43**(3-5), pp. 491–500, 2002.

[9] Wertz, J, Kansa, E.J. & Ling, L.: The role of the Multiquadric Shape Parameters in solving Elliptic Partial Differential Equations. *Comput. Math. Applic.* **51**(8), pp. 1335–1348, 2006.