+ Model
ARTICLE IN PRESS

Available online at www.sciencedirect.com

Mathematics and Computers in Simulation xxx (2006) xxx–xxx

MATHEMATICS AND COMPUTERS IN SIMULATION

www.elsevier.com/locate/matcom

Least squares collocation solution of elliptic problems in general regions

V. Pereyra*, G. Scherer

Abstract

We consider the solution of elliptic problems in general regions by embedding and least squares approximation of overdetermined collocated tensor product of basis functions. The resulting least squares problem will generally be ill-conditioned or even singular, and thus, regularization techniques are required. Large scale problems are solved by either conjugate gradient type methods or by a block Gauss–Seidel approach. Numerical results are presented that show the viability of the new method.

Keywords: Least squares; Collocation; Elliptic

1. Introduction

In this paper we introduce a new class of methods for elliptic problems in general regions that superficially resemble embedding (capacitance), and Galerkin collocation methods, but are quite different in conception and implementation. In fact, they are more related to so called meshless methods.

To start with, we remain with the differential equation form of the problem instead of going to the integral or variational formulation, which would also be possible. Second, as we will see, the resulting method is essentially mesh free and intrincated regions in any dimension do not require any additional care, which makes it considerably simpler to implement than conventional finite differences or finite elements methods.

We consider general linear elliptic second order operators of the form:

$$\begin{aligned} Lu(\mathbf{x}) &= f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \\ \mathcal{B}u(\mathbf{x}) &= g(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega, \end{aligned} \tag{1}$$

where $\mathbf{x} \in R^n$, Ω a multiple connected region of R^n, $\partial\Omega$ the boundary of Ω and $\mathcal{B}$ is a boundary operator that may involve first order derivatives. Since our approach is novel, we will describe and test first the algorithm in two space dimensions.

There are a number of methods similar to the one we are presenting here. To mention a few recent ones: Betcke and Trefethen [2] present a new version of the method of particular solutions for eigenvalue problems that uses a collocation and least squares approach. Larsson and Fornberg [3] review and extend several Radial Basis methods developed in the last 10 years that use collocation. Belytschko et al. [1] discuss extensively meshless methods, including collocation by various basis functions.

* Corresponding author.

doi:10.1016/j.matcom.2006.06.022

We believe that the main differences with this approach are in our use of least squares collocation with tensor product basis functions on a box embedding the irregular region where the problem is set, and the use of regularization to solve the potentially singular problems that result.

2. Discretization

We embed the region Ω in a rectangle $\mathcal{R}$ and consider a tensor product family of basis functions associated with a uniform mesh in $\mathcal{R}$. Thus, we make the Ansatz

$$u(x, y) = \sum c_{ij} B_i(x) B_j(y), \quad i = 1, \dots, N_x;\ j = 1, \dots, N_y,$$

with $N = N_x \times N_y$. Assuming that the basis functions $B_i(x),\ B_j(y)$ are sufficiently differentiable and replacing in the differential equation and boundary conditions we get:

$$\begin{aligned} &L \sum c_{ij} B_i(x) B_j(y) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \\ &\mathcal{B} \sum c_{ij} (B_i(x) B_j(y)) = g(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega. \end{aligned} \tag{2}$$

If there are no cross derivatives in the operator L we obtain:

$$\begin{aligned} &\sum c_{ij} [L B_i(x) B_j(y) + B_i(x) L B_j(y)] = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \\ &\sum c_{ij} [\mathcal{B} B_i(x) B_j(y) + B_i(x) \mathcal{B} B_j(y)] = g(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega. \end{aligned} \tag{3}$$

Now we select collocation points in the interior of the region $(x_s, y_s) \in \Omega$, $s = 1, \dots, M_\mathrm{i}$, and in the boundary $(x_t, y_t) \in \partial\Omega,\ t = 1, \dots, M_\mathrm{b}$. We choose $M = M_\mathrm{i} + M_\mathrm{b} \ggg N$. The resulting overdetermined system can be written in matrix form as:

$$\mathbf{LB} C = F, \qquad \mathbf{B} C = G,$$

where,

$$\begin{aligned} &\mathbf{LB} = L B_i(x_s) B_j(y_s) + B_i(x_s) L B_j(y_s), \\ &\mathbf{B} = \mathcal{B} B_i(x_s) B_j(y_s) + B_i(x_s) \mathcal{B} B_j(y_s), \\ &C = c_{ij}, \\ &F = f(x_s, y_s), \\ &G = g(x_t, y_t). \end{aligned} \tag{4}$$

Observe that we are free to choose the collocation points any way we like. No connectivity information is required and the distribution of points can be guided by any additional need that we may consider important, such as representing sharp variations of the solution. Arbitrary boundaries and even holes can be handled without any additional effort, since the least squares algorithm will use regularization to cope with any singularity or ill-conditioning, as we have shown in Refs. [5,6].

3. Numerical experiments

3.1. Poisson's equation in 2D

We consider now L to be the Laplacian in two dimensions, $\mathcal{B}$ the identity, and the region Ω to be the circle:

$$(x - 0.5)^2 + (y - 0.5)^2 = 0.25.$$

We embed this region in the unit square. The source and boundary functions are chosen as:

$$f(x, y) = -25(\sin 5x + \sin 5y),$$
$$g(\theta) = \sin(5[0.5 + 0.5\ \cos(\theta)]) + \sin(5[0.5 + 0.5\ \sin(\theta)]), \tag{5}$$

which uses the parametric representation of the circle:

$$x = 0.5 + 0.5\ \cos(\theta), \qquad y = 0.5 + 0.5\ \sin(\theta), \quad \theta \in [0, 2\pi].$$

The exact solution is $u(x, y) = \sin 5x + \sin 5y$.

We choose various values for the number of basis functions, for a fixed number of collocation points and thresholds. In the tables below M stands for the sum of the interior and boundary collocation points. Interior points are selected from an uniform mesh in the unit square, which for Test 1.1 has 20×20 points, while for Test 1.2 has 30×30 points. In both cases there are 180 boundary points spaced every 2°.

We use in these tests the Truncated SVD method of Ref. [5] and LSQR, Paige and Saunders [4] conjugate gradient least squares solver. The value of the SV truncation threshold is chosen to be thresh = 1.0e−6. For TSVD, irnk stands for the calculated rank of the least squares matrix. Observe also that the number of control vertices, $\mathrm{nv}x \times \mathrm{nv}y$, does not include the phantom vertices, so that the total number of unknowns is really $(\mathrm{nv}x + 2) \times (\mathrm{nv}y + 2)$. ration stands for the ratio of two consecutive irnk, while ratio_err is the (reversed) ratio between two consecutive resmax, the maximum absolute error at the collocation points. Finally, rms is the residual mean square error (i.e., an approximation to the integral L_2 norm). For LSQR we use an estimate of the condition number, CONLIM = 1.0e6.

Meth.	M	irnk	nv*x*	nv*y*	ration	resmax	ratio_err	rms	Time (s)
TSVD	456	25	3	3	0.00E+00	0.72E+00	0.00E+00	0.15E+00	0.05
TSVD_	456	64	6	6	0.26E+01	0.51E+00	0.14E+01	0.10E+00	0.16
TSVD	456	117	9	9	0.18E+01	0.22E+00	0.24E+01	0.43E−01	0.63
TSVD	456	184	12	12	0.16E+01	0.12E+00	0.18E+01	0.16E−01	2.01
TSVD	456	265	15	15	0.14E+01	0.39E−02	0.30E+02	0.10E−02	4.38
TSVD	456	352	18	18	0.13E+01	0.17E−02	0.23E+01	0.79E−03	8.55
TSVD	828	25	3	3	0.00E+00	0.73E+00	0.00E+00	0.16E+00	0.08
TSVD	828	64	6	6	0.26E+01	0.63E+00	0.12E+01	0.13E+00	0.36
TSVD	828	117	9	9	0.18E+01	0.39E+00	0.17E+01	0.74E−01	1.46
TSVD	828	184	12	12	0.16E+01	0.20E+00	0.18E+01	0.38E−01	4.19
TSVD	828	265	15	15	0.14E+01	0.73E−01	0.29E+01	0.19E−01	8.75
TSVD	828	352	18	18	0.13E+01	0.44E−01	0.13E+01	0.96E−02	16.73

Meth.	M	nv*x*	nv*y*	resmax	ratio_err	rms	Time (s)
PS	456	3	3	0.72E+00	0.00E+00	0.15E+00	0.03
PS	456	6	6	0.51E+00	0.14E+01	0.10E+00	0.07
PS	456	9	9	0.22E+00	0.23E+01	0.41E−01	0.13
PS	456	12	12	0.11E+00	0.19E+01	0.16E−01	0.25
PS	456	15	15	0.48E−02	0.26E+02	0.88E−03	0.39
PS	456	18	18	0.32E−01	0.14E+00	0.94E−02	0.40
PS	828	3	3	0.73E+00	0.00E+00	0.16E+00	0.08
PS	828	6	6	0.63E+00	0.12E+01	0.13E+00	0.18
PS	828	9	9	0.37E+00	0.17E+01	0.74E−01	0.37
PS	828	12	12	0.19E+00	0.18E+01	0.36E−01	0.71
PS	828	15	15	0.76E−01	0.29E+01	0.20E−01	1.37
PS	828	18	18	0.43E−01	0.17E+01	0.72E−02	2.03

We see from this set of experiments that TSVD and LSQR give essentially the same results, except for Test 1.1 with $\mathrm{nv}x = \mathrm{nv}y = 18$, where TSVD is significantly more accurate. However, in terms of efficiency, LSQR is considerably faster, especially for the larger systems, as we have observed in our previous papers. Also, using more collocation points does not seem to be helpful.

3.2. Poisson's equation in 3D

We consider now $\Delta u = -(\sin x + \sin y + \sin z)$ on the sphere Ω with center at (0.5, 0.5, 0.5) and radius 0.5. An uniform mesh with $20 \times 20 \times 20$ points results in 3544 collocation points in Ω and we also take 400 points in the boundary. The spline basis is defined on a regular mesh with $8 \times 8 \times 8$ points in the unit cube. LSQR with CONLIM 10^{10} gives a solution with:

$$L_\infty \text{ error} = 0.027; \qquad \text{rms} = 0.006; \qquad \text{time} = 124\,\text{s}.$$

3.3. Laplace's equation in 2D with a singularity

We consider Laplace's equation in a quarter circle centered at the origin and with radius 1. The solution is $u(x, y) = 0.5/\pi \log r$. We use 432 interior and 36 boundary collocation points, isolating the singularity at the origin with a circle of radius 10^{-2}. We use 12×12 basis functions in the unit square and CONLIM $= 10^6$. LSQR produces in 576 iterations and 264 s of CPU time:

- At collocation points: L_∞ error $= 0.30$; rms $= 0.06$.
- On a radius at 18° with 20 points: L_∞ error $= 0.08$; rms $= 0.017$.
- On a quarter circle of radius 0.2 with 45 points: L_∞ error $= 0.025$; rms $= 0.01$.

3.4. Problems with discontinuities

It is of interest to consider problems with piece-wise smooth solutions. In order to explore the possibilities of extending the method to such problems we consider first a one-dimensional example:

$$\begin{aligned} &y'' = 0, \\ &y(0) = 0, \\ &y(2) = 100, \\ &y(1^-) - y(1^+) + 50 = 0, \\ &y'(1^-) - y'(1^+) = 0, \end{aligned} \tag{6}$$

where $y(1^-)$ and (1^+) stand for the left and right limits of the functions. Thus, the problem's solution has a jump of 50 at $x = 1$ but a continuous derivative. The solution is a piece-wise linear function that rises from 0 to 25 in [0, 1] and from 75 to 100 in [1, 2].

The two subdomains denoted by domain 1:[0, 1] and domain 2:[1, 2], respectively, are each covered with a node mesh with four basis functions and a uniform mesh of 10 collocation points. The results using a TSVD algorithm are:

For domain 1: $\|\text{rel.error}\|_\infty = 0.23\text{e}{-6}$, rms $= 1.89\text{e}{-7}$.
For domain 2: $\|\text{rel.error}\|_\infty = 0.12\text{e}{-6}$, rms $= 5.65\text{e}{-8}$.

Next, we consider as a simple 2D example, a Poisson equation on the rectangle $[0, 2] \times [0, 2]$ with Dirichlet boundary conditions on all external boundaries. There is a discontinuity line at $x = 1$ with the jump conditions: $u(1_-, y) = u(1_+, y) - 10$; and $\partial u(1_-, y)/\partial x = \partial u(1_+, y)/\partial x$.

The two subdomains, domain 1:$[0, 1] \times [0, 2]$ and domain 2:$[1, 2] \times [0, 2]$, respectively, are covered with a nodal mesh and a uniform mesh of collocation points. For domain 1, the boundary function is $g_1(x, y) = 0$, and for domain 2 is $g_2(x, y) = 10$.

The solution of the problem is: $u_1(x, y) = 0$ and $u_2(x, y) = 10$.

Next are some of the test results, using a LSQR based algorithm with different number of collocation points and nodes.

	Coll.mesh	Nodemesh	Iter	$\|rel.error\|_\infty$	rms	Time (s)
Test 1			189			33
Domain 1	10 × 10	5 × 5		0.27e−2	1.04e−3	
Domain 2	10 × 10	5 × 5		0.50e−3	1.44e−3	
Test 2			393			43
Domain 1	12 × 12	8 × 8		0.28e−1	9.20e−3	
Domain 2	12 × 12	8 × 8		0.33e−2	9.70e−3	

4. Conclusions

We have shown in a number of different examples how the proposed method works. These exemplify the ability of the method to handle general regions in two and three dimensions, singularities and discontinuities. Although we have not attempted to produce general implementations, the work to program these tests was fairly small as compared to mesh or element based methods for general regions.

We do not have a clear idea of the interplay between number of collocation points and accuracy. Incrementing the number of basis functions produces more accurate results and in general useful engineering accuracy is obtained with a very modest investment in computer time and storage.

The approach is much more general than this limited application to elliptic problems and also could benefit from the use of other basis functions besides cubic B-splines.

References

[1] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, Meshless methods: an overview and recent developments, Comput. Meth. Appl. Mech. Eng. (Meshless Methods) 139 (1996) 3–47.
[2] T. Betcke, L. Trefethen, Reviving the method of particular solutions, SIAM Rev. NA-03–12, Oxford University, submitted for publication.
[3] E. Larsson, B. Fornberg, A numerical study of some radial function based solution methods for PDE's, Comput. Math. Appl. 46 (2003) 891–902.
[4] Ch.C. Paige, M.A. Saunders, LSQR: an algorithm for sparse linear equations and sparse least squares, ACM Trans. Math. Software 8 (1982) 43–71.
[5] V. Pereyra, G. Scherer, Least squares scattered data fitting by truncated SVDs, Appl. Numer. Math. 40 (2002) 73–86.
[6] V. Pereyra, G. Scherer, Large scale least squares scattered data fitting, Appl. Numer. Math. 44 (2003) 225–239.