INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2003; **57**:509–536 (DOI: 10.1002/nme.687)

A collocation meshless method based on local optimal point interpolation

Carlos Zuppa[1] and Alberto Cardona[2,*,†]

[1] *Departamento de Matemática, Universidad Nacional de San Luis, Chacabuco y Pedernera, 5700 San Luis, Argentina*
[2] *Centro Internacional de Métodos Computacionales en Ingeniería (Cimec-Intec), Universidad Nacional del Litoral—Conicet, Güemes 3450, 3000 Santa Fe, Argentina*

SUMMARY

This paper deals with the use of the local optimal point interpolating (LOPI) formula in solving partial differential equations (PDEs) with a collocation method. LOPI is an interpolating formula constructed by localization of optimal point interpolation formulas that reproduces polynomials and verifies the delta Kronecker property. This scheme results in a truly meshless method that produces high quality output and accurate solutions. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: meshless methods; point collocation methods; optimal point interpolation formula

1. INTRODUCTION

Meshless, or meshfree, methods have been widely studied in recent years and the amount of literature on them is vast: e.g. we refer to the reviews by Belytschko *et al.* [1], Liu *et al.* [2] and Li and Liu [3]. They have advantages over the popular finite element method, which makes them appealing in many situations. We mention, for instance, problems involving large deformations where a finite element mesh could become severely and unacceptably distorted. Meshless methods do not need a mesh for interpolation like finite elements do, and for this reason they can be very easily linked to a CAD system. Adaptation to increase computation accuracy is very easily implemented, and also, they can handle components damage and cracks without any problem.

Meshless methods can be classified into those approximating the PDEs in weak form, and those directly approximating the partial differential equations in strong form. In the former category, some means to perform integration are required. Examples of this class are diffuse elements [4], element free Galerkin [5], reproducing kernel particle method [6], h-p clouds [7],

*Correspondence to: Alberto Cardona, Centro Internacional de Métodos Computacionales en Ingeniería (Cimec-Intec), Universidad Nacional del Litoral—Conicet, Güemes 3450, 3000 Santa Fe, Argentina.
†E-mail: acardona@intec.unl.edu.ar

Received 22 October 2001
Revised 20 May 2002
Accepted 27 August 2002

finite spheres [8], partition of unity [9], etc. In the latter category, PDEs are usually discretized by collocation. Smooth particle hydrodynamics [10], the finite point method [11, 12] and the reproducing kernel particle method proposed by Aluru [13, 14] are all examples of meshless methods in strong form.

In order to perform integration, many methods that approximate PDEs in weak form do need an auxiliary or background mesh, which complicates implementation. On the other hand, the strong form of solution gains particular interest because it leads to truly meshless methods that do not make use of any kind of mesh.

The concept of solving PDEs using *radial basis functions* (RBFs) was introduced by Kansa in 1990 [15, 16]. He implemented this approach using the *multiquadrics* radial basis function (MQ RBF) proposed by Hardy [17, 18] for interpolation of scattered data. Franke [19] compared global RBFs against many popular compactly supported schemes for 2D interpolation and concluded that global RBF schemes were superior in several aspects.

Kansa's MQ method for solving PDEs is a meshless collocation method with global basis functions, which leads to finite-dimensional problems with full matrices. Numerical experiments for parabolic and elliptic PDEs performed by Kansa [15, 16, 20] and Golberg and Chen [21] show high accuracy and efficiency of the MQ scheme. Since the coefficients matrix is full, Kansa [22] suggested using domain decomposition and blending. Foley [23] presented a general formulation of the map and blend technique and Duval [24] showed that domain decomposition can be implemented straightforwardly, the computational effort is considerable decreased, and problems of ill-conditioning are alleviated. However, the global character of this interpolation scheme makes it difficult to apply it in large-scale engineering problems.

Zuppa [25] introduced a meshless local interpolation scheme (LOPI) that compares favorably with Hardy's interpolation. LOPI is *r-reproducing* and has a compactly supported basis. It is worth mentioning that it also verifies the delta-Kronecker property, greatly simplifying the introduction of essential boundary conditions. Roughly speaking, the scheme consists of local interpolation based in Optimal Point interpolants [26]. The main objective of this work is to investigate the use of LOPI in solving PDEs by collocation method.

The paper is organized as follows. In Section 2 we describe the formulation of LOPI and some of its mathematical properties. In Section 3, we give estimates for the interpolation error in LOPI. Section 4 describes the collocation method to discretize elliptic PDEs. In Section 5 we present results of numerical experiments with 2D elliptic problems, some of them of high complexity. Finally, in Section 6 we discuss our results and summarize some conclusions.

2. SCATTERED DATA INTERPOLATION WITH LOPI

The class of interpolants we shall describe was introduced by Zuppa [25] with some slight modifications. We denote $\|\cdot\|$ the Euclidean norm in n-dimensional space $\mathbb{R}^n$ and $B(\mathbf{y};r)$ the open ball $\{\mathbf{x}=(x_1,\ldots,x_n)\in\mathbb{R}^n:\ \|\mathbf{x}-\mathbf{y}\|<r\}$ with centre $\mathbf{y}$ and radius r. Also, we use standard multi-index notation: in particular, given any multi-index $v=(v_1,\ldots,v_n)\in\mathbb{N}^n$, $|\boldsymbol{v}|$ denotes the sum $v_1+\cdots+v_n$, and $D^v f$ the partial derivative $\partial^{|v|}/\partial x_1^{v_1}\cdots\partial x_n^{v_n} f$ for any sufficiently smooth function f.

2.1. Multiquadrics interpolation

Let $\mathbf{f}=(f_\beta)$, $\beta=1,\ldots,K$, be a set of given values at distinct points $\mathscr{X}=\{\mathbf{x}_1,\ldots,\mathbf{x}_K\}$, and g the *multiquadrics* (MQ) radial C^∞ function:

$$g(\mathbf{x}):=\sqrt{1+c_q\,\|\mathbf{x}\|^2},\quad \mathbf{x}\in\mathbb{R}^n$$

for a given constant $c_q>0$. We may construct an exact interpolant of data $\mathbf{f}$ as the linear combination $\sum_{\beta=1}^{K} a_\beta g(\mathbf{x}-\mathbf{x}_\beta)$ of translates of the basic function g by computing the coefficients a_β using

$$\mathbf{a}=V^{-1}\mathbf{f}$$

where the *Vandermondian* (or *Grammian*) V is given:

$$V=\begin{bmatrix} g(\mathbf{x}_1-\mathbf{x}_1) & \cdots & g(\mathbf{x}_K-\mathbf{x}_1)\\ \vdots & \ddots & \vdots\\ g(\mathbf{x}_1-\mathbf{x}_K) & \cdots & g(\mathbf{x}_K-\mathbf{x}_K)\end{bmatrix}$$

and it is assumed to be invertible.

Remark 1
An interpolant $\mathscr{O}$ is *r-reproducing* iff $\mathscr{O}[\mathbf{p}]=P\,\forall\,P$ polynomial of degree r and $\mathbf{p}=(P(\mathbf{x}_i))$, $i=0,\ldots,r$ is a vector of values of P at $r+1$ distinct points $\mathbf{x}_i$.

MQ interpolants constructed as above are not *reproducing*. This property, which is essential for convergence of the numerical method for PDEs solution, can be retrieved by augmentation using a Boolean sum of operators. In order to do this, we must build a projector Q onto the space of polynomials we wish to preserve.

2.2. Least-squares polynomial approximation

For the sake of simplicity, we describe only the case $r=1$. Nevertheless, in the examples we used also $r=2$ for solving elliptic PDEs. Therefore, from now on we consider the linear space of polynomials $\mathscr{P}_1$.

Let $\mathbf{x}_1$ be the *centre node*, with nodes $\mathbf{x}_2,\ldots,\mathbf{x}_K$ sprinkled around. We think of $\mathscr{P}_1$ as generated by a basis of monomials centred at $\mathbf{x}_1$; i.e. $\mathscr{P}_1$ is generated by

$$\{b_0=1,b_1,\ldots,b_p\}=\{1,x_1-x_{1,1},\ldots,x_n-x_{1,n}\}$$

Note that $p\equiv n$ since the degree of polynomials is limited to $r=1$.

We will search for a least-squares approximation to data $\mathbf{f}$, using a ('Taylor' expansion) linear polynomial of the form

$$P_1(\mathbf{x})=\sum_{0\leqslant|k|\leqslant 1} a_k\,(\mathbf{x}-\mathbf{x}_1)^k$$

where $k\in\mathbb{N}^n$.

It is easy to show that coefficients a_k in P_1 are given by

$$\mathbf{a} = Q_{\mathscr{X}}\,\mathbf{f}$$

where

$$Q_{\mathscr{X}} = (B_{\mathscr{X}}^{\mathrm{T}} B_{\mathscr{X}})^{-1} B_{\mathscr{X}}^{\mathrm{T}} \tag{1}$$

and where matrix $B_{\mathscr{X}}$ is written

$$B_{\mathscr{X}} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & x_{2,1} - x_{1,1} & \dots & x_{2,n} - x_{1,n} \\ \vdots & \vdots & & \vdots \\ 1 & x_{K,1} - x_{1,1} & \dots & x_{K,n} - x_{1,n} \end{pmatrix}$$

Remark 2
In order to verify the *r-reproducing* property, the rank of $B_{\mathscr{X}}$ should be equal to $\dim(P_r)$ to assure that polynomials of degree r are univocally determined by values in the set $\mathscr{X}$ [27]. We take it for granted.

2.3. The LOPI interpolant

By setting now

$$P_{\mathscr{X}} = V^{-1}(I - B_{\mathscr{X}} Q_{\mathscr{X}}) \tag{2}$$

together with vector functions

$$\mathbf{v}(\mathbf{x}) = (g(\mathbf{x} - \mathbf{x}_1), \dots, g(\mathbf{x} - \mathbf{x}_K))$$

and

$$\mathbf{b}(\mathbf{x}) = (1, b_1(\mathbf{x}), \dots, b_p(\mathbf{x}))$$

then, an LOPI operator is defined as follows:

$$\tilde{\mathscr{T}}_{\mathscr{X}}[\mathbf{f}](\mathbf{x}) = \langle \mathbf{v}(\mathbf{x}), P_{\mathscr{X}}\mathbf{f} \rangle + \langle \mathbf{b}(\mathbf{x}), Q_{\mathscr{X}}\mathbf{f} \rangle, \quad \mathbf{x} \in \mathbb{R}^n \tag{3}$$

where $\langle \cdot, \cdot \rangle$ is the Euclidean scalar product of vectors.

Remark 3
We remark that $\tilde{\mathscr{T}}_{\mathscr{X}}$ consists on a least-squares approximation of data $\mathbf{f}$ corrected by a multiquadrics to make the approximation interpolant. Clearly, if $f_\alpha = P(\mathbf{x}_\alpha)$, for $P(\mathbf{x}) \in \mathscr{P}_1$, the interpolant will exactly reproduce $P(\mathbf{x})$.

Remark 4
The form in which LOPI makes the MQ interpolation r-reproductive is algebraically more complicated than the approach proposed by Micchelli [28], who proposed the interpolant:

$$\mathscr{M}_{\mathscr{X}}[\mathbf{f}](\mathbf{x}) = \langle \mathbf{v}(\mathbf{x}), \mathbf{c}_v \rangle + \langle \mathbf{b}(\mathbf{x}), \mathbf{c}_b \rangle, \quad \mathbf{x} \in \mathbb{R}^n$$

where coefficients $\mathbf{c}_b, \mathbf{c}_v$ are obtained by solving the linear system

$$\begin{pmatrix} V & B_{\mathscr{X}} \\ B_{\mathscr{X}}^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{c}_v \\ \mathbf{c}_b \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix}$$

Numerical experiences show that this coefficients matrix is worse conditioned than the system to be solved in LOPI. Also, the dimension of the linear system is larger than in LOPI.

Remark 5
Another way to write (3) is as follows. Let $\{\mathbf{e}_\alpha\}_{\alpha=1}^K$ be the canonical basis of $\mathscr{F}$, that is, $\mathbf{e}_\alpha = (0,\ldots,0,1,0,\ldots,0)$ with 1 in the α-component, and $\varphi_\alpha := \tilde{\mathscr{T}}_{\mathscr{X}}[\mathbf{e}_\alpha]$, $\alpha = 1,\ldots,K$. By linearity, (3) is equal to

$$\tilde{\mathscr{T}}_{\mathscr{X}}[\mathbf{f}] = \sum_{\alpha=1}^{K} f_\alpha \varphi_\alpha \tag{4}$$

The latter expression has the standard form of an interpolant written as a linear combination of basis functions $\{\varphi_\alpha\}$. It is easy to see that $\varphi_\alpha(\mathbf{x}_\beta) = \delta_{\alpha\beta}$.

3. APPROXIMATING POWER OF THE LOCAL OPTIMAL POINT INTERPOLATOR

The rate of convergence of the interpolation formula (3) is investigated below. Again, for easiness of presentation, we shall be concerned only with the 1-*reproduction* case.

Let us consider a fixed node $\mathbf{x}_\alpha$ which is *centre* of a *cloud* ω_α of diameter d_α. The cloud has a set of nodes $\mathscr{S}(\alpha)$ (its *star*), with cardinality K_α. In order to simplify notation we shall omit the subscripts $\mathscr{S}(\alpha), \alpha$ in all involved terms. Therefore, $Q = Q_{S(\alpha)}$, $P = P_{S(\alpha)}$, $d = d_\alpha$, $K = K_\alpha$, etc.

We describe next the fundamental result which leads directly to the error estimates.

Theorem 1
There exist a number $C_N(\mathscr{S}(\alpha))$, which is a computable measure of the quality of the star $\mathscr{S}(\alpha)$, and a constant $C_Q(\alpha) = C_\alpha(n, K, C_N(\mathscr{S}(\alpha)))$, such that

$$|a_\eta| \leqslant C_Q(\alpha)\, d^{-|\eta|}\, \|\mathbf{f}\|, \quad 0 \leqslant |\eta| \leqslant 1,\ \mathbf{f} \in \mathbb{R}^K \tag{5}$$

where $\mathbf{a} = (a_\eta)_{0 \leqslant |\eta| \leqslant 1}$ is given by

$$\mathbf{a} = Q\mathbf{f} \tag{6}$$

Proof
We will write Equation (6) as

$$A\,\mathbf{a} = F\,\mathbf{f} \tag{7}$$

with $A = B^{\mathrm{T}}B$ and $F = B^{\mathrm{T}}$. Next, we introduce the change of co-ordinates $\mathbf{y} = \mathbf{x} - \mathbf{x}_\alpha$.

Let us make first the change of co-ordinates:

$$\mathbf{y}_j = \rho_j d \boldsymbol{\psi}_j, \quad j = 1,\ldots,K$$

where $\boldsymbol{\psi}_j$ is the vector of cosine directors, such that $\|\boldsymbol{\psi}_j\| = 1$, and where $0 \leqslant \rho_j \leqslant 1$. Then, we may write:

$$A = \begin{pmatrix} K & d\sum \rho_i \boldsymbol{\psi}_i^{\mathrm{T}} \\ d\sum \rho_i \boldsymbol{\psi}_i & d^2 \sum \rho_i^2 \boldsymbol{\psi}_i \boldsymbol{\psi}_i^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} A_{00} & dA_{01} \\ dA_{10} & d^2 A_{11} \end{pmatrix}$$

$$F = \begin{pmatrix} 1 & \dots & 1 \\ d\rho_1 \boldsymbol{\psi}_1 & \dots & d\rho_K \boldsymbol{\psi}_K \end{pmatrix} = \begin{pmatrix} F_0 \\ dF_1 \end{pmatrix}$$

with A_{00} a 1×1-matrix, A_{01} a $1 \times n$-matrix, etc. Also, F_0 is the $1 \times K$-matrix $(1, 1, \dots, 1)$ and F_1 is an $n \times K$-matrix.

Following this partition, system (6) is written as

$$\begin{pmatrix} A_{00} & dA_{01} \\ dA_{10} & d^2 A_{11} \end{pmatrix} \begin{pmatrix} \bar{a}_0 \\ \bar{\mathbf{a}}_1 \end{pmatrix} = \begin{pmatrix} F_0 \\ dF_1 \end{pmatrix} \mathbf{f}$$

where $\bar{\mathbf{a}}_i$ is the vector of components $(a_\eta)_{|\eta| = i}$. The solution to this system of equations can be written

$$\bar{a}_0 = \frac{1}{\bar{\alpha}} (F_0 - A_{01} A_{11}^{-1} F_1) \mathbf{f}$$

$$\bar{\mathbf{a}}_1 = \frac{1}{\bar{\alpha} d} A_{11}^{-1} (-A_{10} F_0 + (\bar{\alpha} I + A_{10} A_{01} A_{11}^{-1}) F_1) \mathbf{f} = \frac{1}{d} A_{11}^{-1} (F_1 \mathbf{f} - A_{10} \bar{a}_0)$$

with

$$\bar{\alpha} = A_{00} - A_{01} A_{11}^{-1} A_{10}$$

Note that matrices A_{ij}, F_i verify the following bounds:

$$\|F_0\| = K, \quad \|F_1\| \leqslant Kn$$

$$\|A_{00}\| = K, \quad \|A_{01}\| \leqslant Kn, \quad \|A_{11}\| \leqslant Kn^2$$

By defining the number $C_N(\mathscr{S}(\alpha))$ in the form

$$C_N(\mathscr{S}(\alpha)) := \max \left(\left| \frac{1}{\bar{\alpha}} \right|, \|A_{11}^{-1}\| \right)$$

the proof of the theorem follows easily.

Several interesting remarks related to $C_N(\mathscr{S}(\alpha))$ is as follow:

Remark 6
$C_N(\mathscr{S}(\alpha))$ depends only on $(\rho_i, \boldsymbol{\psi}_i)$. Then, it is invariant by translations and dilations $\mathbf{x} \to \lambda(\mathbf{x} - \mathbf{c}) + \mathbf{c}$, $\lambda \in \mathbb{R}$. I.e., $C_N(\mathscr{S}(\alpha))$ is really a measure of the geometrical quality of the *stars*, independently of their size.

Remark 7
In general, $C_N(\mathscr{S}(\alpha))$ has a very stable behaviour, even in border nodes. In several numerical experiments with random points distributions (obtained by perturbing randomly uniform

distributions) a mean of 1.28 has been obtained for cond(A_{11}). Also, note that $|1/\bar{\alpha}| = 1$ for a perfectly centred and symmetric star, and that its value grows with loss of symmetry, e.g. at border stars.

Remark 8
Result (5) can be generalized for higher reproducibility conditions [29]. However, condition numbers tend to be higher in these cases, especially at border nodes.

Remark 9
We will consider coefficient c_q appearing in the radial function g, as a function of the *cloud diameter* d_α. More precisely, we define

$$c_q(\alpha) = \tilde{c}_q / d_\alpha^2 \tag{8}$$

for a given constant $\tilde{c}_q > 0$. In order to get acceptable results in the approximating power of the interpolant, it is important to keep controlled similarly both matrix norms $\|P_{\mathscr{S}(\alpha)}\|$ and $\|Q_{\mathscr{S}(\alpha)}\|$ (see next section). Proposal (8) appears to be a good choice in this direction.

After recalling the expression of the operator P, Equation (2), it is not difficult to demonstrate the following:

Corollary 1
There exists a constant $C_P(\alpha) = C'_\alpha(n, K, C_N(\mathscr{S}(\alpha)))$, such that

$$\|P\| \leqslant C_P(\alpha)\|V^{-1}\|$$

Proof
Given $\mathbf{f} \in \mathbb{R}^K$, using (2) we have

$$\|P\mathbf{f}\| \leqslant \|V^{-1}\|\,(\|\mathbf{f}\| + \|B\mathbf{a}\|)$$

where $\mathbf{a} = Q\mathbf{f}$. Each component $(B\mathbf{a})_j$, $j = 1, \ldots, K$, of $B\mathbf{a}$ has the form

$$(B\mathbf{a})_j = a_0 + \sum_{|\eta|=1} a_\eta (\mathbf{x}_{\beta_j} - \mathbf{x}_\alpha)^\eta$$

Using (5) and the fact that $\|\mathbf{x}_{\beta_j} - \mathbf{x}_\alpha\| \leqslant d$, we get $|(B\mathbf{a})_j| \leqslant (n+1)C_Q(\alpha)\|\mathbf{f}\|$, $j = 1, \ldots, K$, and $\|B\mathbf{a}\| \leqslant \sqrt{K}(n+1)C_Q(\alpha)\|\mathbf{f}\|$. The proof is finished by setting

$$C_P(\alpha) = 1 + \sqrt{K}(n+1)C_Q(\alpha) \qquad \square$$

We are now ready for obtaining estimates for the operator $\tilde{\mathscr{T}}_{\mathscr{S}(\alpha)}$.

Corollary 2
There exists a constant $C_0(\alpha) = C_0(\tilde{c}_q, C_P(\alpha), C_Q(\alpha))$ such that

$$|\tilde{\mathscr{T}}_{\mathscr{S}(\alpha)}[\mathbf{f}](\mathbf{x})| \leqslant C_0(\alpha)\|\mathbf{f}\| \tag{9}$$

for every $\mathbf{x} \in \omega_\alpha$ and $\mathbf{f} \in \mathbb{R}^K$.

Proof
We have

$$\tilde{\mathcal{I}}_{\mathcal{S}(\alpha)}[\mathbf{f}](\mathbf{x}) = \langle \mathbf{v}(\mathbf{x}), P\mathbf{f} \rangle + \langle \mathbf{b}(\mathbf{x}), Q\mathbf{f} \rangle$$

where

$$\mathbf{v}(\mathbf{x}) = (g(\mathbf{x} - \mathbf{x}_{\beta_1}), \ldots, g(\mathbf{x} - \mathbf{x}_{\beta_K})), \quad \beta_j \in \mathcal{S}(\alpha)$$

and

$$\langle \mathbf{b}(\mathbf{x}), Q\mathbf{f} \rangle = a_0 + \sum_{|\eta|=1} a_\eta (\mathbf{x} - \mathbf{x}_{\beta_1})^\eta$$

where we note $\beta_1 \equiv \alpha$ for convenience. Since

$$g(\mathbf{x} - \mathbf{x}_{\beta_j}) = \sqrt{1 + \left(\frac{\tilde{c}_q}{d^2}\right) \|\mathbf{x} - \mathbf{x}_{\beta_j}\|^2} \leqslant \sqrt{1 + \tilde{c}_q}, \quad j = 1, \ldots, K$$

it follows that $\|\mathbf{v}(\mathbf{x})\| \leqslant \sqrt{K(1 + \tilde{c}_q)}$ and

$$|\langle \mathbf{v}(\mathbf{x}), P\,\mathbf{f} \rangle| \leqslant \sqrt{K(1 + \tilde{c}_q)}\, C_P(\alpha) \|V^{-1}\|\, \|\mathbf{f}\|$$

By applying a similar reasoning to the second term, we get

$$|\langle \mathbf{b}(\mathbf{x}), Q\,\mathbf{f} \rangle| \leqslant (n+1) C_Q(\alpha) \|\mathbf{f}\|$$

Summing up the latter two inequalities, we have

$$|\tilde{\mathcal{I}}_{\mathcal{S}(\alpha)}[\mathbf{f}](\mathbf{x})| \leqslant (\sqrt{K(1 + \tilde{c}_q)}\, C_P(\alpha) \|V^{-1}\| + (n+1) C_Q(\alpha)) \|\mathbf{f}\| \qquad \square$$

We also need estimates for the derivatives of $\tilde{\mathcal{I}}_{\mathcal{S}(\alpha)}[\mathbf{f}]$.

Corollary 3
There exists a constant $C_1(\alpha) = C_1(\tilde{c}_q, C_P(\alpha), C_Q(\alpha))$ such that, for every multi-index η, $|\eta| = 1$. we have

$$|D^\eta \tilde{\mathcal{I}}_{\mathcal{S}(\alpha)}[\mathbf{f}](\mathbf{x})| \leqslant C_1(\alpha)\, d^{-1} \|\mathbf{f}\| \tag{10}$$

for every $\mathbf{x} \in \omega_\alpha$ and $\mathbf{f} \in \mathbb{R}^K$.

Proof
The proof follows easily proceeding as above and taking into account that

$$D^\eta \tilde{\mathcal{I}}_{\mathcal{S}(\alpha)}[\mathbf{f}](\mathbf{x}) = \langle D^\eta \mathbf{v}(\mathbf{x}), P\,\mathbf{f} \rangle + \langle \mathbf{i}_\eta, Q\mathbf{f} \rangle$$

where $\mathbf{i}_\eta = (0, \eta)$ and

$$D^\eta g(\mathbf{x} - \mathbf{x}_{\beta_j}) = \left(\frac{\tilde{c}_q}{d^2}\right) (g(\mathbf{x} - \mathbf{x}_{\beta_j}))^{-1} (\mathbf{x} - \mathbf{x}_{\beta_j})^\eta, \quad j = 1, \ldots, K \qquad \square$$

A function f is said to be of class $C^{q,1}$ in $\overline{\omega}_\alpha$ if and only if f is of class C^q in $\overline{\omega}_\alpha$ and the partial derivatives $D^k f$ of order q ($|k|=q$) are Lipschitz continuous in $\overline{\omega}_\alpha$. The seminorm $|.|_{q,1}$ is defined as

$$|f|_{q,1} = \sup\left\{ \frac{|D^k f(\mathbf{x}_1) - D^k f(\mathbf{x}_2)|}{\|\mathbf{x}_1 - \mathbf{x}_2\|} :\ \mathbf{x}_1, \mathbf{x}_2 \in \overline{\omega}_\alpha, \mathbf{x}_1 \neq \mathbf{x}_2,\ |k| = q \right\}$$

The following error estimates will be stated without proof (a demonstration is given in [25]).

Theorem 2
If $f \in C^{q,1}(\overline{\omega}_\alpha)$ and $\tilde{\mathscr{F}}_{\mathscr{S}(\alpha)}$ is q-reproducing, then there exist constants $\tilde{\mathscr{C}}_0, \ldots, \tilde{\mathscr{C}}_q > 0$ such that

$$\|D^\eta (f - \tilde{\mathscr{F}}_{\mathscr{S}(\alpha)}[\mathbf{f}])\|_{L^\infty} \leqslant \tilde{\mathscr{C}}_\ell d^{q+1-\ell} |f|_{q,1}, \quad |\eta| = \ell, \quad \ell = 0, \ldots, q \tag{11}$$

with $\mathbf{f} = (f(x_\alpha)_{\alpha=1}^K)$.

Remark 10
The estimates above tell us that the interpolation formula approximates the function itself with $O(d^{q+1})$ accuracy, its first order derivatives with accuracy $O(d^q)$, and so on. Note that in order to approximate derivatives of order q, we need to use a q-reproducing interpolant. Nevertheless, note also that the experiences shown in the examples section indicate that better approximation properties are being obtained, and that this estimation may be non-optimal.

4. APPROXIMATE SOLUTION TO PDES BY COLLOCATION

The procedure will be illustrated with elliptic PDEs. For the sake of simplicity, we assume the interior and boundary (Neumann and Dirichlet) operators to be linear and defining a well-posed elliptic boundary-value problem:

$$\begin{aligned} Pu(\mathbf{x}) &= f(\mathbf{x}), \quad \mathbf{x} \in \Omega \\ B_{\mathrm{N}} u(\mathbf{x})|_{\Gamma_{\mathrm{N}}} &= s_{\mathrm{N}}(\mathbf{x}) \\ B_{\mathrm{D}} u(\mathbf{x})|_{\Gamma_{\mathrm{D}}} &= s_{\mathrm{D}}(\mathbf{x}) \end{aligned} \tag{12}$$

Here, Ω is an open bounded domain in $\mathbb{R}^n$, B_{D} is a pure Dirichlet operator and B_{N} either a Neumann or a mixed operator.

Let $\mathscr{Q}^N$ denote an arbitrarily chosen set of N points $\mathbf{x}_\alpha \in \bar{\Omega}$ referred to as *nodes*:

$$\mathscr{Q}^N = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}, \quad \mathbf{x}_\alpha \in \bar{\Omega}$$

Let $\mathscr{I}_N := \{\omega_\alpha\}_{\alpha=1}^N$ denote a finite open covering of $\bar{\Omega}$ consisting of N *clouds* ω_α such that $\mathbf{x}_\alpha \in \omega_\alpha$, $\alpha = 1, \ldots, N$ constitutes the *centre* of the star, and

$$\bar{\Omega} \subset \bigcup_{\alpha=1}^N \omega_\alpha$$

After reordering, we can partition the set of nodes $\mathscr{Q}^N$ in the form

$$\mathscr{Q}^N = \{(\mathbf{x}_\beta)|_{\beta=1,\ldots,M_1} \subset \Omega, (\mathbf{x}_\beta)|_{\beta=M_1+1,\ldots,M_2} \subset \Gamma_{\mathrm{N}}, (\mathbf{x}_\beta)|_{\beta=M_2+1,\ldots,N} \subset \Gamma_{\mathrm{D}}\}$$

Let $\mathscr{F} := \mathbb{R}^N$ be the set of function values $\mathbf{f} = (f_\alpha)_{\alpha=1}^N$ evaluated at nodes $\{\mathbf{x}_\alpha\}$. If $\mathscr{G}$ is a subset of $\mathscr{Q}^N$, $\mathscr{F}_\mathscr{G}$ is the subspace $\{(f_\beta)\colon\ \mathbf{x}_\beta \in \mathscr{G}\}$ and $p_\mathscr{G} : \mathscr{F} \to \mathscr{F}_\mathscr{G}$ is the canonical linear projection.

For every node $\mathbf{x}_\alpha$, $\alpha = 1, \ldots, N$, let $\mathscr{S}(\alpha)$ be the subset of nodes in ω_α (the *star*), and $\mathscr{T}_\alpha : \mathscr{F} \to C^\infty(\mathbb{R}^n)$ the linear interpolating operator defined by

$$\mathscr{T}_\alpha := \tilde{\mathscr{T}}_{\mathscr{S}(\alpha)} \circ p_{\mathscr{S}(\alpha)}$$

At each cloud ω_α, we will approximate the solution of (12) in the form

$$u_h(\mathbf{x}) = \mathscr{T}_\alpha[\mathbf{u}](\mathbf{x}) = \sum_{i=1}^{M_2} u_i \cdot \varphi_i^\alpha(\mathbf{x}) + S_\mathrm{D}^\alpha(\mathbf{x}), \quad \mathbf{x} \in \omega_\alpha, \quad \alpha = 1, \ldots, N \tag{13}$$

where $S_\mathrm{D}^\alpha(\mathbf{x}) = \sum_{i=M_2+1}^N s_\mathrm{D}(\mathbf{x}_i) \cdot \varphi_i^\alpha(\mathbf{x})$.

Substituting $u_h(\mathbf{x})$ into (12) and using collocation at the nodes of $\mathscr{Q}^N$, we obtain the linear algebraic system

$$\begin{aligned} &\sum_{i=1}^{M_2} u_i \cdot P\varphi_i^\alpha(\mathbf{x}_\alpha) = f(\mathbf{x}_\alpha) - PS_\mathrm{D}^\alpha(\mathbf{x}_\alpha), \quad \alpha = 1, \ldots, M_1 \\ &\sum_{i=1}^{M_2} u_i \cdot B_\mathrm{N}\varphi_i^\alpha(\mathbf{x}_\alpha) = s_\mathrm{N}(\mathbf{x}_\alpha) - B_\mathrm{N}S_\mathrm{D}^\alpha(\mathbf{x}_\alpha), \quad \alpha = M_1 + 1, \ldots, M_2 \end{aligned} \tag{14}$$

4.1. Computational cost of the method

The computational cost of the LOPI discretization method is evaluated next, for the case of solving a scalar PDE (e.g. the Laplace equation).

In order to minimize cost, computations at each node $\mathbf{x}_\alpha$ of the discretization, are organized in the following form:

$$P\boldsymbol{\varphi}^\alpha(\mathbf{x}_\alpha) = \langle V^{-1}P\mathbf{v}(\mathbf{x}_\alpha), (I - B_\mathscr{X}Q_\mathscr{X})\rangle + \langle P\mathbf{b}(\mathbf{x}_\alpha), Q_\mathscr{X}\rangle \tag{15}$$

Then, the following operations must be performed per node:

1. Computation of the Vandermondian V has a cost of $O(nK^2)$, with n the dimension of the space, and K the number of points inside the cloud.
2. The evaluation of matrices $Q_\mathscr{X}$ and $(I - B_\mathscr{X}Q_\mathscr{X})$ has a cost of $O(LK^2)$, where $L = (\Pi_{k=1,n}(r+k))/(\Pi_{k=1,n}k)$ is the number of terms in the Pascal triangle. It is a function of r, the reproducing order of the approximation, and of the dimension of the space n.
3. Evaluation of $V^{-1}P\mathbf{v}(\mathbf{x}_\alpha)$ involves an order of $O(K^3)$ operations.
4. Finally, the internal products $\langle V^{-1}P\mathbf{v}(\mathbf{x}_\alpha), (I - B_\mathscr{X}Q_\mathscr{X})\rangle$ and $\langle P\mathbf{b}(\mathbf{x}_\alpha), Q_\mathscr{X}\rangle$ require $O(K^2)$ and $O(LK)$ operations, respectively.

Therefore, the total number of operations *per node* of the discretization can be estimated as $O([L + K + n + 2]K^2)$ operations. Then, the evaluation of the coefficients matrix requires $O([L + n + K + 2]NK^2)$ operations. This matrix has N rows, and a semibandwidth equal to K, thus requiring $O(NK^2)$ operations to solve the system of equations.

We see that the cost of computations is directly influenced by parameter K, which is the number of nodes entering in the cloud. This parameter is a function of the cloud radius r_ω. This characteristic is common to most meshless methods, which have a larger overlapping between coefficients than in finite element methods. For this reason, meshless methods are inherently more expensive than the latter.

4.2. Neumann or mixed problems

Although formally straightforward, the direct implementation of this kind of boundary conditions as in Equation (14) creates some side effects that may reduce the accuracy of the collocation method. This is mainly because:

1. The discretization of the boundary condition at the boundary node precludes discretization of the principal governing equation at this node (one equation is generated per degree of freedom). Thus, only internal nodes support the governing equation.
2. The boundary star $\mathscr{S}$ tends to be of strongly unsymmetrical shape, with the centre of mass shifted towards the domain interior.

Two different techniques are described below to avoid this drawback.

4.2.1. Additional nodes placement. As long as the contour does not present singular points, the following strategy may be employed to handle the Neumann boundary condition. At each node $\mathbf{x} \in \Gamma_N$, we add an associated external node in the direction of the outward normal of Γ_N, at a distance that agrees with a local measure of the grid-size (and, of course, with the geometry of $\partial\Omega$).

Thus we increase the number of nodes, one for every node $\mathbf{x} \in \Gamma_N$. Then, we follow the customary algorithm in classical FDM: two equations are imposed at each boundary node in Γ_N, the first equation resulting from the boundary condition, and a second one from the PDE. This strategy was followed successfully in all our numerical experiments.

4.2.2. Additional basis function. The procedure outlined in the previous section complicates the implementation of the method, especially in complex geometries. An alternative technique consists into augmenting the number of basis functions at every Neumann node as follows:

At every *node* $\mathbf{x}_\gamma \in S_\alpha \cap \Gamma_N$, we define a basis function $\psi_\gamma^\alpha(\mathbf{x})$ using:

$$\psi_\gamma^\alpha(\mathbf{x}) = \langle \mathbf{n}(\mathbf{x}_\gamma), \mathbf{x} - \mathbf{x}_\gamma \rangle \varphi_\gamma^\alpha(\mathbf{x})$$

where $\mathbf{n}(\mathbf{x}_\gamma)$ is the normal vector in $\mathbf{x}_\gamma \in \Gamma_N$. It can be observed that $\psi_\gamma^\alpha(\mathbf{x}_\beta) = 0$ for all $\mathbf{x}_\beta \in \mathscr{Q}^N$ and that $\partial/\partial n\, \psi_\gamma^\alpha(\mathbf{x}_\gamma) = \mathbf{n} \cdot \nabla \psi_\gamma^\alpha(\mathbf{x}_\gamma) = 1$.

If $S_\alpha \cap \Gamma_N \neq \emptyset$, we may improve approximation (13) by

$$u_h(\mathbf{x}) = \sum_{i=1}^{M_2} u_i \cdot \varphi_i^\alpha(\mathbf{x}) + \sum_{i=M_1+1}^{M_2} g_i \cdot \psi_i^\alpha(\mathbf{x}) + S_D^\alpha(\mathbf{x}), \quad \mathbf{x} \in \omega_\alpha, \quad \alpha = 1, \ldots, N \tag{16}$$

where $S_{\mathrm{D}}^{\alpha}(\mathbf{x}) = \sum_{i=M_2+1}^{N} s_{\mathrm{D}}(\mathbf{x}_i) \cdot \varphi_i^{\alpha}(\mathbf{x})$. Again, two equations are imposed at every *node* $\mathbf{x}_\alpha \in \Gamma_{\mathrm{N}}$.

$$\begin{aligned} &\sum_{i=1}^{M_2} u_i \cdot P\varphi_i^{\alpha}(\mathbf{x}_\alpha) + \sum_{i=M_1+1}^{M_2} g_i \cdot P\psi_i^{\alpha}(\mathbf{x}_\alpha) = f(\mathbf{x}_\alpha) - PS_{\mathrm{D}}^{\alpha}(\mathbf{x}_\alpha), \quad \alpha = 1, \ldots, M_2 \\ &\sum_{i=1}^{M_2} u_i \cdot B_{\mathrm{N}}\varphi_i^{\alpha}(\mathbf{x}_\alpha) + \sum_{i=M_1+1}^{M_2} g_i \cdot B_{\mathrm{N}}\psi_i^{\alpha}(\mathbf{x}_\alpha) = s_{\mathrm{N}}(\mathbf{x}_\alpha) - B_{\mathrm{N}}S_{\mathrm{D}}^{\alpha}(\mathbf{x}_\alpha), \quad \alpha = M_1 + 1, \ldots, M_2 \end{aligned} \tag{17}$$

This technique is somehow similar to a *p-refinement* at the Neumann nodes.

In Section 5.3.2 we present a comparison of numerical results obtained for an example consisting on a Poisson equation with mixed boundary conditions on a unit square domain. We computed the error of the solution when using the three mentioned techniques. The error computed with the technique of additional nodes placement is almost the same as that obtained with the additional basis function. On the other hand, when using a standard discretization (Equations (14)), the error is evidently larger.

5. NUMERICAL EXPERIMENTS

We shall now perform several numerical tests to investigate the approximating properties of the collocation method (14). We first make all settings in our experiments explicit.

5.1. Cloud ω_α and constant $\tilde{c}_q$

Let $(\mathscr{Q}^N, h)$ be a uniform mesh over $\bar{\Omega}$. The shape of the cloud ω_α can be quite arbitrary. However, in this work, ω_α will always be an open ball $B_{r_\alpha}(\mathbf{x}_\alpha; r_\alpha)$, $r_\alpha > 0$. A number $r_\omega > 1$ is selected such that

$$\omega_\alpha = B(\mathbf{x}_\alpha; r_\omega h) \tag{18}$$

In our tests $r_\omega = 2.2$ always. This selection assures that clouds ω_α tie together and there are enough nodes in $\mathscr{S}(\alpha)$. Constant $\tilde{c}_q$ in Equation (8) was taken equal to 0.08. These choices were adopted after experimentation on several test cases. For instance, Figure 1 displays the variation of error when varying parameter $\tilde{c}_q$, for an example in which a Poisson equation with mixed boundary conditions is solved on a unit square domain (see Section 5.3.2 for a full description of the problem). We may appreciate that $\tilde{c}_q = 0.08$ is here an appropriate value for computations, value which was observed in most performed tests. Similar tests were conducted to determine an optimum value of r_ω.

5.2. Error measures

Errors have been computed at the discretization points. Since our examples are all two-dimensional, we have measured errors not only in function u but also in the derivatives u_x and u_y. The different error measures evaluated in all tests are summarized below:

$$eu_\infty = \frac{1}{\max_{\beta=1,\ldots,M_2} |u(\mathbf{x}_\beta)|} \max_{\beta=1,\ldots,M_2} |(u - u_h)(\mathbf{x}_\beta)|$$

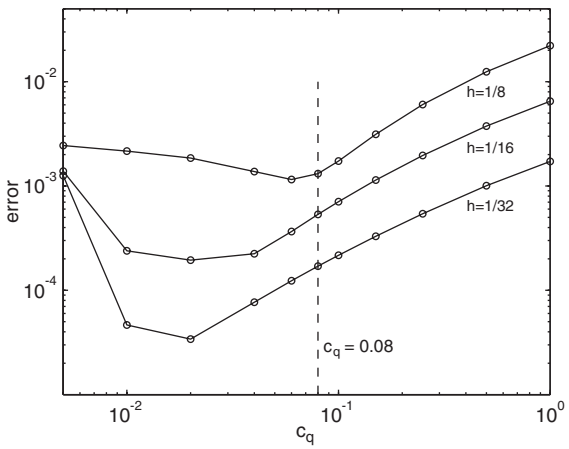

Figure 1. Variation of error for different values of parameter $\tilde{c}_q$, for the model problem of Section 5.3.2, with uniform point spacing and quadratic polynomial reproduction.

$$eu_{x\infty} = \frac{1}{\max_{\beta=1,\ldots,M_2} |u_x(\mathbf{x}_\beta)|} \max_{\beta=1,\ldots,M_2} |(u_x - u_{h,x})(\mathbf{x}_\beta)|$$

$$eu_{y\infty} = \frac{1}{\max_{\beta=1,\ldots,M_2} |u_y(\mathbf{x}_\beta)|} \max_{\beta=1,\ldots,M_2} |(u_y - u_{h,y})(\mathbf{x}_\beta)|$$

$$eu_{\ell^2} = \frac{1}{\max_{\beta=1,\ldots,M_2} |u(\mathbf{x}_\beta)|} \sqrt{\frac{1}{M_2} \sum_{\beta=1,\ldots,M_2} |(u - u_h)(\mathbf{x}_\beta)|^2}$$

$$eu_{x\ell^2} = \frac{1}{\max_{\beta=1,\ldots,M_2} |u_x(\mathbf{x}_\beta)|} \sqrt{\frac{1}{M_2} \sum_{\beta=1,\ldots,M_2} |(u_x - u_{h,x})(\mathbf{x}_\beta)|^2}$$

$$eu_{y\ell^2} = \frac{1}{\max_{\beta=1,\ldots,M_2} |u_y(\mathbf{x}_\beta)|} \sqrt{\frac{1}{M_2} \sum_{\beta=1,\ldots,M_2} |(u_y - u_{h,y})(\mathbf{x}_\beta)|^2}$$

where M_2 is the number of unknowns in the discretization (see Equation (13)).

5.3. *Model problems and results*

Several elliptic two-dimensional problems have been solved to illustrate the performance of the method. Tests with both random and on uniformly spaced nodes were performed. In the former case, nodes were generated by adding a random perturbation of value $0.25h$ to a uniform grid with h-spacing. Errors displayed in tables in the randomly distributed points case, correspond to averages over ten runs. The measure h of random grids on the unit square is defined as

$$h = \frac{\text{length(side)}}{\sqrt{N} - 1} = \frac{1}{\sqrt{N} - 1} \tag{19}$$

Table I. Model 1. Random grids.

	h	eu_∞	$eu_{x\infty}$	$eu_{y\infty}$	eu_{ℓ^2}	$eu_{x\ell^2}$	$eu_{y\ell^2}$
	1/8	6.267e − 4	7.305e − 3	6.551e − 3	2.364e − 4	1.767e − 3	1.596e − 3
R1	1/16	2.164e − 4	3.659e − 3	3.127e − 3	7.823e − 5	6.082e − 4	5.730e − 4
	1/32	1.078e − 4	1.773e − 3	1.780e − 3	3.782e − 5	2.583e − 4	2.576e − 4
	1/8	3.133e − 4	3.246e − 3	5.310e − 3	1.364e − 4	7.858e − 4	1.121e − 3
R2	1/16	7.831e − 5	1.320e − 3	1.406e − 3	2.422e − 5	1.928e − 4	1.967e − 4
	1/32	1.602e − 5	3.205e − 4	2.707e − 4	5.732e − 6	3.995e − 5	3.873e − 5

Note: Mean values of errors over 10 runs. Errors from computations with 1-reproducing interpolants (R1) and 2-reproducing interpolants (R2) are displayed.

Table II. Model 1. Uniform square grid.

	h	eu_∞	$eu_{x\infty}$	$eu_{y\infty}$	eu_{ℓ^2}	$eu_{x\ell^2}$	$eu_{y\ell^2}$
	1/8	3.016e − 4	3.667e − 3	3.667e − 3	2.080e − 4	1.106e − 3	1.106e − 3
R1	1/16	9.098e − 5	1.854e − 3	1.854e − 3	5.040e − 5	3.348e − 4	3.348e − 4
	1/32	2.617e − 5	9.282e − 4	9.282e − 4	1.250e − 5	1.019e − 4	1.019e − 4
	1/8	1.503e − 4	5.956e − 4	5.956e − 4	9.900e − 5	3.237e − 4	3.237e − 4
R2	1/16	1.831e − 5	1.296e − 4	1.296e − 4	1.176e − 5	8.528e − 5	8.528e − 5
	1/32	2.204e − 6	3.033e − 5	3.033e − 5	1.427e − 6	2.258e − 5	2.258e − 5

Note: Computed errors. Errors from computations with 1-reproducing interpolants (R1) and 2-reproducing interpolants (R2) are displayed.

5.3.1. Model 1: Laplace equation with Dirichlet boundary condition. This test is a very well-behaved elliptic problem. Most methods work fine in this case.

$$u_{xx} + u_{yy} = 0, \quad \Omega = \{(x, y) | \, 0 < x, y < 1\}$$

$$u|_{\partial\Omega} = g(x, y)$$

where the exact solution is $u(x, y) = g(x, y) = -x^3 - y^3 + 3x^2 y + 3xy^2$.

Errors for computations with random and uniform points distributions (9×9, 17×17 and 33×33) are summarized in Tables I and II, respectively. In both cases, computations with 1-reproducing interpolants (R1) and 2-reproducing interpolants (R2) were performed.

Figure 2 displays three samples of random grids used in computations. Convergence logarithmic plots for the solution and its x-derivative are shown in Figure 3. In the 1-reproduction case, the convergence rate for the solution u is 2.02 with uniformly spaced points, and 1.32 with randomly spaced ones. Solution derivatives converge at a rate equal to 1.72 and 1.38, in the uniform and random spacing, respectively. When passing to the 2-reproduction case, the convergence rate increased to 3.05 when computing the solution with uniformly spaced points, and 2.29 with randomly spaced ones (mean value over ten runs). Derivatives converge at a rate equal to 1.92 and 2.27 with uniformly spaced and randomly spaced points, respectively.

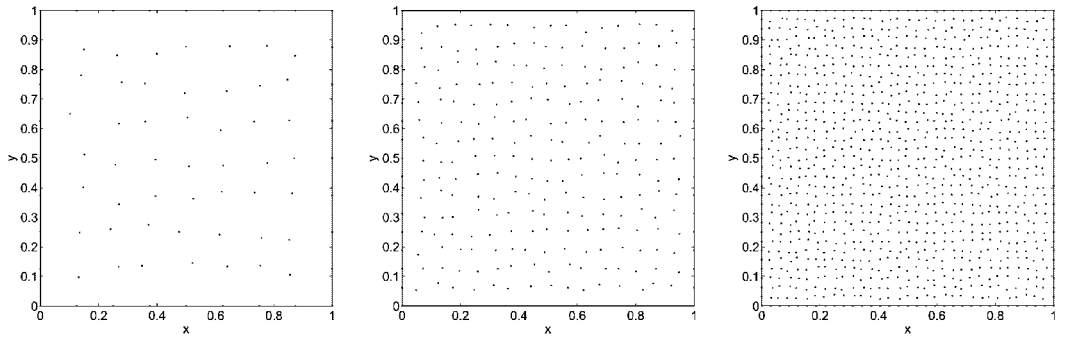

Figure 2. Model 1. Random grids. Points distribution samples for different point densities.

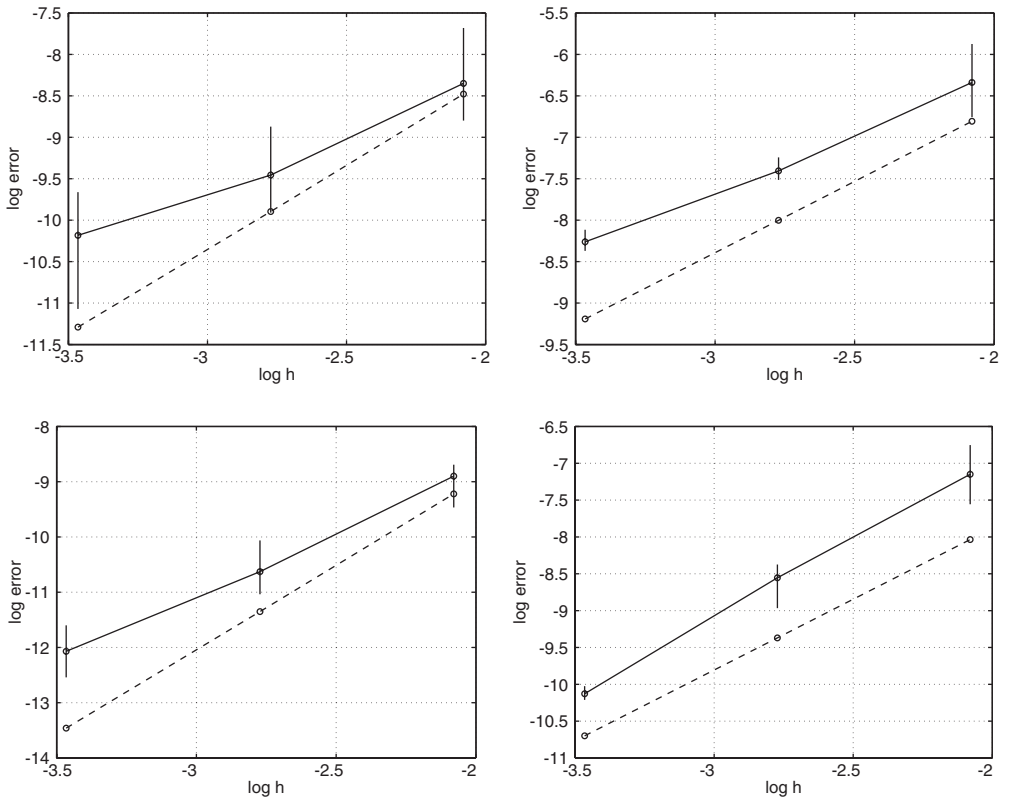

Figure 3. Model 1. Convergence of LOPI method for random grids (continuous line) and uniform grids (dashed line). Top plots: 1-reproduction case; bottom plots: 2-reproduction case. Plot on the left: u error; plots on the right: u_x error. Vertical bars show min/max regions for ten different random grids.

This example has been treated by Aluru, with a modified Reproducing Kernel Particle Method (RKPM) of his own [13]. Table III compares errors from computations with Aluru scheme and ours on uniformly spaced points grids. We may appreciate that LOPI with

Table III. Model 1.

		Function			Derivatives		
		LOPI			LOPI		
h	$\ln(h)$	R1	R2	RKPM	R1	R2	RPKM
1/8	−2.08	−8.48	−9.22	−8.06	−6.81	−8.04	−5.58
1/16	−2.77	−9.90	−11.35	−9.71	−8.00	−9.37	−7.11
1/32	−3.46	−11.29	−13.45	−11.80	−9.19	−10.70	−8.82

Note: Comparison of errors obtained by LOPI method with those obtained by RKPM of Aluru [13].

Table IV. Model 3. Random grid.

	h	eu_∞	$eu_{x\infty}$	$eu_{y\infty}$	$eu_{\ell 2}$	$eu_{x\ell 2}$	$eu_{y\ell 2}$
	1/8	1.398e − 1	6.1e − 2	1.150e − 1	3.884e − 2	1.796e − 2	2.841e − 2
R1	1/16	7.887e − 3	7.841e − 3	9.788e − 3	2.144e − 3	1.614e − 3	2.174e − 3
	1/32	7.111e − 3	8.964e − 3	2.013e − 2	1.417e − 3	1.097e − 3	1.662e − 3
	1/8	1.839e − 1	7.124e − 2	1.555e − 1	4.500e − 2	1.974e − 2	3.783e − 2
R2	1/16	1.791e − 2	1.255e − 2	2.137e − 2	3.635e − 3	2.462e − 3	3.364e − 3
	1/32	4.600e − 3	5.407e − 3	7.499e − 3	9.892e − 4	6.861e − 4	9.281e − 4

Note: Mean values of errors over 10 runs. Errors from computations with 1-reproducing interpolants (R1) and 2-reproducing interpolants (R2) are displayed.

1-reproduction interpolants gives very similar convergence rates as Aluru scheme, while the convergence rate for the 2-reproduction case is considerably better than that of Aluru for this test problem.

5.3.2. Model 2: Poisson equation with mixed boundary conditions. This test example is a Poisson equation with a harmonic source on a unit square domain. Mixed boundary conditions are imposed: on two sides of the square we impose homogeneous Dirichlet boundary conditions while on the other sides we impose homogeneous Neumann boundary conditions.

$$u_{xx} + u_{yy} = -8\pi^2 \cos(2\pi x) \sin(2\pi y), \quad \Omega = \{(x, y) | \, 0 < x, y < 1\}$$
$$u = 0 \quad \text{on } y = 0, 1$$
$$u_x = 0 \quad \text{on } x = 0, 1$$

The exact solution is written

$$u(x, y) = \cos(2\pi x) \sin(2\pi y)$$

Results for random and uniform points distributions (9×9, 17×17 and 33×33) are summarized in Tables IV and V, respectively. Convergence logarithmic plots for the solution and its x-derivative are shown in Figure 4.

Figure 5 displays a series of approximated solutions computed with uniformly spaced points for the 1-reproduction case. Figures 6 and 7 give the distribution of errors for the 1- and 2-reproduction cases, respectively.

Table V. Model 3. Uniform grid.

	h	eu_∞	$eu_{x\infty}$	$eu_{y\infty}$	eu_{ℓ^2}	$eu_{x\ell^2}$	$eu_{y\ell^2}$
	1/8	3.700e − 3	4.148e − 3	1.039e − 2	1.966e − 3	2.285e − 3	3.613e − 3
R1	1/16	2.021e − 3	2.072e − 3	3.811e − 3	6.764e − 4	1.095e − 3	1.375e − 3
	1/32	1.660e − 3	1.512e − 3	2.247e − 3	5.872e − 4	8.161e − 4	8.622e − 4
	1/8	2.612e − 3	4.449e − 3	2.035e − 2	1.314e − 3	2.207e − 3	5.437e − 3
R2	1/16	1.288e − 3	8.818e − 4	2.871e − 3	5.341e − 4	4.016e − 4	5.205e − 4
	1/32	3.740e − 4	2.503e − 4	6.050e − 4	1.702e − 4	1.352e − 4	1.240e − 4

Note: Computed errors. Errors from computations with 1-reproducing interpolants (R1) and 2-reproducing interpolants (R2) are displayed.

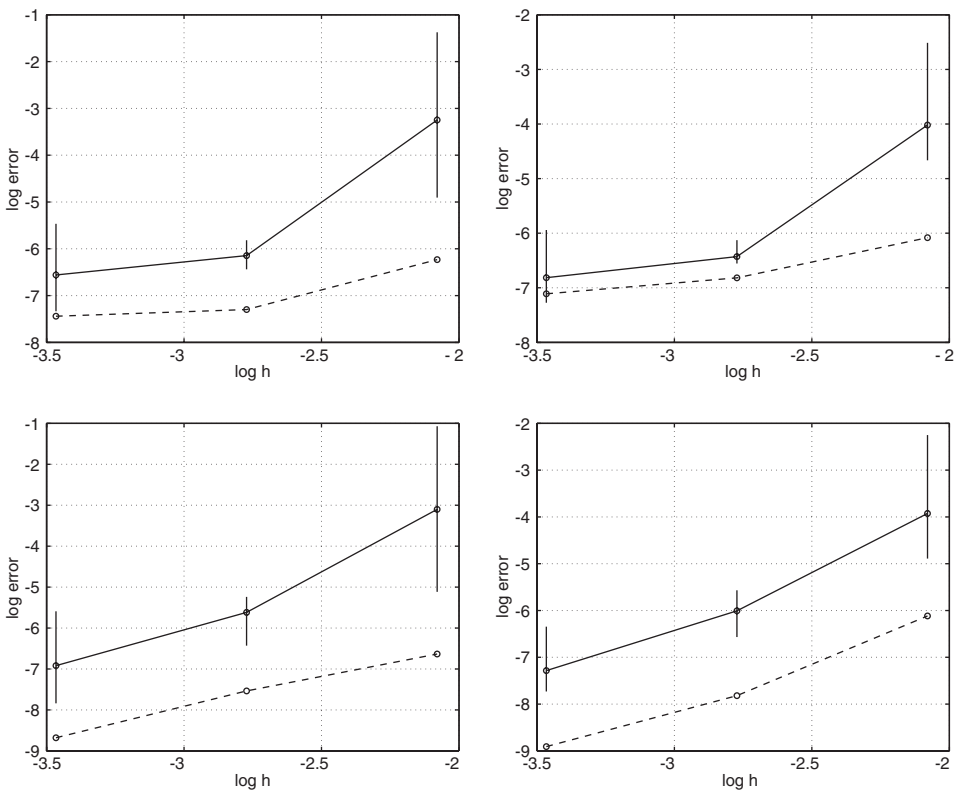

Figure 4. Model 2. Convergence of LOPI method for random grids (continuous line) and uniform grids (dashed line). Top plots: 1-reproduction case; bottom plots: 2-reproduction case. Plot on the left: u error; plots on the right: u_x error. Vertical bars show min/max regions for ten different random grids.

The convergence rate in the 1-reproduction case is somehow deteriorated with respect to that obtained in cases with Dirichlet boundary conditions. We may note that results for the 17×17 and 33×33 cases have almost the same accuracy. Errors at the Neumann boundaries seem to be responsible for such behaviour. (However, we should point out that the error value

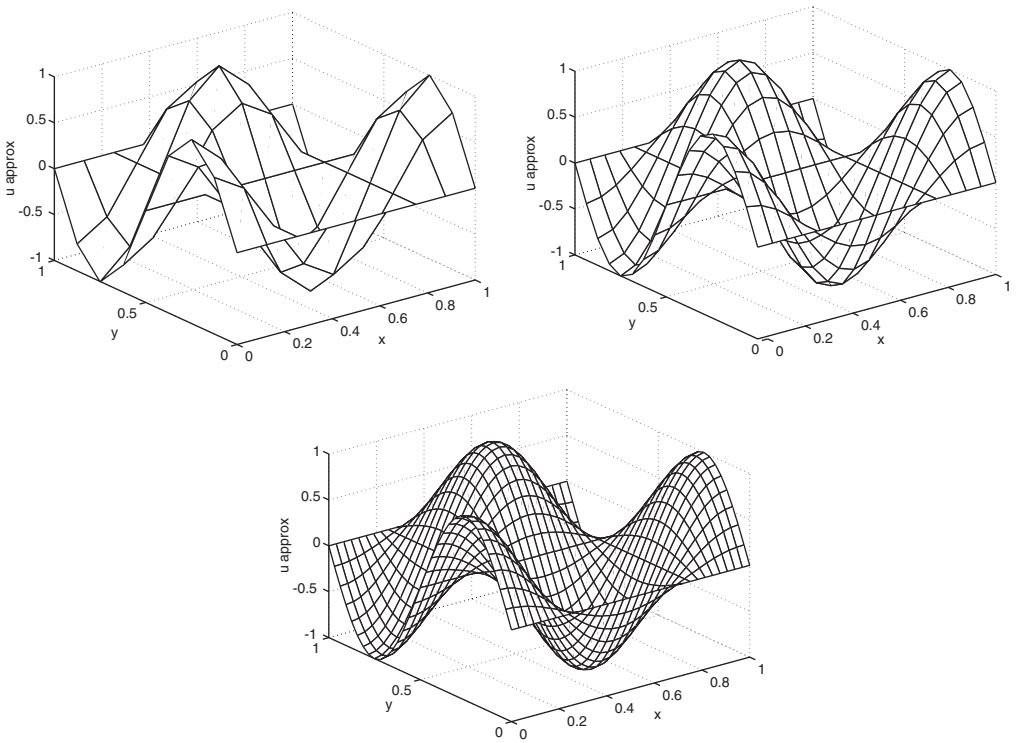

Figure 5. Model 2. Approximate solutions computed with the LOPI scheme with three different levels of points spacing. Points are distributed uniformly on the square.

for the 17×17 points grid is already quite small, and of the same order we got with element free Galerkin methods for this same problem [31].)

When we impose the 2-reproduction property, this anomaly disappears, as seen in Figure 7. In this case, the Neumann boundary condition is correctly represented.

We finally performed analysis of this model problem using the three techniques presented for evaluation of the Neumann boundary condition. Figure 8 displays convergence logarithmic plots for the solution and its x-derivative, when computing the Neumann boundary condition using the *standard* implementation (Equation (14)), the *additional nodes* technique, and the *additional basis function* technique (Equation (17)). Displayed error values are eu_{ℓ^2} and $eu_{x\ell^2}$. We may appreciate that the standard technique gives much larger error values than the two other techniques.

5.3.3. Model 3: Poisson equation with exponential source. In this case, we put a source term that produces a high gradient at the centre of the domain. Homogeneous Dirichlet boundary conditions are imposed.

$$u_{xx} + u_{yy} = f(x, y), \quad \Omega = \{(x, y) | \, 0 < x, y < 1\}$$

$$u|_{\partial\Omega} = 0$$

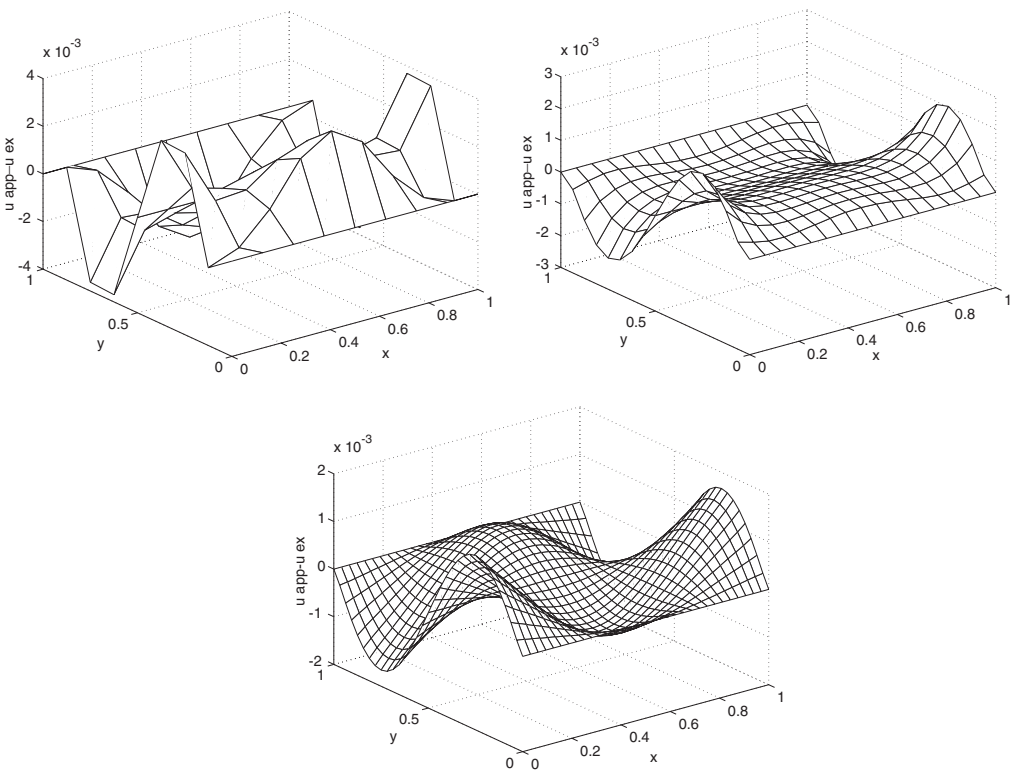

Figure 6. Model 2. Plots of errors computed with the LOPI scheme, 1-reproduction case, with three different levels of points spacing. Points are distributed uniformly on the square. Errors tend to concentrate at the Neumann boundaries as seen.

where

$$f(x,y) = (-2ky(1-y) + (ky(1-x)(1-y) - kxy(1-y))^2 - 2kx(1-x) + (kx(1-y)(1-x) - kxy(1-x))^2)\frac{e^{kxy(1-x)(1-y)}}{1-e^{k/16}}$$

and $k = 200$ in the tested example. The exact solution to this equation is written

$$u(x,y) = \frac{e^{kxy(1-x)(1-y)}}{1-e^{k/16}}$$

Errors for computations with random and uniform points distributions ($9, \times 9$, 17×17 and 33×33) are summarized in Tables VI and VII, respectively. In both cases, computations with 1-reproducing interpolants (R1) and 2-reproducing interpolants (R2) were performed. We note that, in this example, using 2-reproducing interpolants did not improve the accuracy of computations.

Convergence logarithmic plots for the solution and its x-derivative are shown in Figure 9, for both 1- and 2-reproducing interpolants. With the 1-reproducing interpolant, the convergence rate for the solution u is 3.44 with uniformly spaced points, and 3.71 with randomly spaced

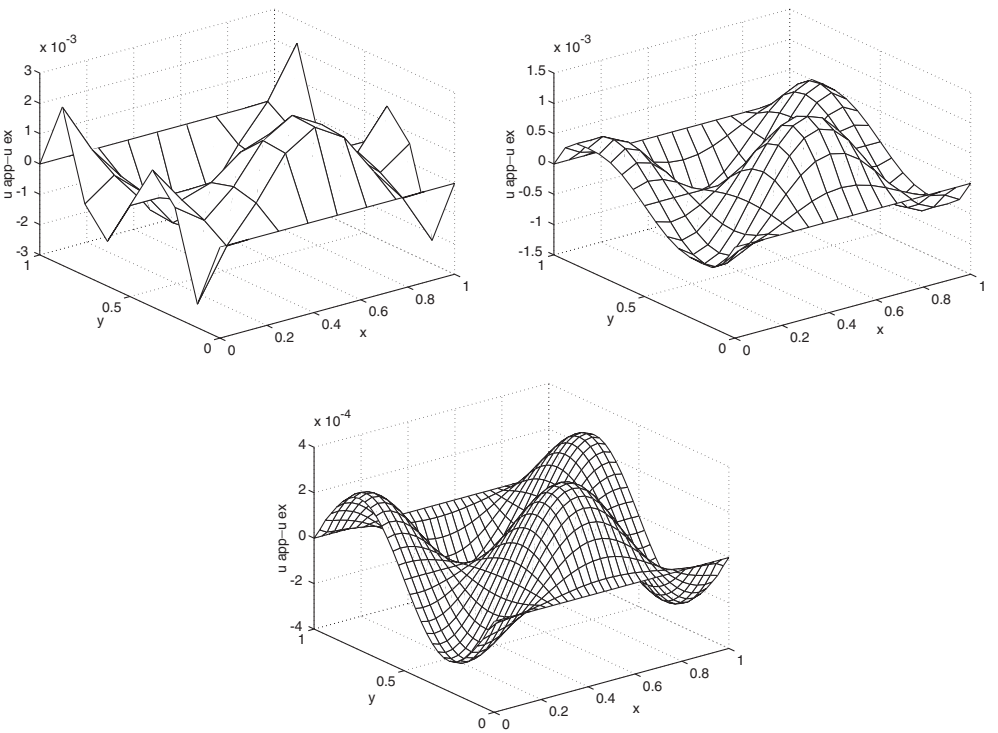

Figure 7. Model 2. Plots of errors computed with the LOPI scheme, 2-reproduction case, with three different levels of points spacing. Points are distributed uniformly on the square. Contrary to the 1-reproduction case, errors are uniformly distributed in the square.

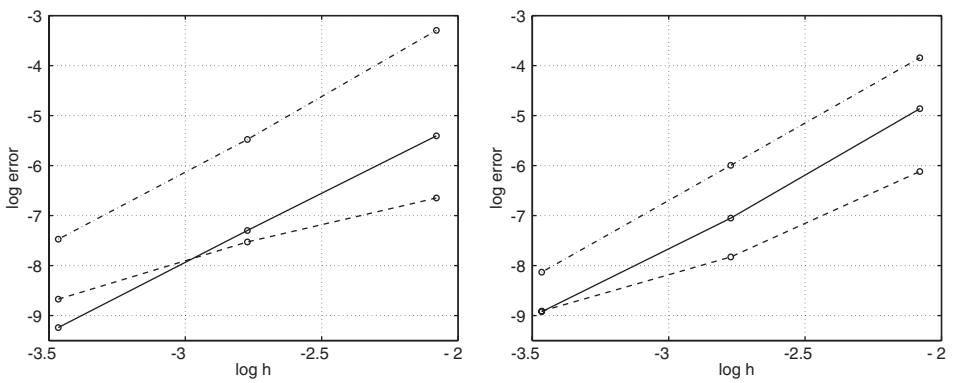

Figure 8. Convergence of LOPI method for different techniques of handling the Neumann boundary condition, 2-reproduction case. Continuous line: additional basis function technique; dashed line: additional nodes technique; dash and points: standard technique. Plot on the left: u error; plots on the right: u_x error.

Table VI. Model 2. Random grid.

	h	eu_∞	$eu_{x\infty}$	$eu_{y\infty}$	eu_{ℓ^2}	$eu_{x\ell^2}$	$eu_{y\ell^2}$
	1/8	1.845e − 1	2.512e − 1	2.221e − 1	5.544e − 2	6.474e − 2	6.238e − 2
R1	1/16	2.754e − 2	5.883e − 2	5.067e − 2	6.949e − 3	9.532e − 3	9.165e − 3
	1/32	2.079e − 3	5.2e − 3	6.090e − 3	5.312e − 4	8.240e − 4	8.344e − 4
	1/8	1.844e − 1	2.804e − 1	2.433e − 1	6.161e − 2	7.512e − 2	7.003e − 2
R2	1/16	2.220e − 2	5.804e − 2	6.069e − 2	5.936e − 3	8.230e − 3	8.950e − 3
	1/32	3.432e − 3	1.030e − 2	1.123e − 2	7.375e − 4	1.228e − 3	1.241e − 3

Note: Mean values of errors over 10 runs. Errors from computations with 1-reproducing interpolants (R1) and 2-reproducing interpolants (R2) are displayed.

Table VII. Model 2. Uniform grid.

	h	eu_∞	$eu_{x\infty}$	$eu_{y\infty}$	eu_{ℓ^2}	$eu_{x\ell^2}$	$eu_{y\ell^2}$
	1/8	9.872e − 2	9.091e − 2	9.091e − 2	1.794e − 2	3.148e − 2	3.148e − 2
R1	1/16	2.383e − 3	2.206e − 2	2.206e − 2	5.084e − 4	4.063e − 3	4.063e − 3
	1/32	4.082e − 4	1.343e − 3	1.343e − 3	1.508e − 4	3.260e − 4	3.260e − 4
	1/8	9.698e − 2	9.103e − 2	9.103e − 2	1.740e − 2	3.133e − 2	3.133e − 2
R2	1/16	1.457e − 3	2.272e − 2	2.272e − 2	5.283e − 4	4.120e − 3	4.120e − 3
	1/32	1.209e − 3	6.389e − 4	6.389e − 4	1.855e − 4	1.923e − 4	1.923e − 4

Note: Computed errors. Errors from computations with 1-reproducing interpolants (R1) and 2-reproducing interpolants (R2) are displayed.

ones. Solution derivatives converge at a rate equal to 3.64 and 3.53, in the uniform and random spacing, respectively. Similar convergence rates are obtained with 2-reproducing interpolants in this test case. Figures 10 and 11 display the computed approximate solutions and the error distribution, respectively, both of them for the uniform grids case and 1-reproducing interpolants.

Del Pin *et al.* [30] used this example to assess the performance of various meshless methods. Figure 12 compares convergence curves for the various methods implemented by Del Pin *et al.* together with the LOPI method, 1-reproducing case. In particular, results from linear triangular finite elements, element free Galerkin (with the implementation given in Reference [31]), fixed least squares [12], smooth particle hydrodynamics [32], natural element method [33] and meshless finite element method (a new method proposed in Reference [30]) are displayed. We may observe the better convergence rate exhibited by the LOPI method both in the evaluation of the function itself as well as in the evaluation of its derivatives.

5.3.4. Model 4: Poisson equation with mixed boundary conditions and high local gradient. This test example is a Poisson equation with an exponential source on a unit square domain. Mixed boundary conditions are imposed: on two sides of the square we impose non-homogeneous Dirichlet boundary conditions while on the other sides we impose non-homogeneous Neumann boundary conditions.

The governing equation and imposed boundary conditions may be written:

$$u_{xx} + u_{yy} = f(x, y), \quad \Omega = \{(x, y) | \, 0 < x, y < 1\}$$

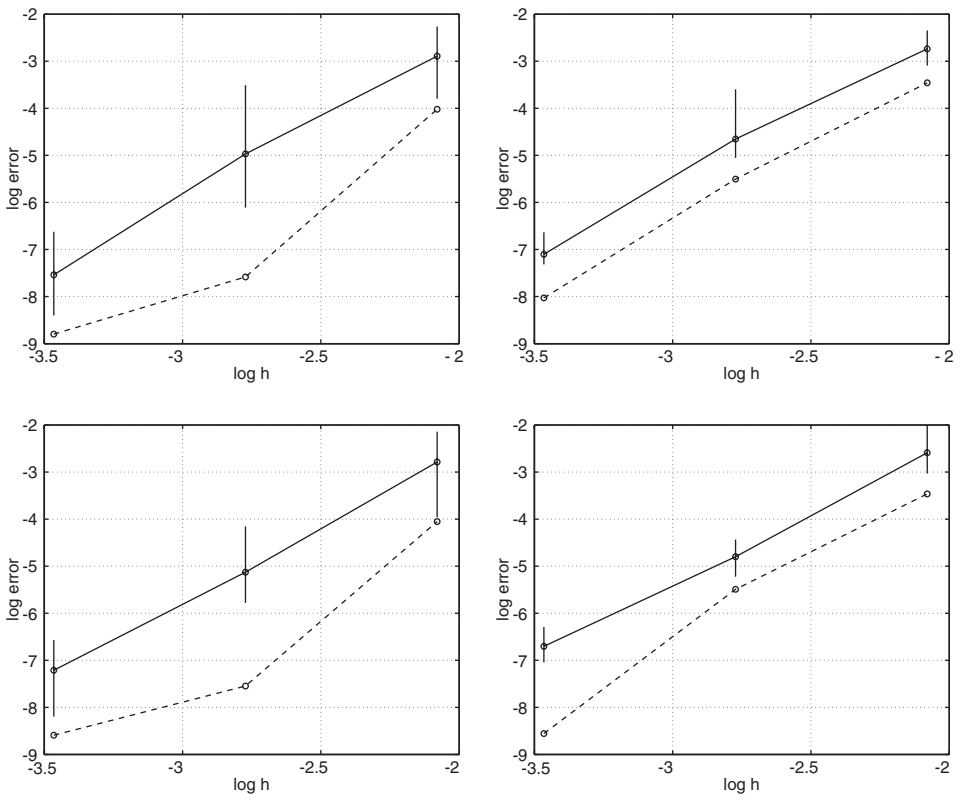

Figure 9. Model 3. Convergence of LOPI method for random grids (continuous line) and uniform grids (dashed line). Top plots: 1-reproduction case; bottom plots: 2-reproduction case. Plot on the left: u error; plots on the right: u_x error. Vertical bars show min/max regions for ten different random grids.

where

$$f(x,y) = -6x - 6y - \left[\frac{4}{\alpha^2} - 4\left(\frac{x-\beta}{\alpha^2}\right)^2 - 4\left(\frac{y-\beta}{\alpha^2}\right)^2\right] \exp\left[-\left(\frac{x-\beta}{\alpha}\right)^2 - \left(\frac{y-\beta}{\alpha}\right)^2\right]$$

$$u(x=0) = -y^3 + \exp\left[-\left(\frac{\beta}{\alpha}\right)^2 - \left(\frac{y-\beta}{\alpha}\right)^2\right]$$

$$u(x=1) = -1 - y^3 + \exp\left[-\left(\frac{1-\beta}{\alpha}\right)^2 - \left(\frac{y-\beta}{\alpha}\right)^2\right]$$

$$u_y(y=0) = \frac{2\beta}{\alpha^2} \exp\left[-\left(\frac{\beta}{\alpha}\right)^2 - \left(\frac{x-\beta}{\alpha}\right)^2\right]$$

$$u_y(y=1) = -3 - 2\frac{1-\beta}{\alpha^2} \exp\left[-\left(\frac{x-\beta}{\alpha}\right)^2 - \left(\frac{1-\beta}{\alpha}\right)^2\right]$$

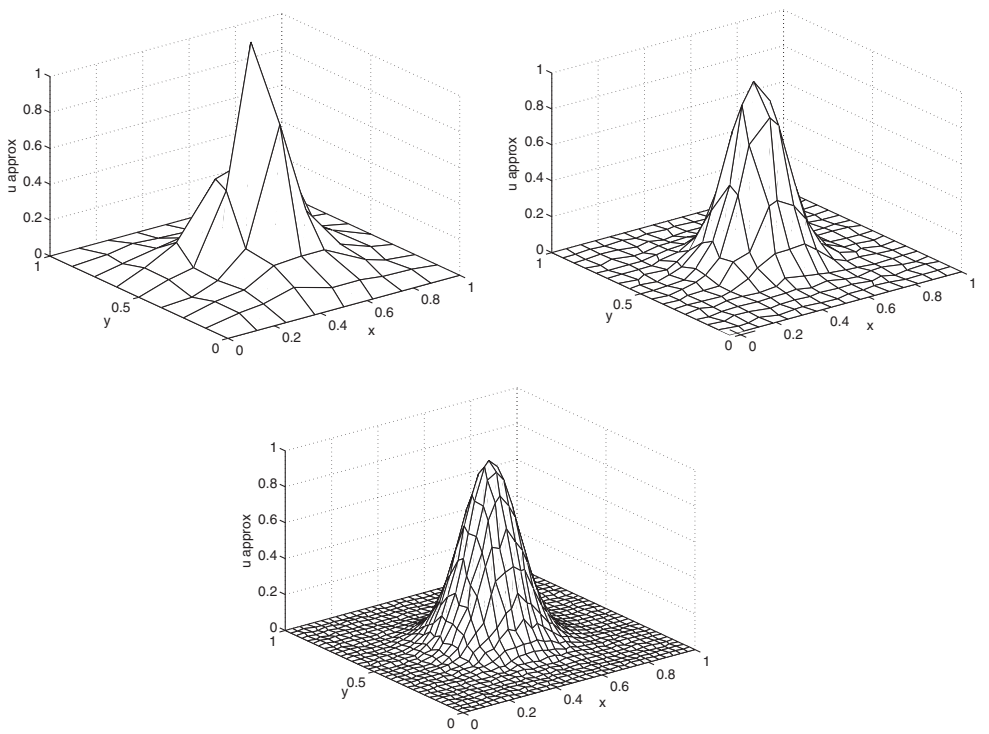

Figure 10. Model 3. Approximate solutions computed with the LOPI scheme (1-reproduction case) with three different levels of points spacing. Points are distributed randomly on the square.

and $\beta = 0.5$; $\alpha = 0.05$ in the tested example. The exact solution to this equation is written

$$u(x, y) = -x^3 - y^3 + \exp\left[-\left(\frac{x-\beta}{\alpha}\right)^2 - \left(\frac{y-\beta}{\alpha}\right)^2\right]$$

This example was proposed by Aluru [14] to test performance of the methods in presence of both high local gradients and mixed non-homogeneous boundary conditions.

Figure 13 displays two computed solutions for grids with uniform point spacing of $h = 1/32$ and $1/64$. We may see the local high gradients produced at the centre of the square. Figure 14 gives the distribution of computation errors on the domain. We may see that for $h = 1/32$, the maximum errors are located at the high gradients zone. When the points spacing is reduced to $h = 1/64$, errors induced by the Neumann boundary condition become noteable.

Finally, Figure 15 shows the convergence curve computed from LOPI scheme compared to results obtained by Aluru for this same problem. We may appreciate that the LOPI scheme displays a slightly better convergence rate than Aluru's in this case.

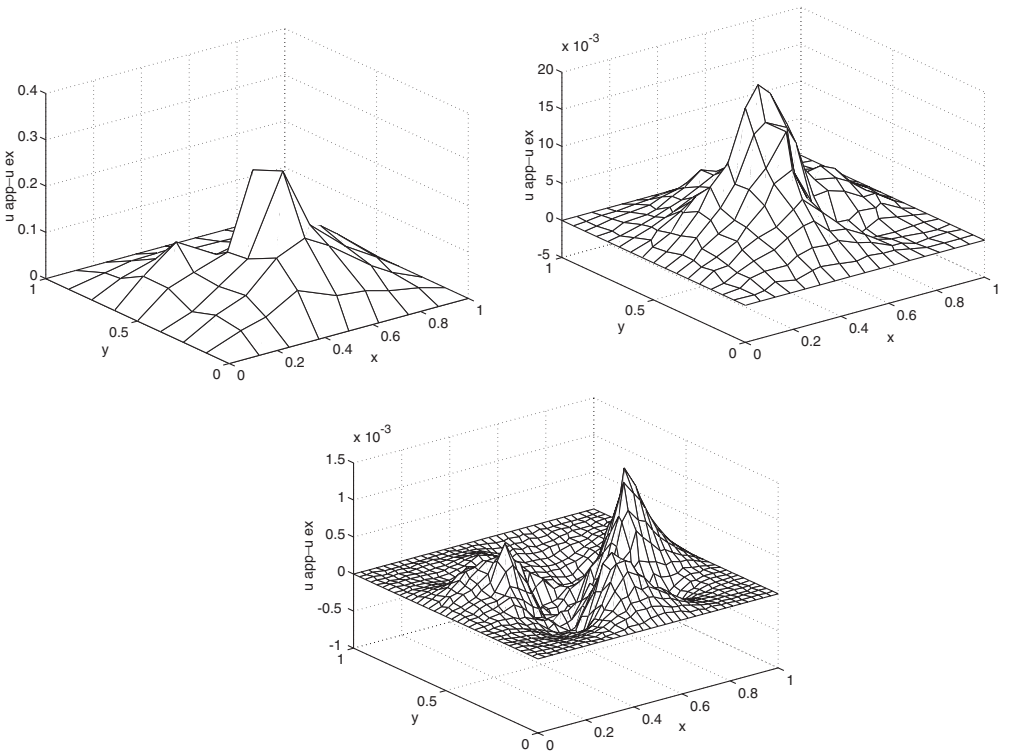

Figure 11. Model 3. Plots of errors computed with the LOPI scheme (1-reproduction case) with three different levels points spacing. Points are distributed randomly on the square.

6. CONCLUSIONS

Several computational features of the local optimal point interpolator for solving PDEs by collocation are demonstrated in this paper:

- The numerical experiments performed show that the LOPI scheme is a very efficient method. The method is highly flexible and produces accurate results. We have demonstrated that even a reduced number of nodes is enough to obtain an accurate solution in simple cases.
- The method is truly meshless, without requiring any kind of grid. We have tested it with reasonable random points distributions.
- Unlike many other particle methods, the proposed scheme is able to handle Dirichlet boundary conditions without trouble. The interpolation functions satisfy the delta Kronecker property.
- The examples have shown computations with both 1- and 2-reproducing interpolants. When using collocation with 2-reproducing interpolation, results evidenced an $O(h^3)$ accuracy, in accordance with predictions of the theory. Nevertheless, it should be pointed out that in most cases, the 1-reproducing interpolants gave accurate results.

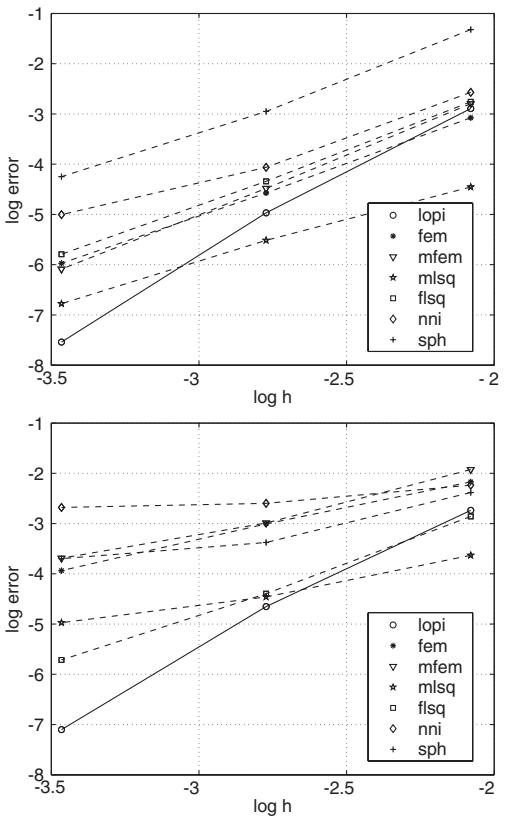

Figure 12. Model 3. Convergence of the LOPI method, 1-reproduction case, for random grids (continuous line) compared to results from other discretization methods. Lopi: this method; fem: linear triangular finite elements; mfem: meshless finite element method [30]; mlsq: element free Galerkin [31]; flsq: fixed least squares [12]; nni: natural element method [33]; sph: smooth particle hydrodynamics [32]. Top figure: convergence curves for function values; bottom figure: convergence curves for derivatives. Curves represent the mean of ten computations on random points distributions, with a $0.25h$ perturbation. Tests were taken from the work of Del Pin *et al.* [30].

- The displayed examples treated only square domains. However, it is clear that the methodology may be applied to general domains without losing its local character, at least when their geometry is not too complicate.

The high flexibility and accuracy of this method also presents some challenging problems. One issue in need of further study is the effect of constant $\tilde{c}_q$ in (8) and constant r_ω in (18) on the convergence rate of the method. Changing these values can sometimes drastically alter the accuracy of the solutions. Kansa [15] introduced the concept of variable shape parameters in the global MQ scheme. Also, a strategy for selecting the shape parameter based upon the local radius of curvature of the solution surface was found to give good results [34]. The approach adopted here appears naturally when studying the convergence rate of the local optimal interpolation formula, as we shall show in a forthcoming paper.

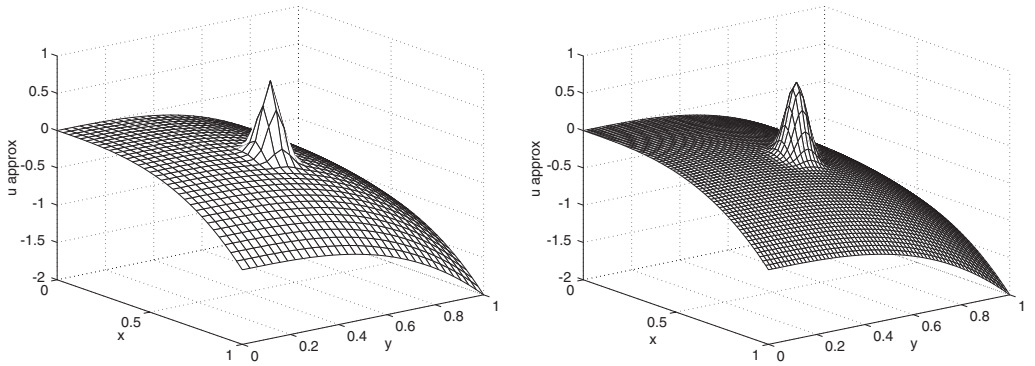

Figure 13. Model 4. Approximate solutions computed with the LOPI scheme, 1-reproduction case, with two different levels of points spacing. Points are distributed uniformly in the square.

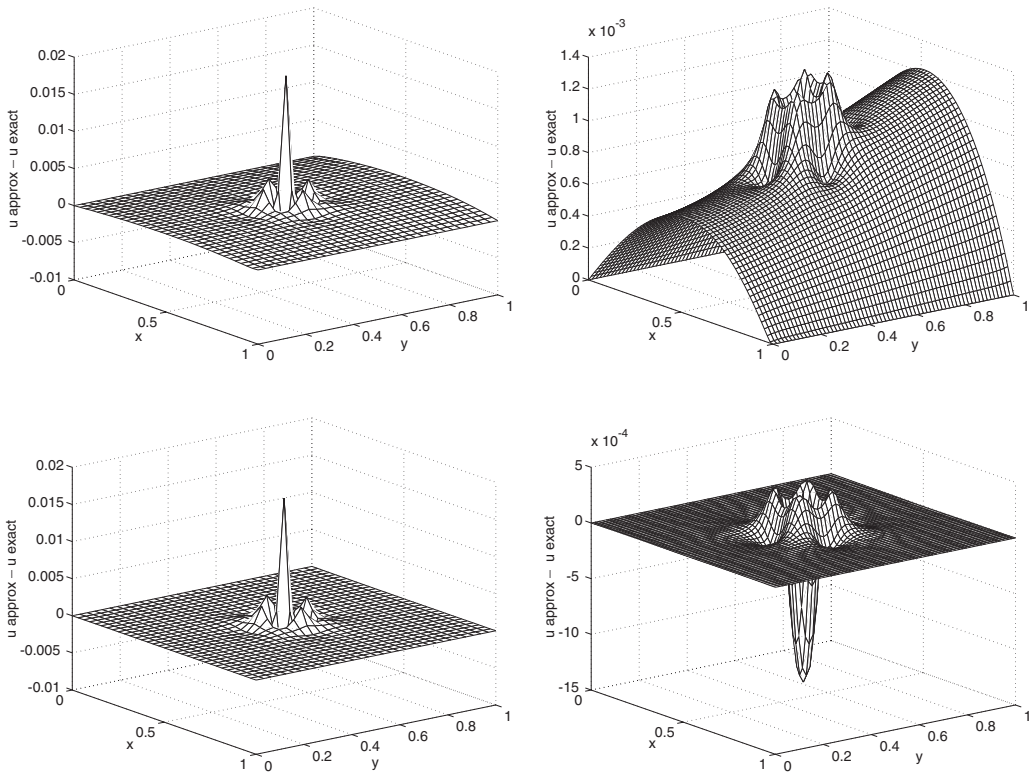

Figure 14. Model 4. Plots of errors computed with the LOPI scheme with two different levels of points spacing. Points are distributed uniformly in the square. Figures on top display the 1-reproduction case, while on bottom display the 2-reproduction case. We note that in the latter case, errors on the Neumann boundaries are smaller than in the former one.

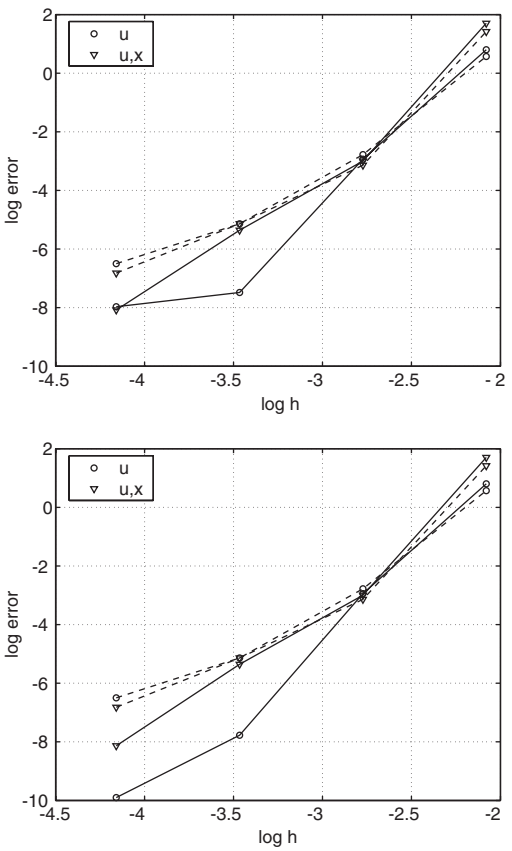

Figure 15. Model 4. Convergence of the LOPI method for uniform grids (continuous line) compared to results from RPKM scheme by Aluru [14]. Top plot: 1-reproduction case; bottom plot: 2-reproduction case.

Much additional work remains to be done. Nevertheless, the results presented in this paper are quite encouraging and show that the LOPI scheme has a great potential to become a very competitive method for the solution of a broad class of boundary-value problems.

REFERENCES

1. Belytchsko T, Krongauz Y, Organ D, Fleming M, Krysl P. Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:3–48.
2. Liu WK, Chen Y, Jun S, Chen JS, Belytchsko T, Uras TA, Chang CT. Overview and applications of the reproducing kernel particle methods. *Archives of Computational Mechanics in Engineering: State of the Reviews* 1996; **3**:3–80.
3. Li S, Liu WK. Meshfree and particle methods and their applications. *Applied Mechanics Reviews* (ASME) 2002; **55**:1–34.
4. Nayroles B, Touzot G, Villon P. Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational Mechanics* 1992; **10**:307–318.
5. Belytchsko T, Lu YY, Gu L. Element free Galerkin methods. *International Journal for Numerical Methods in Engineering* 1994; **37**:229–256.
6. Liu WK, Jun S, Zhang S. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids* 1995; **20**:1081–1106.

7. Duarte CA, Oden JT. An hp adaptive method using clouds. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:237–262.
8. De S, Bathe KJ. The method of finite spheres. *Computational Mechanics* 2000; **25**:329–345.
9. Babuška I, Melenk JM. The partition of unity method. *International Journal for Numerical Methods in Engineering* 1997; **40**:727–758.
10. Monaghan JJ. Why particle methods work (Hydrodynamics). *SIAM Journal on Scientific and Statistical Computing* 1982; **3**:422–433.
11. Oñate E, Idelsohn S, Zienkiewicz OC, Taylor RL. A finite point method in computational mechanics. Applications to convective transport and fluid flow. *International Journal for Numerical Methods in Engineering* 1996; **39**:3839–3866.
12. Oñate E, Sacco C, Idelsohn S. A finite point method for incompressible flow problems. *Computing and Visualization in Science* 2000; **3**:67–75.
13. Aluru NR. A point collocation method based on reproducing kernel approximations. *International Journal for Numerical Methods in Engineering* 2000; **47**:1083–1121.
14. Aluru NR, Li G. Finite cloud method: a true meshless technique based on a fixed reproducing kernel approximation. *International Journal for Numerical Methods in Engineering* 2001; **50**:2373–2410.
15. Kansa EJ. Multiquadratics—a scattered data approximation scheme with applications to computational fluid dynamics—I. surface approximations and partial derivative estimates. *Computers and Mathematics with Applications* 1990; **19**:127–145.
16. Kansa EJ. Multiquadratics—a scattered data approximation scheme with applications to computational fluid dynamics—II. solutions to hyperbolic, parabolic, and elliptic partial differential equations. *Computers and Mathematics with Applications* 1990; **19**:147–161.
17. Hardy RL. Multiquadratic equations of topography and other irregular surfaces. *Journal of Geophysical Research* 1971; **76**:1905–1915.
18. Hardy RL. Theory and applications of the multiquadratic-biharmonic methods: 20 years of discovery. *Computers and Mathematics with Applications* 1990; **19**:163–208.
19. Franke R. Scattered data interpolation: tests on some methods. *Mathematics of Computation* 1982; **38**:181–199.
20. Sharan M, Kansa EJ, Gupta S. Application of the multiquadratic method for numerical solution of elliptic partial differential equations. *Applied Mathematics and Computation* 1997; **84**:275–302.
21. Golberg M, Chen CS. Improved multiquadratic approximation for partial differential equations. *Engineering Analysis with Boundary Elements* 1996; **18**:9–17.
22. Hagen RE, Kansa EJ. Studies on the *r* parameter in the multiquadratic functions applied to ground water pumping. *Journal of Applied Sciences and Computation* 1994; **1**:266–281.
23. Foley TA. The map and blend scattered data interpolation on a sphere. *Computers and Mathematics with Applications* 1992; **24**:41–60.
24. Duval MR. Domain decomposition and local refinement for multiquadratic approximations. I: Second-order equations in one dimension. *Journal of Applied Sciences and Computing* 1994; **1**:146–171.
25. Zuppa C. A local optimal point interpolation formula. *Revista de Matemática Aplicada*, *Universidad de Chile* 2002; **23**:7–28.
26. Lancaster P, Salkauskas K. *Curve and Surface Fitting*: *An Introduction*. Academic Press, San Diego, 1986.
27. Zuppa C. Good quality point sets for moving least squares and error estimates, *Technical Report*, Depto. de Matemáticas, Universidad de San Luis, 2002.
28. Micchelli C. Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constructive Approximation* 1986; **2**:11–22.
29. Zuppa C. Error estimates for moving least square approximations. *Technical Report*. Depto de Matemáticas, Universidad de San Luis, 2001.
30. Del Pin F, Idelsohn S, Calvo N, Storti M. Comparación del método de elementos finitos con métodos meshless en nubes de puntos random. In *Mecánica Computacional—Actas ENIEF*, Flores F, *et al*. (eds). Córdoba, Argentina, 2001, Asociación Argentina de Mecánica Computacional.
31. Simonetti G, Cardona A. Métodos sin malla para resolver la ecuación de conducción del calor. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería* 2000; **16**:33–47
32. Koshizuka S, Oka Y. Moving particle semi-implicit method for fragmentation of incompressible fluid. *Nuclear Science and Engineering* 1996; **123**:421–434.
33. Sukumar N. *The natural element method in solid mechanics. Ph.D. Thesis*, North-Western University, 1998.
34. Kansa EJ, Hon YC. Circumventing the ill-conditioning problem with multi-quadratics radial basis functions: Applications to elliptic partial differential equations. *Computers and Mathematics with Applications* 2000; **39**:123–137.

