Mecánica Computacional Vol. XXI, pp. 1444–1460
S. R. Idelsohn, V. E. Sonzogni and A. Cardona (Eds.)
Santa Fe–Paraná, Argentina, October 2002

APPLICATION OF LOPI METHOD TO SOLVE PDES

Jose Risso*, Alberto Cardona* and Carlos Zuppa†

*Centro Internacional de Métodos Computacionales en Ingeniería
(CIMEC - INTEC). CONICET - Universidad Nacional del Litoral
Güemes 3450, (3000) Santa Fe, Argentina.
e-mail: jrisso@intec.unl.edu.ar

†Departamento de Matemática.
Universidad Nacional de San Luis.
Chacabuco y Pedernera, (5700) San Luis, Argentina.

Key Words: Meshless, Interpolation, Polynomial reproduction, Point collocation, Boundary conditions.

Abstract.

The Local Optimal Point Interpolation (LOPI) method is a "truly meshless" method based on the boolean sum of a radial basis function interpolator and a least squares approximation in a polynomials space. In this way, it can interpolate solutions in data points, while at the same time fit exactly polynomial solutions up to certain degree. Systems of PDEs can be solved in strong form using point collocation, without meshes or integration cells. Essential boundary conditions (which cause problems in many other meshless methods) are applied directly; and natural boundary conditions are implemented by means of additional equations. This work presents a description of the method, and some examples of applications.

1 INTRODUCTION

Meshless methods are the subject of great attention in recent years, and many schemes were developed and tested. We refer for instance to reviews done by Belytschko et al.,[1] Li and Liu,[4] Liu et al.,[5] and Del Pin et al.[6] Applications of these methods are found in large deformation problems, strong discontinuities analysis, Lagrangian formulations in fluid dynamics, and many other areas.

From a computational point of view, *truly meshless* methods solving PDEs in strong form have many advantages compared with meshless methods using integration cells (a background mesh) to solve PDEs in weak form.

The approach followed in this work was introduced by Zuppa and Cardona[9] and is based on the use of Local Optimal Point Interpolators with compact support radial basis functions.

2 DESCRIPTION OF LOPI METHOD

The following is a brief description of the method. Mathematical details, and considerations about approximation properties were analyzed by Zuppa and Cardona.[9]

2.1 Data approximation

In LOPI method, data is approximated by means of a lineal combination of values φ in scattered points, multiplied by an interpolation function, so:

$$f_{(\mathbf{x})} = \sum_{i=1}^{K} \mathcal{T}_{\mathcal{X}_i(\mathbf{x})} \cdot \varphi_i \tag{1}$$

The set $\mathcal{X}$, defined by points in the neighborhood (*cloud*) of the evaluation point, characterizes this scheme as compactly supported, lowering its computational cost.

The interpolation function is defined as:

$$\mathcal{T}_{\mathcal{X}(\mathbf{x})} \;=\; \mathbf{v}_{(\mathbf{x})} \cdot P_{\mathcal{X}} \;+\; \mathbf{b}_{(\mathbf{x})} \cdot Q_{\mathcal{X}} \tag{2}$$

where the second internal product is the Least Squares approximation to the data, and the first one is the correction used to make the approximation interpolant.

Matrix $Q_{\mathcal{X}}$ is the standard Weighted Least Squares normal matrix, defined as:

$$Q_{\mathcal{X}} = (B_{\mathcal{X}}^t \cdot W \cdot B_{\mathcal{X}})^{-1} \cdot B_{\mathcal{X}}^t \cdot W \tag{3}$$

where $B_{\mathcal{X}}$ is the matrix constructed from a polynomial basis for the desired reproduction order r (x_c are the coordinates of evaluation point, $x_{(i)}$ are the coordinates of scattered points in the set $\mathcal{X}$) :

$$B_{\mathcal{X}} = \begin{pmatrix} 1 & \mathbf{x}_1 - \mathbf{x}_c & \dots & (\mathbf{x}_1 - \mathbf{x}_c)^r \\ \vdots & \vdots & & \vdots \\ 1 & \mathbf{x}_K - \mathbf{x}_c & \dots & (\mathbf{x}_K - \mathbf{x}_c)^r \end{pmatrix}. \tag{4}$$

and W is the weight matrix, usually a diagonal matrix based on a decreasing radial basis function (or even the Identity matrix if standard least squares is used as in ref.[9]).

Vector $\mathbf{b}_{(x)}$ is the polynomial basis for the desired approximation order:

$$\mathbf{b}_{(\mathbf{x})} = (1, \mathbf{x}_c, \ldots, \mathbf{x}_c^r) \tag{5}$$

Matrix $P_{\mathcal{X}}$ is constructed as:

$$P_{\mathcal{X}} = V^{-1}(I \ - \ B_{\mathcal{X}}.Q_{\mathcal{X}}) \tag{6}$$

where V is the *Vandermondian* (or *Grammian*) of the radial basis function (RBF) g :

$$V = \begin{bmatrix} g_{(\mathbf{x}_1-\mathbf{x}_1)} & \cdots & g_{(\mathbf{x}_K-\mathbf{x}_1)} \\ \vdots & \ddots & \vdots \\ g_{(\mathbf{x}_1-\mathbf{x}_K)} & \cdots & g_{(\mathbf{x}_K-\mathbf{x}_K)} \end{bmatrix} \tag{7}$$

and it is assumed to be invertible, g is a radial basis function, for example the multiquadric radial C^∞ function :

$$g_{(\mathbf{x})} = \sqrt{1 + c_q \left\|\mathbf{x}\right\|^2}, \quad \mathbf{x} \in \mathbb{R}^n \tag{8}$$

Vector $\mathbf{v}_{(\mathbf{x})}$ is defined as:

$$\mathbf{v}_{(\mathbf{x})} = (g_{(\mathbf{x}_c-\mathbf{x}_1)}, \cdots g_{(\mathbf{x}_c-\mathbf{x}_K)}) \tag{9}$$

other radial basis functions (like Gauss exponentials) can be used to define $\mathbf{v}$ and V.

2.2 Calculation of derivatives

Since derivatives are calculated considering matrices $P_{\mathcal{X}}$ and $Q_{\mathcal{X}}$ fixed in an evaluation point, only vectors $\mathbf{v}$ and $\mathbf{b}$ should be differentiated:

$$\frac{\partial \mathcal{T}_{\mathcal{X}(\mathbf{x})}}{\partial \mathbf{x}} \ = \ \frac{\partial \mathbf{v}_{(\mathbf{x})}}{\partial \mathbf{x}} \cdot P_{\mathcal{X}} \ + \ \frac{\partial \mathbf{b}_{(\mathbf{x})}}{\partial \mathbf{x}} \cdot Q_{\mathcal{X}} \tag{10}$$

higher order derivatives are calculated following the same scheme.

Alternative schemes, in which matrices $P_{\mathcal{X}}$ and $Q_{\mathcal{X}}$ are also differentiated were used without significative improvements in the results, and were discarded because their computational cost is considerably higher.

2.3 Solution of PDE boundary value problems

LOPI is a point collocation method, therefore PDEs are solved in strong form and there is no need to use integration cells or backgroud meshes as in other meshless methods. As an example, a solution will be developed for a well-posed elliptic boundary value problem:

$$\begin{aligned} &Pu(\mathbf{x}) = f(\mathbf{x}), && \mathbf{x} \in \Omega \\ &B_N u(\mathbf{x})|_{\Gamma_N} = s_N(\mathbf{x}) \\ &B_D u(\mathbf{x})|_{\Gamma_D} = s_D(\mathbf{x}) \end{aligned} \tag{11}$$

Here, Ω is a bounded domain in $\mathbb{R}^n$, B_D a Dirichlet operator and B_N a Neumann or mixed operator.

Let $\mathcal{X}^N$ denote an arbitrarily chosen set of N points $\mathbf{x}_\alpha \in \overline{\Omega}$ (nodes):

$$\mathcal{X}^N = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}, \quad \mathbf{x}_\alpha \in \overline{\Omega}$$

Let $\mathcal{I}_N := \{\omega_\alpha\}_{\alpha=1}^N$ denote a finite open covering of $\overline{\Omega}$ consisting of N clouds ω_α such that $\mathbf{x}_\alpha \in \omega_\alpha$, $\alpha = 1, \ldots, N$ constitutes the center of the cloud, and

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

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

$$\mathcal{X}^N = \{(\mathbf{x}_\beta)|_{\beta=1,\ldots M_1} \subset \Omega, (\mathbf{x}_\beta)|_{\beta=M_1+1,\ldots M_2} \subset \Gamma_N, (\mathbf{x}_\beta)|_{\beta=M_2+1,\ldots N} \subset \Gamma_D\}$$

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

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

$$\mathcal{T}_\alpha := \widetilde{\mathcal{T}}_{\mathcal{S}(\alpha)} \circ p_{\mathcal{S}(\alpha)}.$$

At each cloud the solution will be approximated as

$$u_h(\mathbf{x}) = \mathcal{T}_\alpha[\mathbf{u}](\mathbf{x}) = \sum_{i=1}^{M_2} u_i \cdot \varphi_i^\alpha(\mathbf{x}) + S_D^\alpha(\mathbf{x}), \qquad \mathbf{x} \in \omega_\alpha, \quad \alpha = 1, \ldots N \tag{12}$$

where $S_D^\alpha(\mathbf{x}) = \sum_{i=M_2+1}^{N} s_D(\mathbf{x}_i) \cdot \varphi_i^\alpha(\mathbf{x})$.

Substituting $u_h(\mathbf{x})$ into the PDE and using collocation at the nodes, we have the linear system

$$\begin{aligned} \sum_{i=1}^{M_2} u_i \cdot P\varphi_i^\alpha(\mathbf{x}_\alpha) &= f(\mathbf{x}_\alpha) - PS_D^\alpha(\mathbf{x}_\alpha), & \alpha = 1, \ldots M_1 \\ \sum_{i=1}^{M_2} u_i \cdot B_N\varphi_i^\alpha(\mathbf{x}_\alpha) &= s_N(\mathbf{x}_\alpha) - B_N S_D^\alpha(\mathbf{x}_\alpha), & \alpha = M_1+1, \ldots M_2 \end{aligned} \tag{13}$$

2.4 Boundary conditions

Essential boundary conditions are applied straightforwardly. This is a considerable advantage compared with other meshless methods based on the use of radial basis functions.

Zuppa and Cardona[9] determined the convenience of increasing the number of basis functions at every Neumann node to impose this type of boundary conditions.

At every node $\mathbf{x}_\gamma \in S_\alpha \cap \Gamma_N$, a basis function $\psi^\alpha_{\gamma(\mathbf{x})}$ is defined using:

$$\psi^\alpha_{\gamma(\mathbf{x})} = < \mathbf{n}_{(\mathbf{x}_\gamma)}, (\mathbf{x} - \mathbf{x}_\gamma) > \varphi^\alpha_{\gamma(\mathbf{x})}$$

where $\mathbf{n}(\mathbf{x}_\gamma)$ is the normal vector in $\mathbf{x}_\gamma \in \Gamma_N$, $\psi^\alpha_{\gamma(\mathbf{x}_\beta)} = 0$ for all $\mathbf{x}_\beta \in \mathcal{X}^N$ and $\frac{\partial}{\partial n}\psi^\alpha_\gamma(\mathbf{x}_\gamma) = \mathbf{n} \cdot \nabla\psi^\alpha_\gamma(\mathbf{x}_\gamma) = 1$.

If $S_\alpha \cap \Gamma_N \neq \emptyset$, we may improve the approximation by

$$u_h(\mathbf{x}) = \sum_{i=1}^{M_2} u_i \cdot \varphi^\alpha_i(\mathbf{x}) + \sum_{i=M_1+1}^{M_2} g_i \cdot \psi^\alpha_i(\mathbf{x}) + S^\alpha_D(\mathbf{x}), \qquad \mathbf{x} \in \omega_\alpha, \quad \alpha = 1, \dots N \tag{14}$$

where $S^\alpha_D(\mathbf{x}) = \sum_{i=M_2+1}^{N} s_D(\mathbf{x}_i) \cdot \varphi^\alpha_i(\mathbf{x})$. Again, two equations are imposed at every node $\mathbf{x}_\alpha \in \Gamma_N$.

$$\begin{aligned} \sum_{i=1}^{M_2} u_i \cdot P\varphi^\alpha_i(\mathbf{x}_\alpha) + \sum_{i=M_1+1}^{M_2} g_i \cdot P\psi^\alpha_i(\mathbf{x}_\alpha) &= f(\mathbf{x}_\alpha) - PS^\alpha_D(\mathbf{x}_\alpha) \\ & \qquad \alpha = 1, \dots M_2 \quad (15) \\ \sum_{i=1}^{M_2} u_i \cdot B_N\varphi^\alpha_i(\mathbf{x}_\alpha) + \sum_{i=M_1+1}^{M_2} g_i \cdot B_N\psi^\alpha_i(\mathbf{x}_\alpha) &= s_N(\mathbf{x}_\alpha) - B_N S^\alpha_D(\mathbf{x}_\alpha), \\ & \qquad \alpha = M_1 + 1, \dots M_2 \end{aligned}$$

3 PARAMETERS OF THE METHOD

Several parameters of the method can be tuned in order to improve solutions. We review next the influence of these parameters in the accuracy of the method. Comparisons were made analyzing 3 cases (with analytical solution) used in the past[6] to compare meshless methods.

Errors evaluated were:

$$er_\infty = \frac{1}{\max_{\beta=1,\dots M_2} |u(\mathbf{x}_\beta)|} \max_{\beta=1,\dots M_2} \left|(u - u_h)(\mathbf{x}_\beta)\right| \tag{16}$$

$$er_2 = \frac{1}{\max_{\beta=1,\dots M_2} |u(\mathbf{x}_\beta)|} \sqrt{\frac{1}{M_2} \sum_{\beta=1,\dots M_2} \left|(u - u_h)(\mathbf{x}_\beta)\right|^2} \tag{17}$$

where M_2 is the number of points in the discretization.

Model 1: 2D Laplace equation with Dirichlet boundary condition

$$\begin{aligned} &u_{xx} + u_{yy} = 0, \qquad \Omega = \{(x,y)|\, 0 < x, y < 1\} \\ &u|_{\partial\Omega} = g(x,y) \end{aligned}$$

Model 2: 2D Poisson equation with mixed boundary conditions

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

Model 3: 2D Poisson equation with exponential source

$$\begin{aligned} &u_{xx} + u_{yy} = f(x,y), \qquad \Omega = \{(x,y)|\, 0 < x, y < 1\} \\ &u|_{\partial\Omega} = 0 \end{aligned}$$

where $f_{(x,y)} = \frac{[-2ay(1-y)+(ay(1-x)(1-y)-axy(1-y))^2-2ax(1-x)+(ax(1-y)(1-x)-axy(1-x))^2]e^{axy(1-x)(1-y)}}{1-e^{a/16}}$ and the values $a = 200$ and $a = 1000$ were used.

3.1 Interpolation function

Several Radial basis functions (RBFs) can be used in the method, and all of them have a numerical parameter that can be adjusted, producing variations in the results of the method. We restricted the scope to functions with smooth second order derivatives, because our aim was the analysis of elliptic problems involving second order derivatives of the variable.

Zuppa and Cardona[9] used the multiquadrics function:

$$g_{(\mathbf{x})} = \sqrt{1 + c_q \|\mathbf{x}\|^2} \tag{18}$$

where $c_q = k/r_c^2$, and r_c is the cloud radius (maximum distance from the evaluation point to another point of the cloud).

In this work, we propose the use of a slighty modified multiquadrics:

$$g_{(\mathbf{x})} = \sqrt{c_q + \|\mathbf{x}\|^2} \tag{19}$$

where $c_q = kr_c^2$.

Following Lancaster and Salkauskas[3] we used also the rotated Gaussian:

$$g_{(\mathbf{x})} = e^{c_q \|\mathbf{x}\|^2} \tag{20}$$

where $c_q = k/r_c^2$.

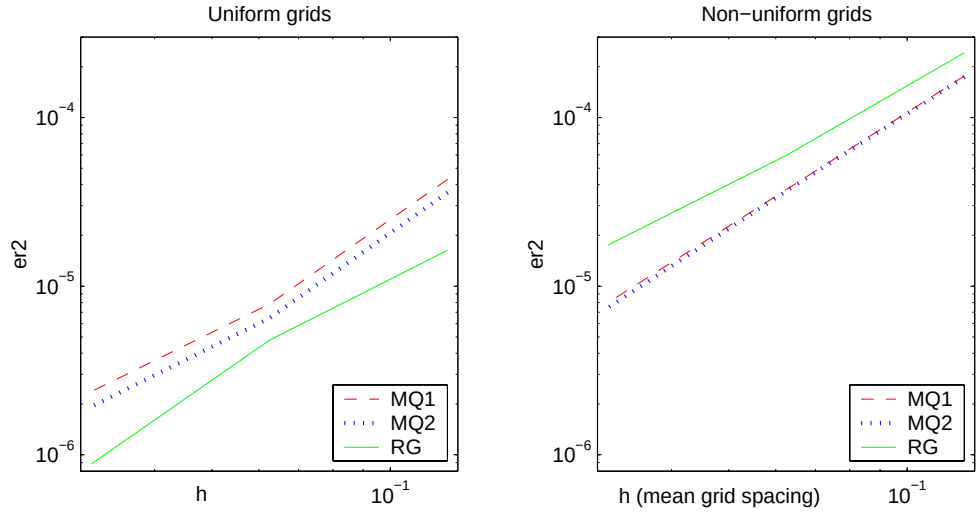

Figure 1: Error comparison - Model 1

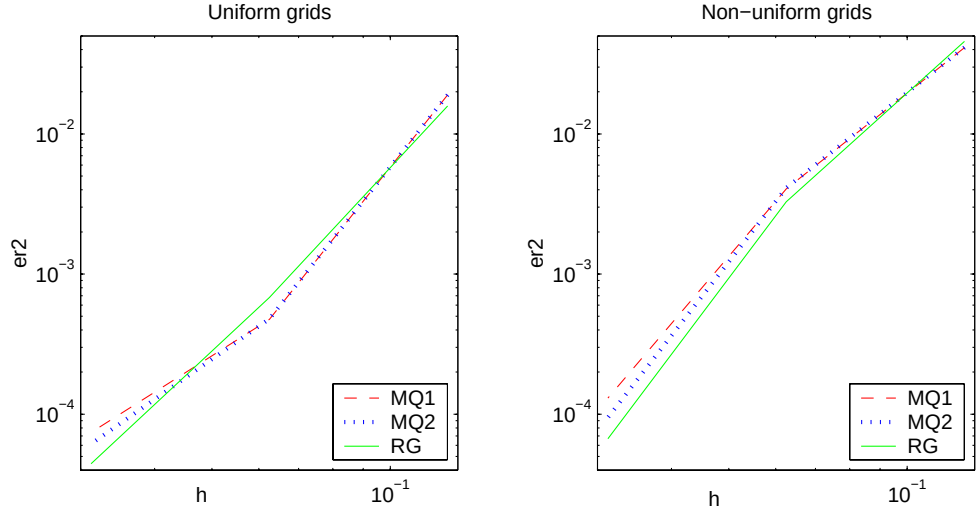

Figure 2: Error comparison - Model 3 a=200

Figures 1 and 2 show a comparison of the results obtained using these functions. There are no appreciable differences using one or another RBF, and only in specific problems some of them show minor advantages. We conclude that the type of RBF used is less important than its dilation parameter.

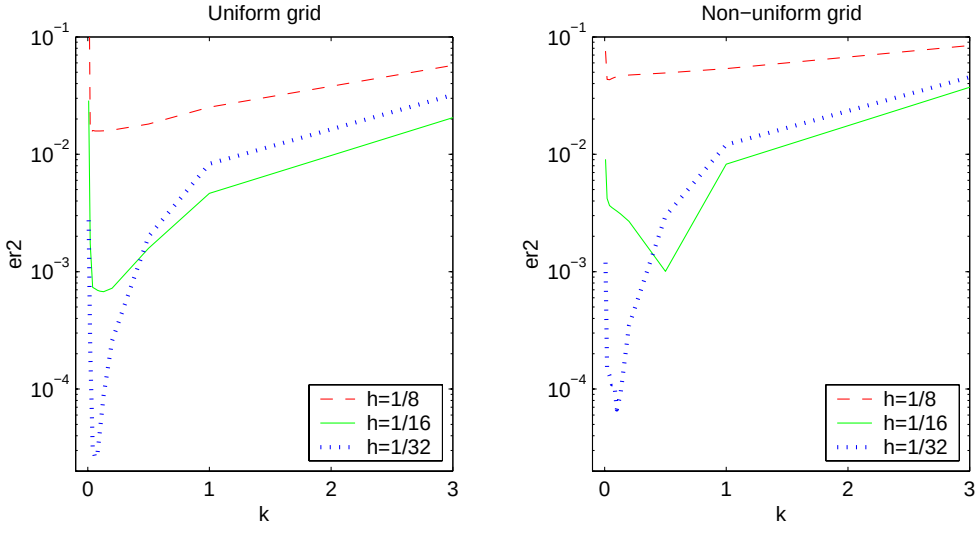

Figure 3: Influence of c_q parameter - Model 3 a=200 - RBF:Rotated gaussian

Analyzing the influence of c_q parameter, we could say that better results are found when the weight of the function is concentrated near the evaluation point. An analysis of the results obtained with the rotated Gaussian shows that error decreases as we reduce the k constant, up to a point where a sudden increment in the error is caused by the ill-conditioning of resulting matrices. This is similar to the results obtained by Zuppa and Cardona[9] using multiquadrics, and also similar to the behavior observed for other meshless methods by Simonetti and Cardona.[7]

3.2 Cloud determination

Extensive tests were carried out to determine the best way to define the clouds used for interpolation at a given point. From a qualitative point of view, for smooth functions and uniform grids of points, the best results were found using a cloud radius approximately equal to $2h$, where h is the distance between two consecutive points of the grid.

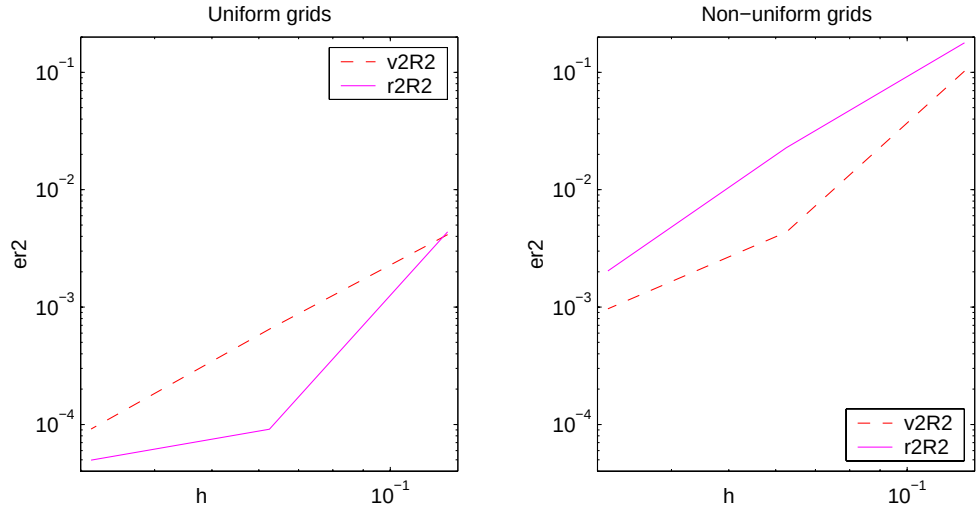

Figure 4: Influence of cloud determination - Model 2

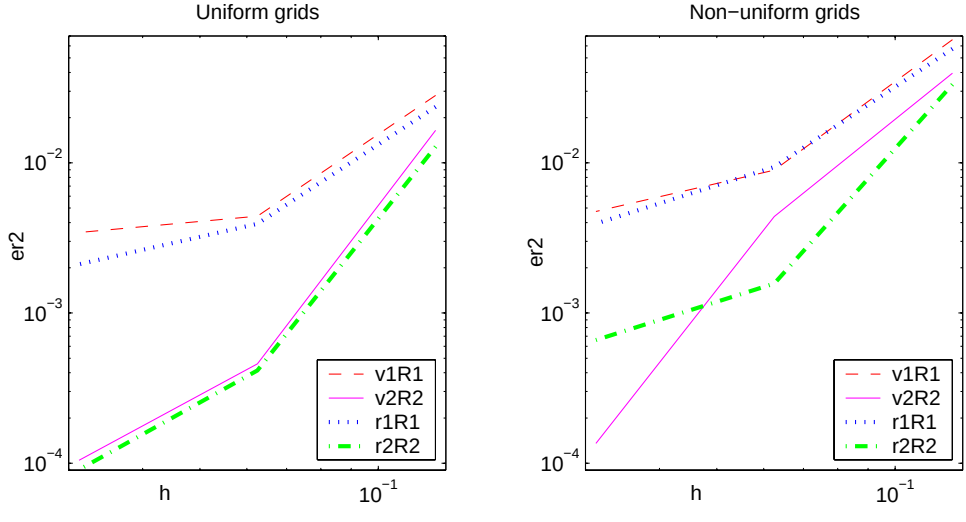

Figure 5: Influence of cloud determination - Model 3 a=200

Using fast Delaunay triangulation algorithms (see e.g. Calvo et al.[2]) with a quasi-linear computational cost, a cloud determination method based on the evaluation of neighbors in a Delaunay triangulation is 20% faster than traditional methods (circular ball) based on computing distances between the evaluation point and the other nodes.

Figures 4 and 5 show a comparison between results using these schemes. In all cases we used as RBF the multiquadrics with $k = 14$ in c_q calculation. Cloud radius parameter r_c was 1.5 for $v1$ (first neighbors) and $r1$ (circular ball) cases, and 2.5 for $v2$ (second neighbors) and $r2$ (circular ball) cases.

The use of clouds with radius near $2h$ is clearly advantageous. Moreover, small radius clouds (and use of order 1 reproduction) are not appropiated for Neumann boundaries.

There is only one case where we detected clear advantages for small radius clouds. As it can be seen in the next section, the use of small radius or first neighbor schemes near Dirichlet boundaries improves the accuracy of the method. This could be explained in part because in these nodes second-neighbor clouds become very asymmetric. In next section we show the results obtained with this scheme.

It can be observed that solutions obtained using circular ball or neighbor methods are of similar quality in uniform grids, with advantages for the neighbor method in non-uniform grids. Based on this fact and its advantages in computational cost, we prefer the neighbor method to construct the clouds.

The neighbor method permits also to use an adaptive scheme to determine the cloud radius parameter r_c used to normalize the RBF. Best results were found using a cloud radius equal to the maximum distance from the evaluation point to another point in the cloud $r_c = \max(\left|\mathbf{x} - \mathbf{x}_c\right|)$.

3.3 Polynomial reproduction order

For analysis of smooth functions, order 2 reproduction gives better results than order 1 reproduction. However, computations are more expensive and many times the improvement in results is not worth the price of the increment in the computational cost.

As seen in figures 6 and 7, order 2 reproduction is clearly advantageous only in cases involving Neumann boundaries (like Model 2).

4 POINT DISTRIBUTION REFINEMENT

The original formulation of LOPI produced bad quality results when point distributions were refined near high-gradient zones, specially in the proximity of Dirichlet boundaries.

Unidimensional and bidimensional models indicated a severe increment in the error when non-uniform grids were used. This problem was solved in part by using a progressive grid refinement. However, the problem continued in the vicinity of Dirichlet boundaries. As seen in fig.11, a successful way to solve this problem was the use of first neighbor clouds combined with order 1 polynomial reproduction in clouds containing Dirichlet points.

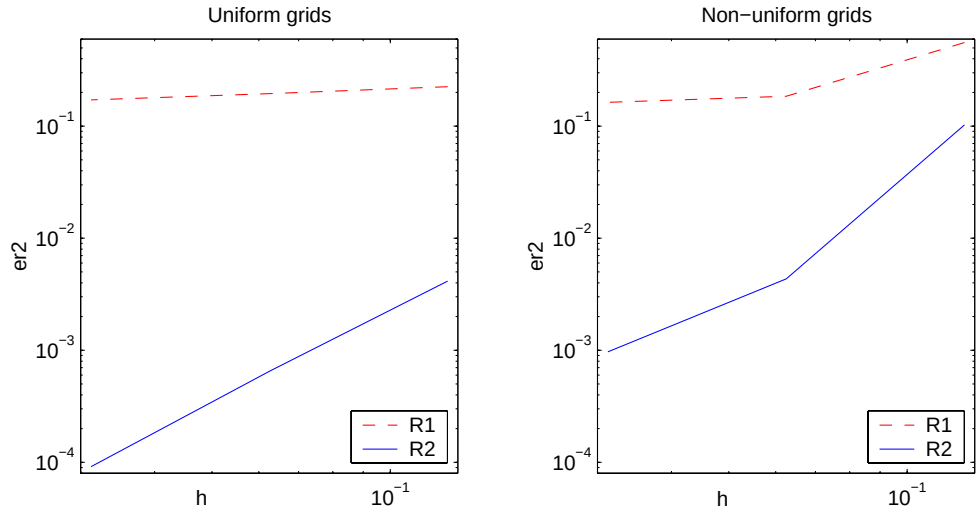

Figure 6: Influence of reproduction order - Model 2

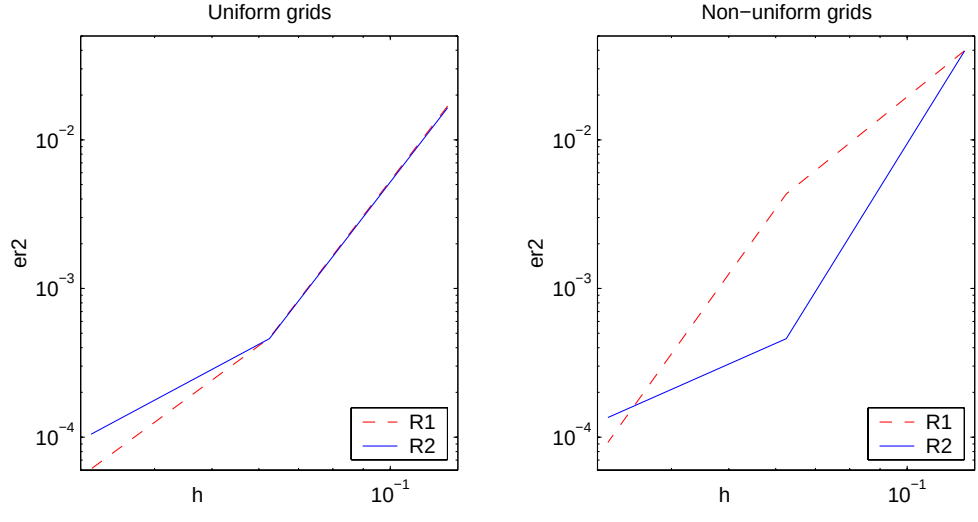

Figure 7: Influence of reproduction order - Model 3 a=200

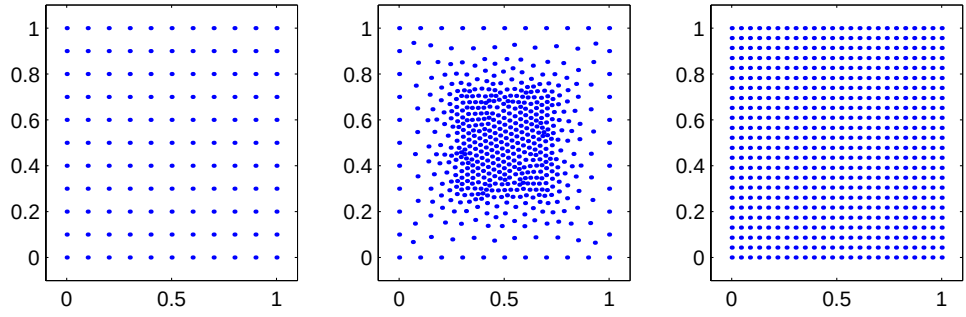

Figure 8: Grids used to compare refinement behavior

To evaluate the behavior of LOPI method in refined grids, we compare results for 3 different meshes: an original 121-points uniform mesh, a 535-points refined mesh, and a 576-points uniform mesh.

In the first example, we start using Model 3 (exponential source) with $a = 200$ in a 11x11 grid (fig.9). In fig.10, after refinement and using the original LOPI formulation, high errors can be seen near Dirichlet boundaries, so the quality of the solution is poor.

The improvement in the behavior of LOPI using local v1R1 schemes is clear analyzing fig.11. In this scheme, the errors are of the same order than in an uniform mesh of the same size (fig.12).

case	er_2	er_∞
first mesh	0.00620	0.04391
refined − original	0.00200	0.00293
refined − new method	0.00030	0.00082
uniformly refined	0.00016	0.00078

Table 1: Model 3 a=200. Computed errors

The behavior of the refined grid is even better when using Model 3 with $a = 1000$ (a more concentrated source) . In this case, the refined mesh covers not only the high-gradient zone, but also a small area around it, and refined grid errors (fig.13) are smaller than those of an uniform grid (fig.14) of the same number of points.

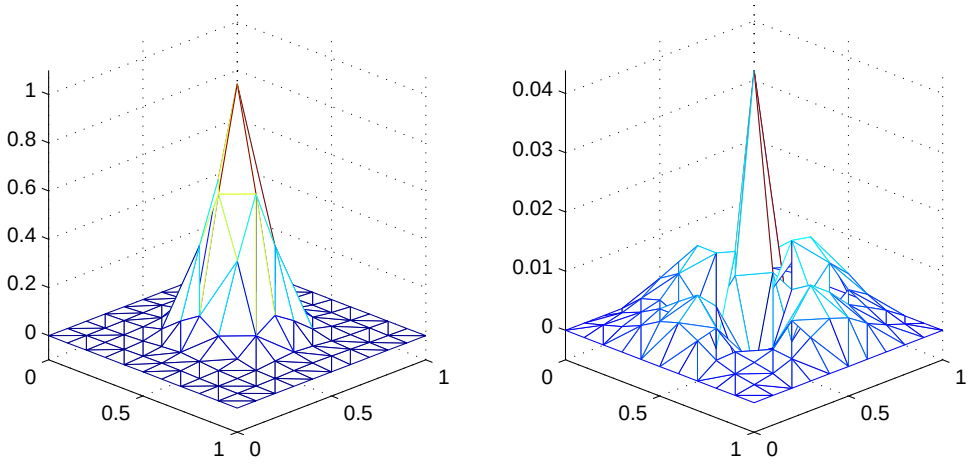

Figure 9: Initial mesh - Model 3 a=200 - Solution and error

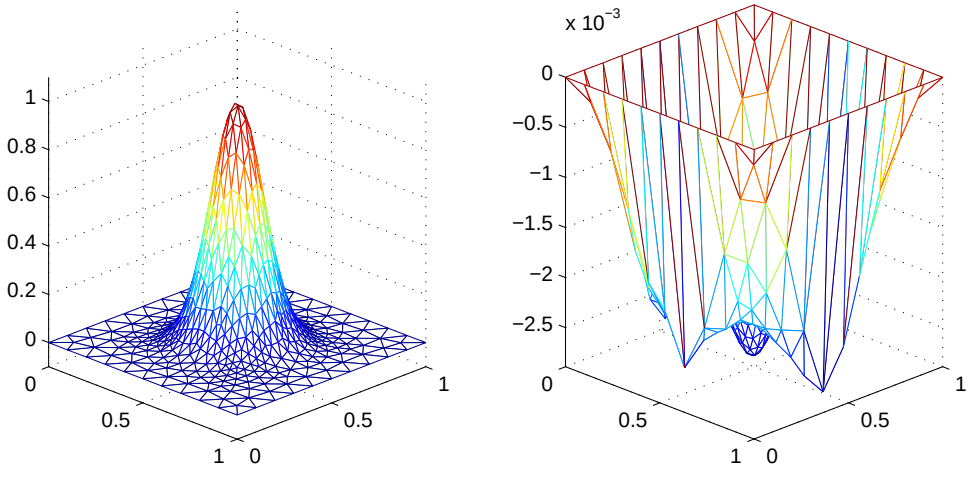

Figure 10: Refined mesh with original method - Model 3 a=200 - Solution and error

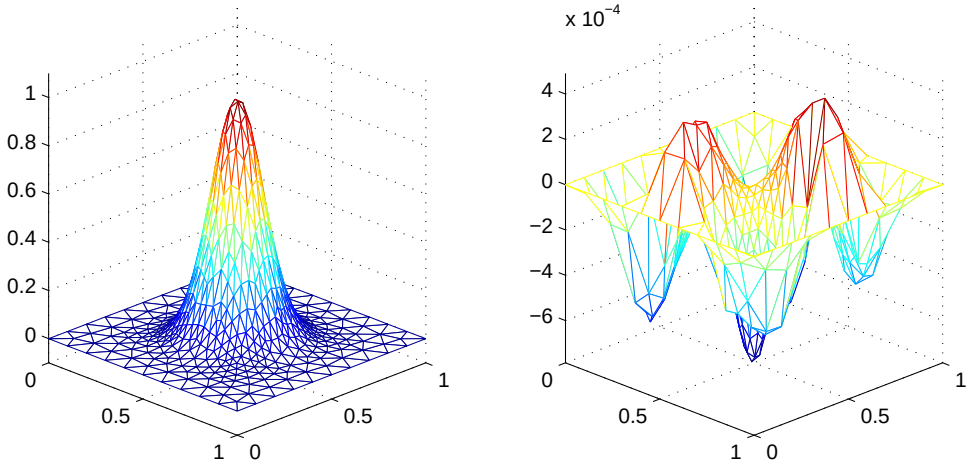

Figure 11: Refined mesh with new method - Model 3 a=200 - Solution and error

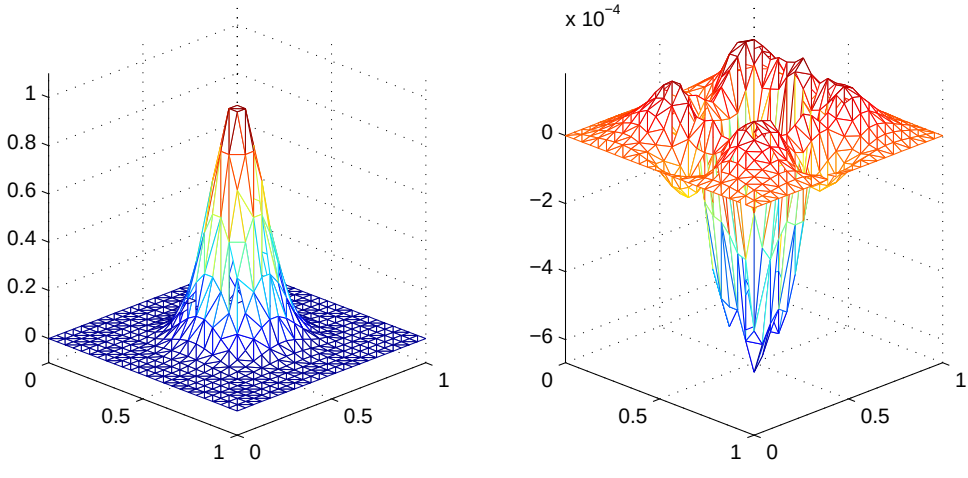

Figure 12: Uniform refined mesh - Model 3 a=200 - Solution and error

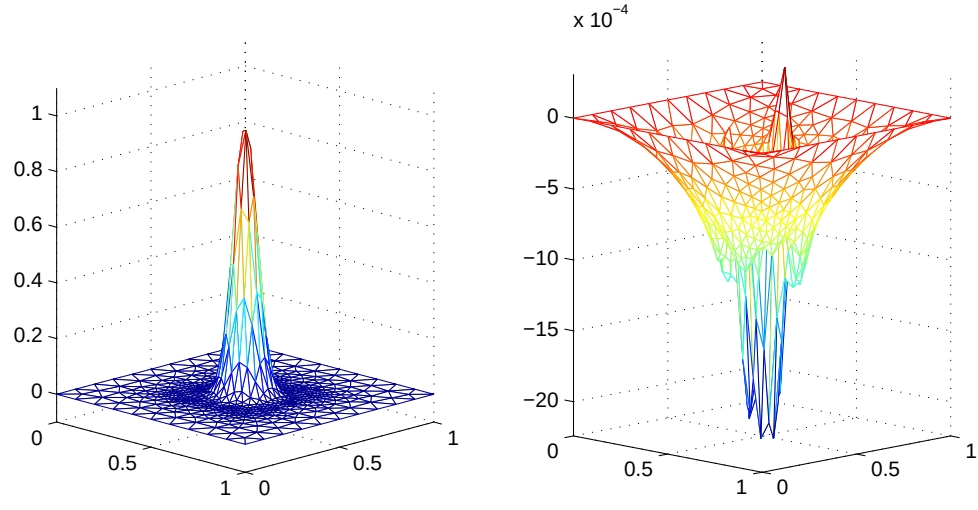

Figure 13: Refined mesh with new method - Model 3 a=1000 - Solution and error

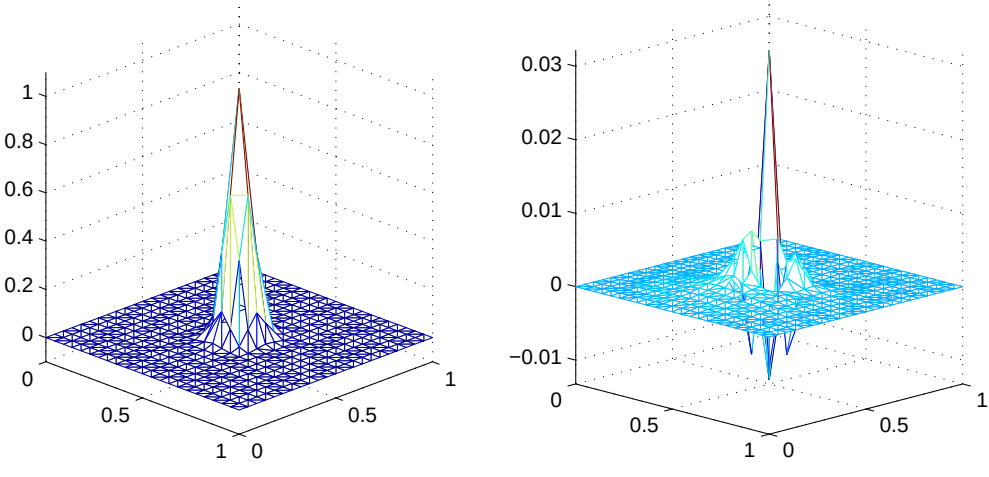

Figure 14: Uniform refined mesh - Model 3 a=1000 - Solution and error

case	er_2	er_∞
refined – new method	0.00072	0.00221
uniformly refined	0.00200	0.03220

Table 2: Model 3 a=1000. Computed errors

5 CONCLUSIONS

The LOPI method works remarkably well in uniform grids, presenting several advantages over competitive methods, as:

- It is "truly meshless", without need of any explicit or background mesh.
- Since the interpolation satisfies the delta Kronecker property, the implementation of Dirichlet boundary conditions is trivial.
- Convergence rates, accuracy and computational cost are better than those of many other meshless methods.

The neighbor method was proved to be the best for cloud determination, and second-neighbor schemes were the best suited for general cases. However, first-neigbor schemes should be used near Dirichlet boundaries. A way to calculate good r_c parameters for RBFs was also determined.

The influence of the dilation parameter k (and c_q) of Radial Basis Functions was analized and best results were found using values that concentrate the weight of the function near the evaluation point, but are also sufficiently conservative to avoid problems due to ill-conditioning of matrices. This was independent of the type of RBF used.

Other parameters, as reproduction order have a less definite behavior, and could be selected in a case-by-case basis, depending of the balance between accuracy and computational cost.

Even when refinement behavior was improved, further work must be done to improve results using non-uniform grids, specially when high gradients of point density are found.

Applications of LOPI to solve Navier-Stokes equations in incompressible fluid flow will be subject of future research.

REFERENCES

[1] T. Belytchsko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl, "Meshless methods: An overview and recent developments", *Computer Methods in Applied Mechanics and Engineering*,**139**,3–48,(1996).

[2] N. Calvo, S.R.Idelsohn and E.Oñate, "The Extended Delaunay Tessellation", *International Journal of Numerical Methods in Engineering*, to appear.

[3] P. Lancaster and K. Salkauskas, *Curve and Surface Fitting: An Introduction*,Academic Press, San Diego, (1986).

[4] S. Li and W.K. Liu, "Meshfree and particle methods and their applications",*Applied Mechanics Review*, to appear.

[5] W.K. Liu, Y. Chen, S. Jun, J.S. Chen, T. Belytchsko, R.A. Uras, and C.T. Chang, "Overview and applications of the reproducing kernel particle methods", *Archives of Computational Mechanics in Engineering: State of the Art and Reviews*, **3**,3–80, (1996).

[6] F. Del Pin, S. Idelsohn, N. Calvo, and M. Storti, "Comparación del método de elementos finitos con métodos meshless en nubes de puntos random", In F. Flores et al., editor, *Mecánica Computacional - Actas ENIEF*, Córdoba, Argentina, Asociación Argentina de Mecánica Computacional, (2001) .

[7] G. Simonetti and A. Cardona, "Métodos sin malla para resolver la ecuación de conducción del calor", *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, **16**,33–47, (2000).

[8] C. Zuppa, "A local optimal point interpolation formula", *Revista de Matemática Aplicada, Universidad de Chile*, **23**,7–28, (2002).

[9] C. Zuppa. and A. Cardona, "A Collocation Meshless Method based on Local Optimal Point Interpolation", *International Journal of Numerical Methods in Engineering*, to appear.