An efficient linear-precision partition of unity basis for unstructured meshless methods

Petr Krysl* Ted Belytschko†

October 1, 1999

Abstract

We describe an approach to construct approximation basis functions for meshless methods, which is based on the concept of a partition of unity. The approach has the following properties: (i) the grid consists of scattered nodes, (ii) the basis reproduces exactly complete linear polynomials, (iii) only the values of the approximated function at the nodes are used as unknowns, (iv) the construction of the basis is only slightly more expensive than the Shepard constant-precision method, and finally, (v) the method is applicable in any number of spatial dimensions.

Keywords: Meshless methods, partition of unity, linear-precision basis

Introduction

The partition of unity (PU) approach to the construction of interpolations and approximations has been known for a long time, although perhaps not explicitly recognized as such. The methods of Shepard[1], McLain[2,3], Franke and Nielson[4] are all in this category. In recent years, the partition of unity method (PUM) has received increased attention especially due to the work of Babuška and Melenk[5–7] and Duarte and Oden[8–10]. The element-free Galerkin method as introduced by Belytschko, Lu and Gu[11] also generates a partition-of-unity basis (the nodal functions are constants). The main reasons for the upsurge of interest are the potentially meshless character of methods based on these approximations, and their good approximation properties. The criteria of when a method can be classified as "meshless" are not well established in the literature. We consider a method as meshless if the approximation basis is constructed on arbitrarily

*Research Assistant Professor, Civil Engineering, Northwestern University; currently with California Institute of Technology, Pasadena, CA.

†Walter P. Murphy Professor of Civil and Mechanical Engineering, Northwestern University, Evanston, IL, USA.

overlapping supports of scattered nodes, without recourse to a partition of the domain into non-overlapping subdomains of the finite element type.

The PUM can be concisely described as follows. Let the union of compact, overlapping sets Ω_I be a cover of the domain Ω, and construct PU functions $\omega_I(\boldsymbol{x})$ subordinate to this cover. On each set Ω_I construct an approximation space $V_I(\boldsymbol{x})$, which should be able to locally capture the solution. The global approximation space is then defined as a blend of the local approximation spaces through the PU functions, $V = \sum_I \omega_I(\boldsymbol{x}) V_I(\boldsymbol{x})$.

To implement the PUM there are two issues to be resolved. First, how to construct the PU functions $\omega_I(\boldsymbol{x})$, and second, how to design the local approximation spaces $V_I(\boldsymbol{x})$. If a meshless character of the approximation is important, the PU can be constructed in a well-known way from weight functions $W_I(\boldsymbol{x})$ defined on the sets Ω_I: $\omega_I(\boldsymbol{x}) = W_I(\boldsymbol{x}) / \sum W_k(\boldsymbol{x})$. These partition of unity functions are identical with the Shepard basis[1].

The local approximation spaces can be tailored to the known properties of the solution[6], or they can be polynomial spaces generated by a Taylor series expansion of the solution. In many cases, a polynomial is a good choice for the local spaces. The order of the polynomial space should be chosen judiciously. For example, for second order partial differential equations in elasticity at least a linear polynomial should be reproduced exactly. This is the so-called consistency condition, which is related to the patch test of Strang and Fix[12]. For the Taylor series approximation space, this leads to degrees of freedom which correspond to the derivatives of the solution. This may be undesirable, because it may well lead to numerical difficulties due to the conditioning of the global matrices. Therefore, local spaces $V_I(\boldsymbol{x})$ in which the coefficients are all nodal values of the solution can be advantageous. Such spaces have been proposed by Babuška and Melenk[6] in 1-D in the form of Lagrange interpolation polynomials. The extension of this idea to two and more dimensions is difficult.

In this paper, we present an approach to the construction of a linear PU basis. We use the Shepard functions as the PU. In contrast to other PU methods, we design the nodal functions so that the degrees of freedom are exclusively the values of the approximated function at the nodes (or their equivalents if the non-interpolating Shepard basis is used). The benefits of this approach include better conditioning of the discrete equations and easier handling of essential boundary conditions in applications to PDE's. Furthemore, compared to moving least squares approximations, the construction of the present basis is quite fast.

The outline of the paper is as follows: In Section 1 we review the Shepard's method, which is used as the partition of unity. We list the properties of the weight functions, and the characteristics of the functions $\omega_I(\boldsymbol{x})$. Section 2 then deals with modifications of the Shepard method, which have been proposed with the goal of making the Shepard interpolation more accurate. These improvements can be recognized as variants of the general PUM. In Section 3 we explain the construction of the local spaces $V_I(\boldsymbol{x})$ for the proposed approximation. Sec-

tion 4 illustrates the properties of the present approach, such as accuracy, and performance, on a number of numerical examples.

1 Shepard's method

We consider discretization of domain Ω with an approximation based on a set of scattered nodes. Each node affects the approximation in its neighborhood, or *domain of influence* Ω_I. The domains of influence can be of any shape: square, circular, etc. A weight function $W_I(\boldsymbol{x})$ is associated with each node I. It is non-negative inside Ω_I, vanishes on the boundary $\partial\Omega_I$, and is non-zero at node I.

The Shepard's approximation[1] can be written as

$$u_h(\boldsymbol{x}) = \sum_{I=1}^{N} \omega_I(\boldsymbol{x}) u_I \ , \tag{1.1}$$

where u_I are the nodal parameters, and $\omega_I(\boldsymbol{x})$ are the basis functions of compact support, which are constructed from the weight functions associated with the nodes, $W_I(\boldsymbol{x})$, by

$$\omega_I(\boldsymbol{x}) = \frac{W_I(\boldsymbol{x})}{\sum_{k=1}^{N} W_k(\boldsymbol{x})} \ . \tag{1.2}$$

It is trivially shown that

$$\sum_{I=1}^{N} \omega_I(\boldsymbol{x}) = 1 \ . \tag{1.3}$$

Equation (1.3) expresses the fact that the functions $\omega_I(\boldsymbol{x})$ represent a partition of unity: A constant function $u(\boldsymbol{x}) = C$ is reproduced exactly. (This property is also called "constant precision".) If $u_I = C$, for $I = 1, \ldots, N$, it follows from (1.1) that

$$u_h(\boldsymbol{x}) = \sum_{I=1}^{N} \omega_I(\boldsymbol{x}) u_I = \sum_{I=1}^{N} \omega_I(\boldsymbol{x}) C = C \sum_{I=1}^{N} \omega_I(\boldsymbol{x}) = 1 \ . \tag{1.4}$$

Other properties of the Shepard functions depend on the weight functions. To achieve interpolation, Franke and Nielson[4] propose the following singular weight function with compact-support

$$W_I(\boldsymbol{x}_I, \boldsymbol{x}) = \begin{cases} \left[\dfrac{(R_w - ||\boldsymbol{x} - \boldsymbol{x}_I||)}{R_w ||\boldsymbol{x} - \boldsymbol{x}_I||} \right]^2 & \text{for } ||\boldsymbol{x} - \boldsymbol{x}_I|| < R_w, \\ 0 & \text{elsewhere,} \end{cases} \tag{1.5}$$

where R_w is the radius of the support. The PU basis $\omega_I(\boldsymbol{x})$ generated via (1.2) from the weight function of (1.5) has the following properties:

$$\omega_I(\boldsymbol{x}_K) = \delta_{IK} \tag{1.6}$$

$$\frac{\partial\,\omega_I(\boldsymbol{x}_K)}{\partial\,x_m} = 0\ , \tag{1.7}$$

where the derivatives are with respect to all spatial dimensions, $m = 1, 2, \ldots$, and δ_{IK} is the Kronecker delta.

If interpolation is not required, a non-singular weight function is an option. In this work we have used the radial quartic weight function

$$W_I(\boldsymbol{x}) = \begin{cases} (1 - 6r^2 + 8r^3 - 3r^4) & \text{for } 1 > r \geq 0, \\ 0 & \text{for } r \geq 1, \end{cases} \tag{1.8}$$

where $r = ||\boldsymbol{x} - \boldsymbol{x}_I||/R_w$. Other choices of the weight function are also acceptable, but the above is one of the simplest; see Reference 13.

2 Modified Shepard's method

The PU basis $\omega_I(\boldsymbol{x})$ defined above is only of constant precision. Therefore, the approximation (1.1) does not have much appeal. It does not converge for second-order partial differential equations, and it displays so-called flat spots at the nodes[14]. In order to enhance the approximation, Franke and Nielson have proposed a modification leading to an inverse distance weighted least-square interpolation[4], which can be explained in the framework of a general PU method.

The basic idea is to replace the nodal values in (1.1) by a local approximating functions $V_I(\boldsymbol{x})$ (also called a local fit, or nodal function[4,15]; the span of this space can be chosen to fit the expected behaviour of the solution, or may be selected to reproduce polynomials of certain degree

$$u_h(\boldsymbol{x}) = \sum_{I=1}^{N} \omega_I(\boldsymbol{x}) V_I(\boldsymbol{x}) \tag{2.1}$$

where $\omega_I(\boldsymbol{x})$ is the PU function. The resulting approximation is a blend of the functions $V_I(\boldsymbol{x})$ through the PU basis $\omega_I(\boldsymbol{x})$.

The approximation (2.1) reproduces exactly any function contained in all the $V_I(\boldsymbol{x})$ functions. Let nodal functions $V_I(\boldsymbol{x})$ contain the function $f(\boldsymbol{x})$. Then we can rewrite (2.1) as

$$u_h(\boldsymbol{x}) = \sum_{I=1}^{N} \omega_I(\boldsymbol{x}) V_I(\boldsymbol{x}) = \sum_{I=1}^{N} \omega_I(\boldsymbol{x}) f(\boldsymbol{x}) = f(\boldsymbol{x}) \sum_{I=1}^{N} \omega_I(\boldsymbol{x}) = f(\boldsymbol{x})\ , \tag{2.2}$$

where the last step follows from (1.3).

If the singular weight function (1.5) is used, the approximation (2.1) interpolates at the nodes, and the derivatives of the approximation at the nodes are equal to the derivatives of the nodal functions $V_I(\boldsymbol{x})$. This follows from the properties (1.6) and (1.7) of the singular weights.

Shepard proposed using the derivative data (linear terms of a Taylor series) to achieve linear precision[1] of the approximation basis. In order to enhance the capabilities of the method for data fitting (surface approximation), Franke and Nielson[4] proposed the quadratic local least-squares fit

$$\begin{aligned} V_I(\boldsymbol{x}) &= c_1^{(I)}(x-x_I)^2 + c_2^{(I)}(x-x_I)(y-y_I) + c_3^{(I)}(y-y_I)^2 \\ &\quad + c_4^{(I)}(x-x_I) + c_5^{(I)}(y-y_I) + c_6^{(I)} \end{aligned} \tag{2.3}$$

If the PU functions $\omega_I(\boldsymbol{x})$ allow for interpolation, the coefficient $c_6^{(I)}$ can be identified with the value of the approximated function at the node I; otherwise it needs to be determined by a weighted least-squares fit[4]. The fact that the coefficients $c_j^{(I)}, j = 1, \ldots, 5$ correspond to the derivatives of the solution, is a distinct disadvantage for applications of the approximation (2.1) in numerical solutions of partial differential equations: the nodal parameters have different physical dimensions, and the number of degrees of freedom per node increases. Therefore, a formulation using only the values of the sought function at nodes as degrees of freedom is desirable.

3 Linear nodal PU approximation

Our goal is to design the local approximating space in such a way as to achieve (i) linear precision, and (ii) use of only one type of nodal parameter. Therefore, we are looking for a set of linear functions $\widehat{\mu}_m^{(I)}(\boldsymbol{x})$ such that

$$V_I(\boldsymbol{x}) = \widehat{\mu}_I^{(I)}(\boldsymbol{x})u_I + \sum_{k=1}^{N_I} \widehat{\mu}_k^{(I)}(\boldsymbol{x})u_k \ ,$$

where u_I is the nodal parameter associated with the node I, and u_k are the nodal parameters associated with some other nodes, which are "close" to node I. We shall call this set of nodes the *star nodes*; see Fig. 1, where the star nodes are enclosed in square boxes. The selection of the star nodes is quite arbitrary. Note especially that the star nodes are not required to be in the domain of influence of the node I, and node I is not required to be in the domains of influence of the star nodes. However, since the star nodes are essential in determining the derivative data, the closest nodes to the node I can be expected to work best. (Figure 1 looks similar to those used by Perrone and Kao [16] to explain their irregular finite difference method. The methods are not related, though.)

If the singular weights are used, interpolation may be achieved if the nodal function V_I interpolate at the node I for any values of nodal parameters at the

nodes of the nodal function. Furthermore, if the nodal parameters are set to values of a linear polynomial function at the nodes, the nodal function should interpolate at all the nodes contained by the V_I. The last requirement is posed so that in this case we *know* the nodal parameters to be the values of the linear function at the nodes, and we can formulate the linear consistency conditions.

Given the above, we write the nodal function as

$$V_I(\boldsymbol{x}) = u_I + \sum_{k=1}^{N_I} \mu_k^{(I)}(\boldsymbol{x})(u_k - u_I) = u_I \left(1 - \sum_{k=1}^{N_I} \mu_k^{(I)}(\boldsymbol{x})\right) + \sum_{k=1}^{N_I} \mu_k^{(I)}(\boldsymbol{x}) u_k \ . \tag{3.1}$$

The functions $\mu_k^{(I)}(\boldsymbol{x})$ are associated with node k in the star of node I, as indicated by the superscript (I).

In what follows, we shall specialize the discussion to a two-dimensional setting. However, the conclusions also apply to three and more dimensions. To facilitate further developments, we introduce another Cartesian coordinate system, $\bar{x}, \bar{y}$, parallel to the global Cartesian axes x, y. Its origin is placed at node I, as shown in Fig. 1.

Local basis functions. For the nodal function V_I to interpolate at node I, we construct the linear functions $\mu_k^{(I)}(\boldsymbol{x})$ so as to vanish at node I. The functions $\mu_k^{(I)}(\boldsymbol{x})$ associated with the star nodes will be

$$\mu_k^{(I)}(\boldsymbol{x}) = \mu_k^{(I)}(\bar{\boldsymbol{x}}) = a_k^{(I)} \frac{\bar{x}}{d_k} + b_k^{(I)} \frac{\bar{y}}{d_k} \ , \tag{3.2}$$

where d_k is the distance of the star node k from the node I, $d_k = ||\bar{\boldsymbol{x}}_k||$. The scaling by d_k is introduced to make the coefficients $a_k^{(I)}$, $b_k^{(I)}$ non-dimensional, thus minimizing numerical errors in the linear algebra operations described below.

In order to fully specify $V_I(\boldsymbol{x})$, we need to find the coefficients $a_k^{(I)}$, $b_k^{(I)}$ for each star node $k = 1, \ldots, N_I$. Our goal is to construct a basis of linear precision, which reproduces a linear function exactly, so the nodal functions need to be able to reproduce exactly any complete linear polynomial, $f(\boldsymbol{x}) = \alpha x + \beta y + \gamma$. Equivalently, we may introduce a linear transformation, $\bar{x} = x - x_I, \bar{y} = y - y_I$, and reformulate the polynomial to be reproduced as $f(\boldsymbol{x}) = A\bar{x} + B\bar{y} + C$ ($A = \alpha$, $B = \beta$, and $C = \gamma + \alpha x_I + \beta y_I$).

The nodal functions $V_I(\boldsymbol{x})$ are constructed so as to interpolate at node I and all the star nodes for nodal parameters corresponding to a linear polynomial, $u_m = f(\boldsymbol{x}_m) = A\bar{x}_m + B\bar{y}_m + C$. Therefore (note that $\bar{x}_I = \bar{y}_I = 0$)

$$\begin{aligned} V_I(\boldsymbol{x}) = f(\boldsymbol{x}) &= A\bar{x} + B\bar{y} + C \\ &= f(\boldsymbol{x}_I)\left(1 - \textstyle\sum_{k=1}^{N_I} \mu_k^{(I)}(\boldsymbol{x})\right) + \sum_{k=1}^{N_I} \mu_k^{(I)}(\boldsymbol{x}) f(\boldsymbol{x}_k) \\ &= C\left(1 - \textstyle\sum_{k=1}^{N_I} \mu_k^{(I)}(\boldsymbol{x})\right) + \sum_{k=1}^{N_I} \mu_k^{(I)}(\boldsymbol{x})(A\bar{x}_k + B\bar{y}_k + C) \\ &= C + \textstyle\sum_{k=1}^{N_I} \mu_k^{(I)}(\boldsymbol{x})(A\bar{x}_k + B\bar{y}_k) \ . \end{aligned} \tag{3.3}$$

Constant precision is guaranteed by construction; compare with (3.1). Linear precision is achieved if the following equations hold for any A and B

$$\sum_{k=1}^{N_I} a_k^{(I)} \left(A\frac{\bar{x}_k}{d_k} + B\frac{\bar{y}_k}{d_k} \right) = A \tag{3.4}$$

$$\sum_{k=1}^{N_I} b_k^{(I)} \left(A\frac{\bar{x}_k}{d_k} + B\frac{\bar{y}_k}{d_k} \right) = B \tag{3.5}$$

Since the above must hold for arbitrary A and B, the coefficients $a_k^{(I)}$ have to satisfy the following two equations

$$\sum_{k=1}^{N_I} a_k^{(I)} \frac{\bar{x}_k}{d_k} = 1 \tag{3.6}$$

$$\sum_{k=1}^{N_I} a_k^{(I)} \frac{\bar{y}_k}{d_k} = 0 \tag{3.7}$$

and the coefficients $b_k^{(I)}$ have to satisfy

$$\sum_{k=1}^{N_I} b_k^{(I)} \frac{\bar{x}_k}{d_k} = 0 \tag{3.8}$$

$$\sum_{k=1}^{N_I} b_k^{(I)} \frac{\bar{y}_k}{d_k} = 1 \tag{3.9}$$

It becomes clear that there must be at least two star nodes in order to be able to construct a basis with a linear precision: in that case there are two coefficients $a_k^{(I)}$ and two coefficients $b_k^{(I)}$ so equations (3.6)–(3.9) can be met. If there are more than two star nodes, one can solve for two of the coefficients, but the values of the remaining $(N_I - 2)$ coefficients can be chosen arbitrarily; the selection of the coefficients is discussed below.

Existence of a solution. When does a unique solution of the set of equations (3.6) and (3.7) exist? Let us assume we solve for the coefficients a_1 and a_2. The left-hand side matrix is then

$$[L] = \begin{bmatrix} \bar{x}_1/d_1 & \bar{x}_2/d_2 \\ \bar{y}_1/d_1 & \bar{y}_2/d_2 \end{bmatrix} \tag{3.10}$$

It is easy to see that the columns are linearly independent provided the star nodes 1 and 2 are not located on a line radially emanating from the node I, or in other words, matrix $[L]$ is non-singular if the vectors $\{\bar{x}_1, \bar{y}_1\}$ and $\{\bar{x}_2, \bar{y}_2\}$ are linearly independent.

Determination of the coefficients. How should one select the coefficients $a_k^{(I)}$ and $b_k^{(I)}$ for $k = 3, \ldots, N_I$? Because of (1.7), these coefficients determine

the slope of the approximated function at the nodes. The question is then, how should the coefficients $a_k^{(I)}$ be distributed to achieve a good approximation to the derivatives of a given function at the node I? (Analogous reasoning applies to the determination of the coefficients $b_k^{(I)}$.)

It is reasonable to require some kind of symmetry of the distribution with respect to the axes $\bar{x}, \bar{y}$. Also, for a perfectly symmetric distribution of nodes the coefficients should be also symmetric. For instance, for the star nodes of Fig. 2, both distributions of the coefficients $a_k^{(I)}, k = 0, \ldots, 7$ are acceptable. The left side in fact loosely corresponds to a central difference estimate of the first derivative with respect to x at the central node. One can expect a connection between the design of the nodal function in the present method and the finite difference approximations of derivative data. In particular, the connection to the generalized finite difference method of Liszka and Orkisz [17] seems worth exploring.

We propose the following rule for the initial selection of the coefficients $a_k^{(I)}$ for any irregular distribution of the star nodes (analogous estimates apply to the coefficients $b_k^{(I)}$)

$$a_k^{(I)} = \left(\sum_{m=1}^{N_I} \frac{\bar{x}_m^2}{d_m^2} \right)^{-1} \frac{\bar{x}_k}{d_k} \ . \tag{3.11}$$

As can be easily verified, for symmetric arrangements of the star nodes, (3.11) satisfies (3.6) by construction. If either one of equations (3.6) and (3.7) is not satisfied exactly for the values estimated from (3.11), the values of $a_1^{(I)}$ and $a_2^{(I)}$ are obtained via the solution of (3.6) and (3.7).

Some comments are in order: (i) one can expect deterioration of the absolute accuracy for non-optimal distributions of the coefficients, but the convergence rate is *not* affected, since the linear precision conditions are met, (ii) further investigation is needed to determine if there are optimality criteria for the selection of the stars and for the distribution of the coefficients in the stars.

Implementation. Applying the above results, equation (2.1) can be written as

$$u_h(\boldsymbol{x}) = \sum_{I=1}^{N} \omega_I(\boldsymbol{x}) \left[u_I \left(1 - \sum_{k=1}^{N_I} \mu_k^{(I)}(\boldsymbol{x}) \right) + \sum_{k=1}^{N_I} \mu_k^{(I)}(\boldsymbol{x}) u_k \right] \ , \tag{3.12}$$

which can be also converted to the standard form

$$u_h(\boldsymbol{x}) = \sum_{I=1}^{N} \varphi_I(\boldsymbol{x}) u_I \ , \tag{3.13}$$

where the resulting linear-precision basis functions $\varphi_I(\boldsymbol{x})$ are expressed as

$$\varphi_I(\boldsymbol{x}) = \omega_I(\boldsymbol{x}) \left(1 - \sum_{k=1}^{N_I} \mu_k^{(I)}(\boldsymbol{x}) \right) + \sum_{k=1}^{A_I} \omega_k(\boldsymbol{x}) \mu_I^{(k)}(\boldsymbol{x}) \ . \tag{3.14}$$

Note the distinction in the sums: A_I is the number of stars in which the node I participates.

4 Numerical examples

In order to gain some insight into the properties of the proposed approximation basis, we present L_2 interpolation errors. The schemes compared are:

1. Moving least squares, which are used in the element-free Galerkin (EFG) method, with a linear basis[11,13,18],
2. FEM with linear triangles,
3. Interpolating PUM (the singular weight function of (1.5) is used),
4. Non-interpolating PUM (non-singular weight function of (1.8) is used).

Two surfaces have been approximated, $S(x,y) = (-x^3 + \sin(2y) + xy/2)/4$, and $R(x,y) = -xy\exp(-2x^4y^4)$; both on a square domain $\boldsymbol{x} \in \langle -2;2\rangle^2$. Four grids have been used. The spacing of the grids was set to 1/5, 1/10, 1/20 and 1/40 of the length of the domain edge. An unstructured triangulation with edges approximately the same length was generated for each spacing, and the nodes in the meshless methods were placed at the vertices of the triangulation. (The FEM solution was obtained on the triangulations, of course.) The support size was set to $1.7l_e$ for the meshless methods, EFG and the interpolating and non-interpolating PU; l_e is the average length of edges in the triangulation. The EFG method used a linear basis, and a quartic polynomial weight function on a spherical support. For the EFG and the non-interpolating PU methods the nodal values were found from the interpolation conditions

$$[\phi_I(\boldsymbol{x}_j)]\,\{f_I\} = \{f(\boldsymbol{x}_j)\} \tag{4.1}$$

The star nodes were selected from nodes within the domain of influence of the central node. As will be discussed later, the number of star nodes was either unlimited, or the nodes were sorted by distance, and only the closest to the central node have been considered.

The interpolation error is measured by the L_2 norm

$$||f - f_h||_{L_2}^2 = \int \left[f(\boldsymbol{x}) - f_h(\boldsymbol{x})\right]^2 \mathrm{d}\Omega \ , \tag{4.2}$$

and error in derivatives is measured by the L_2 seminorm

$$||\partial f - \partial f_h||_{L_2}^2 = \int \left\{ \left[\frac{\partial f}{\partial x}(\boldsymbol{x}) - \frac{\partial f_h}{\partial x}(\boldsymbol{x})\right]^2 + \left[\frac{\partial f}{\partial y}(\boldsymbol{x}) - \frac{\partial f_h}{\partial y}(\boldsymbol{x})\right]^2 \right\} \mathrm{d}\Omega \ , \tag{4.3}$$

Figure 3 shows the convergence of the interpolation error norms for the surface S, and the corresponding interpolation errors for the surface R are shown in Fig. 4. The expected convergence rate for linear triangles is two in the norm (4.2), and one in the norm (4.3). It can be seen that the EFG method and the non-interpolating PUM display similar absolute accuracy and convergence rates. The interpolating PUM is only slightly more accurate than the linear finite elements. The kink in Fig. 4 can be probably attributed to pre-asymptotic behavior, but the convergence rate of the EFG and the present non-interpolating PUM is not completely understood yet.

Dependence on the star coefficients In order to assess how the star coefficients influence the accuracy we compare interpolation errors for the surface S for coefficients obtained from (3.11), with results obtained for obviously non-optimal coefficients $a_k = b_k = 0$, $k \geq 3$. As can be seen from Fig. 5, the latter distribution gives slightly worse absolute accuracy, but the convergence rate is, as expected, unchanged.

Dependence on the support size The support size, and in a related manner the number of star nodes selected at a given point, has a modest influence on the absolute approximation errors. To assess this characteristic, we compare the interpolation errors for varying support sizes. The support size is measured in multiples of the smallest nodal spacing for a regular grid. The interpolation errors are evaluated for the surface S on a perfectly regular grid of 10×10 nodes, and on a grid of 10×10 nodes with slightly randomly shifted nodes (approx. 1/5 of the node spacing). Figure 6 shows the interpolation error for the case where all the nodes inside the domain of influence of the center node are selected as star nodes; figure 7 depicts analogous results for the case the star nodes are limited to the four closest nodes in the domain of influence of the center node. The accuracy of the present interpolating PUM is almost independent of the support size; the accuracy of the non-interpolating PUM is comparable to the accuracy of the EFG method, but does not improve as markedly for larger supports. Interestingly, the present approach seems to be less sensitive to the irregularity of the grid than the EFG method.

5 Performance

The cost of the PU method of (2.1) is composed of the cost of constructing the functions $\omega_I(\boldsymbol{x})$ and of the cost involved in the computation of the nodal functions $V_I(\boldsymbol{x})$. Therefore, it is not possible to reduce the total CPU time cost below that needed to construct the functions $\omega_I(\boldsymbol{x})$. We use equation (1.2) to construct the functions $\omega_I(\boldsymbol{x})$ (or, in other words, the Shepard approximation is used as the PU). In view of the preceding argument, we compare the cost of the presented approach with the cost of the EFG method, and also with the cost of the evaluation of the PU functions $\omega_I(\boldsymbol{x})$. The efficiency of the construction of the functions $\omega_I(\boldsymbol{x})$ depends crucially on the efficiency with which nodes associated

with non-zero weight can be found at any given point. We use a bounding-box tree search technique with the EFG method.

Star-based searching. Interestingly, the present approach allows us to use a star-based searching technique which is more efficient than the bounding-box tree search. The star-based search makes use of the fact that one can search for only a single node using the bounding-box tree search, and then the star connectivity can be used to collect all the remaining nodes. (In order to make this work, *all* the nodes in the domain of influence need to be included in the star.)

Timing comparisons. We use a grid with 10,000 nodes, and we evaluate the basis functions and their first derivatives at $10,000 \times M$ randomly scattered points, $M = 1, 4, 9, 16$. We measure the CPU time for the following methods: (i) Shepard method (or construction of the PU functions $\omega_I(\boldsymbol{x})$), (ii) EFG method with a linear basis, (iii) the present approach with a bounding-box tree search structure, (iv) the present approach with star-based searching.

The domains of influence of the nodes are square, and of two sizes leading to 12 and 32 neighbors per evaluation point, respectively. The times are compared in Fig. 8. As can be seen, the present approach using the bounding-box tree search is only slightly slower than the construction of the PU functions $\omega_I(\boldsymbol{x})$. This indicates that the effort expended in making the approximation basis linearly precise is minor, and in fact, the present method can be seen to be close to optimal in the sense that the bulk of the cost is associated with the functions $\omega_I(\boldsymbol{x})$.

As already mentioned, the connectivity information stored with the star at each node allows us to use a more efficient way to construct the functions $\omega_I(\boldsymbol{x})$, since the necessary searches can be performed more quickly. Therefore, the present method in conjunction with the more efficient searching technique is even faster than the construction of the functions $\omega_I(\boldsymbol{x})$ using the slower bounding-box tree searching method; compare with Fig. 8.

Conclusions

We have described how to construct approximation basis functions for a partition-of-unity (PU) method. The PU is in our case the Shepard basis. In order to enhance its accuracy, we have designed linear-precision nodal functions (local fits) in such a way as to involve only the magnitude of the solution at the nodes as the degrees of freedom. The nodal functions are sought as linear combinations of the nodal parameters at the central node and a set of near-by nodes, called the star nodes. The unknown coefficients are computed in a pre-processing step. The resulting approximation is meshless, and it is applicable to any number of spatial dimensions. Two variants can be constructed depending on the weight functions which generate the PU: an interpolating PU method (for singular weight functions), and a non-interpolating PU method. The non-interpolating PU method is comparable in accuracy with the element free Galerkin (moving least squares)

method. The present method is very efficient, in fact the cost of the precision enhancement constitutes only a small fraction of the cost involved in the construction of the Shepard basis. Furthermore, the connectivity information stored in the star allows for more efficient search. In this way the present linear-precision basis can be constructed with the more efficient search technique more quickly than a constant-precision Shepard basis using the usual search strategy.

To summarize, the present approach has the following desirable properties: (i) the grid consists of scattered nodes, (ii) the basis exactly reproduces complete linear polynomials, (iii) only the values of the approximated function at the nodes are used as unknowns, (iv) the construction of the basis is only slightly more expensive than the Shepard constant-precision method, and, (v) the method is applicable in any number of spatial dimensions.

Acknowledgments

The support of the Office of Naval Research is gratefully acknowledged. The anonymous reviewers are thanked for useful comments.

References

[1] D. Shephard. A two dimensional interpolation function for irregularly spaced data. In *Proc. 23rd Nat. Conf. ACM*, pages 517–523. ACM, 1968.

[2] D. H. McLain. Drawing contours from arbitrary data points. *Comput. J.*, 17:318–324, 1974.

[3] D. H. McLain. Two-dimensional interpolation from random data. *Comput. J.*, 19:178–181, 1976.

[4] R. Franke and G. Nielson. Smooth interpolation of large sets of scattered data. *International Journal of Numerical Methods in Engineering*, 15:1691–1704, 1980.

[5] I. Babuška and J. M. Melenk. The partition of unity finite element method. Technical Report BN-1185, Institute for Physical Science and Technology, University of Maryland, 1995.

[6] I. Babuška and J. M. Melenk. The Partition of Unity Method. *International Journal of Numerical Methods in Engineering*, 40:727–758, 1997.

[7] J. M. Melenk and I. Babuška. The Partition of Unity Method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139:289–314, 1996.

[8] C. A. Duarte. A review of some meshless methods to solve partial differential equations. Technical Report 95-06, Texas Institute for Computational and Applied Mathematics, University of Texas at Austin, 1995.

[9] C. A. Duarte and J. T. Oden. *Hp* clouds—a meshless method to solve boundary-value problems. Technical Report 95-05, Texas Institute for Computational and Applied Mathematics, University of Texas at Austin, 1995.

[10] C. A. Duarte and J. T. Oden. *H-p* clouds—an *h-p* meshless method. *Numerical Methods for Partial Differential Equations*, pages 1–34, 1996.

[11] T. Belytschko, Y.Y. Lu, and L. Gu. Element free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37:229–256, 1994.

[12] G. Strang and G. Fix. *An Analysis of the Finite Element Method*. Prentice-Hall, Englewood Cliffs, N.J., 1973.

[13] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139:3–47, 1996.

[14] W. J. Gordon and J. A. Wixom. Shephard's method of metric interpolation to bivariate and multivariate data. *Math. Comput.*, 32:253–264, 1978.

[15] R. Franke. Scattered data interpolation: Test of some methods. *Mathematics of Computation*, 38:181–200, 1982.

[16] N. Perrone and R. Kao. A general finite difference method for arbitrary meshes. *Computers and Structures*, 5:45–47, 1975.

[17] T. Liszka and J. Orkisz. The finite difference method for arbitrary irregular meshes — a variational approach to applied mechanics problems. *GAMNI*, pages 227–235, 1980.

[18] P. Krysl and T. Belytschko. ESFLIB: A library to compute the element free Galerkin shape functions. *Computer Methods in Applied Mechanics and Engineering*, 1998, to appear.

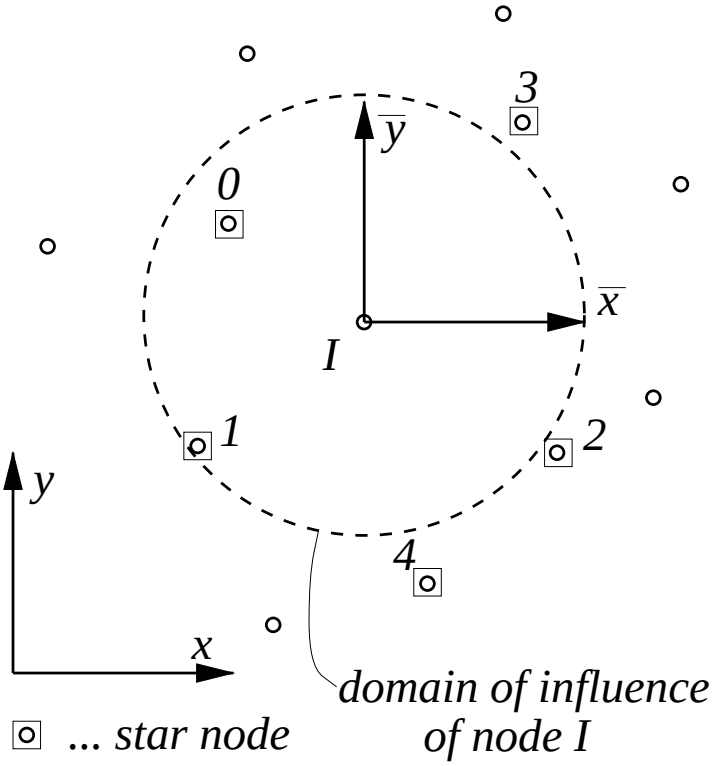


Figure 1: The nodes constituting the star of node I, with the coordinate system $\bar{x}, \bar{y}$.

.

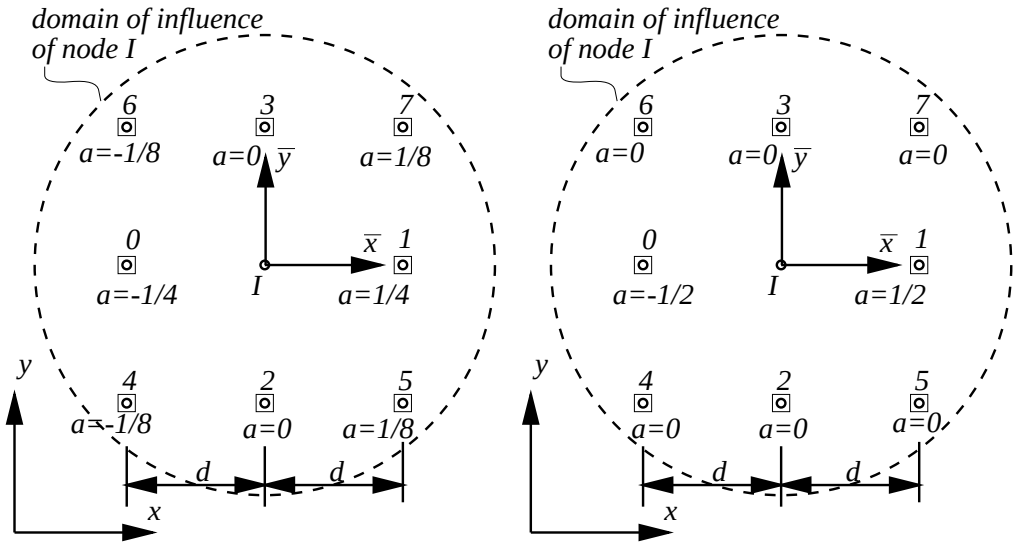
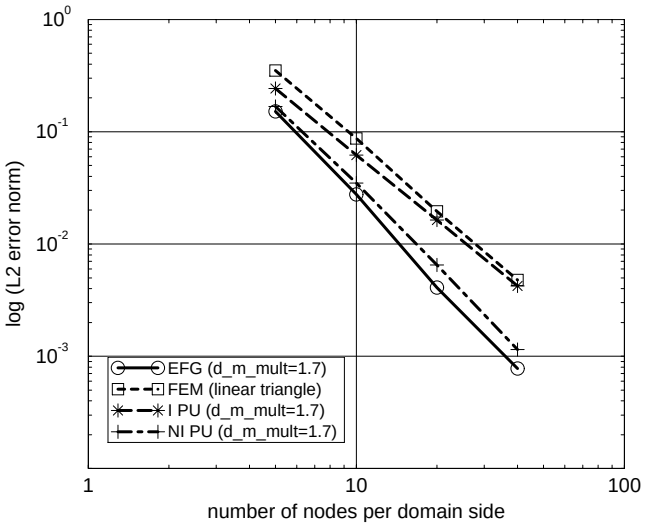
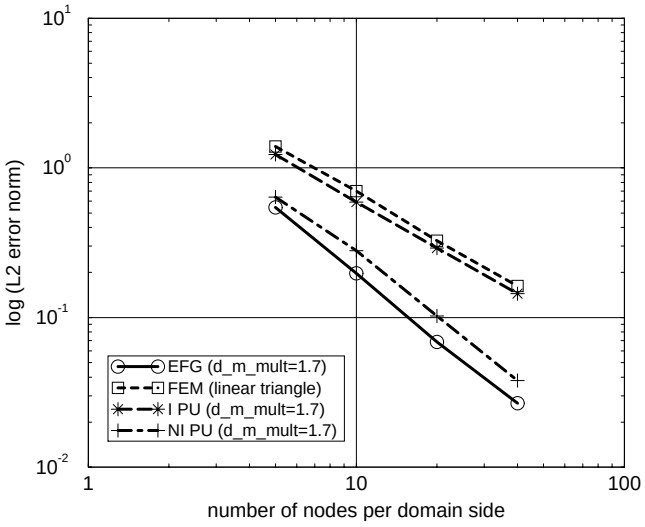


Figure 2: Regular arrangment of nodes.

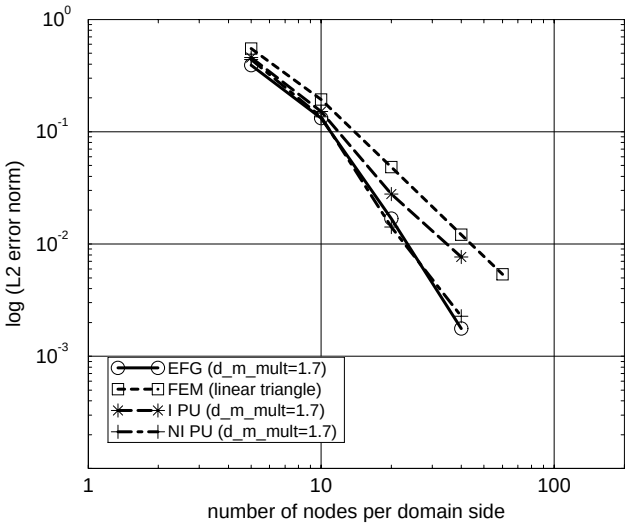


(a) Error $||f - f_h||_{L_2}$.

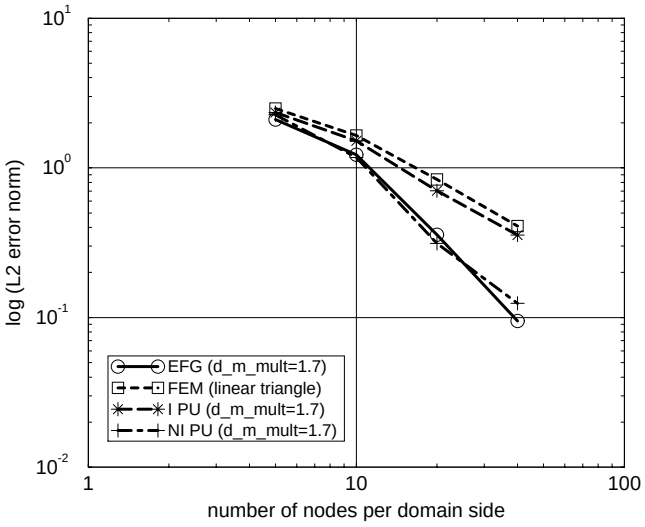


(b) Error $||\partial f - \partial f_h||_{L_2}$.

Figure 3: Interpolation errors for the surface $S(x, y) = (-x^3 + \sin(2y) + xy/2)/4$.



(a) Error $||f - f_h||_{L_2}$.



(b) Error $||\partial f - \partial f_h||_{L_2}$.

Figure 4: Interpolation errors for the surface $R(x, y) = -xy \exp(-2x^4y^4)$.

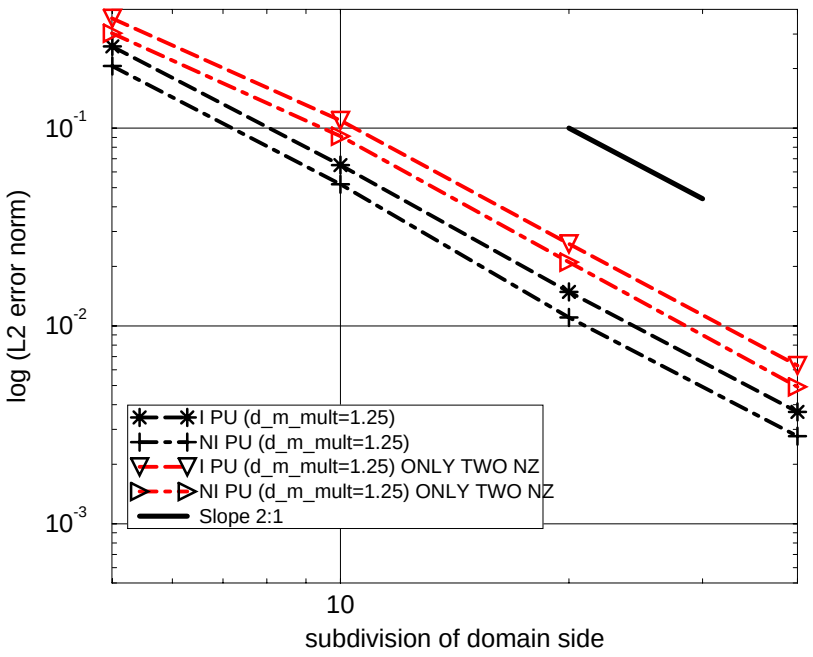
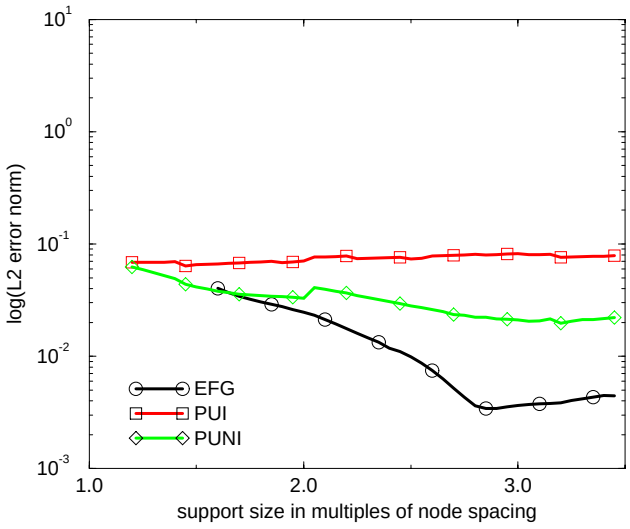
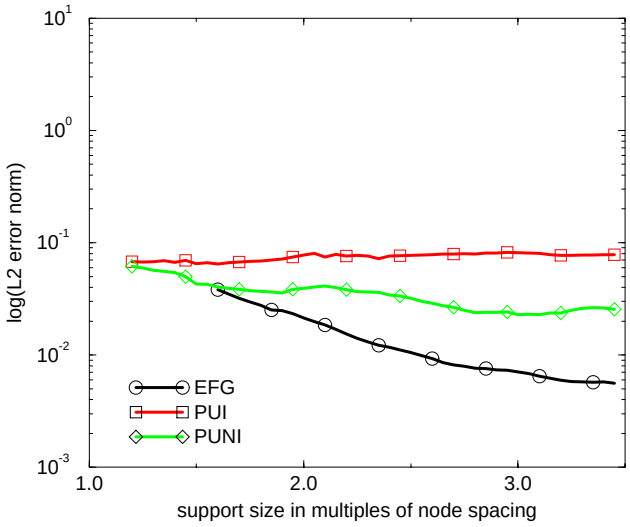


Figure 5: Dependence of the accuracy on the star coefficients, a_k and b_k. `ONLY TWO NZ` means that $a_k = b_k = 0$ for $k \geq 3$.

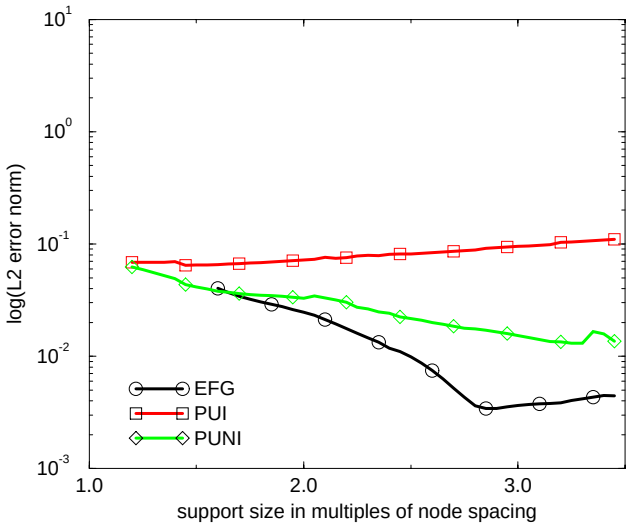


(a) Regular grid

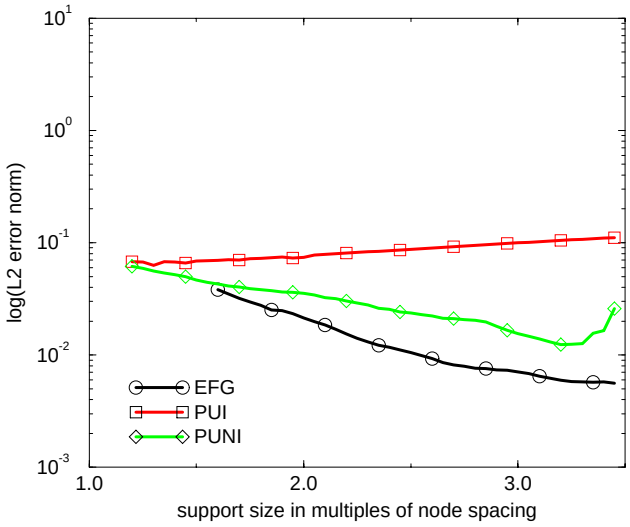


(b) Irregular grid

Figure 6: Interpolation errors $||f - f_h||_{L_2}$ for varying support size. Grid 10×10 nodes. Quartic circular weight. All the nodes inside the domain of influence of the center node are selected as the star nodes.

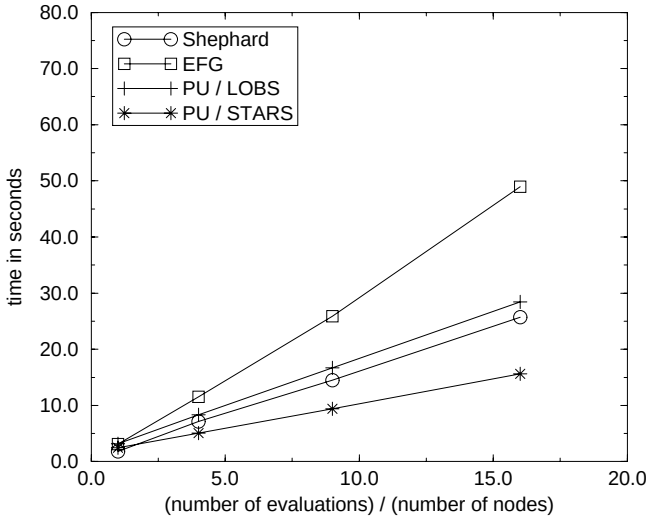


(a) Regular grid

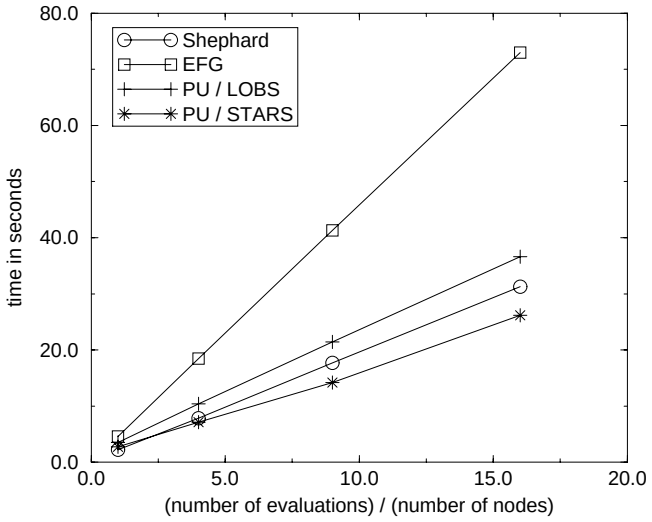


(b) Irregular grid

Figure 7: Interpolation errors $||f - f_h||_{L_2}$ for varying support size. Grid 10×10 nodes. Quartic circular weight. Only the closest four nodes are selected as the star nodes.



(a) 12 neighbors



(b) 32 neighbors

Figure 8: Timings for different average number of neighbors per evaluation point. Methods shown: `Shepard` = $\omega_I(\boldsymbol{x})$ basis (Shepard method), `EFG` = EFG method, `PU/LOBS` = present approach with bounding-box tree search structure, `PU/STARS` = present approach with star-based searching.