

A computational tool for comparing all linear PDE solvers

Error-optimal methods are meshless

Robert Schaback

Received: date / Accepted: date

Abstract The paper provides a computational technique that allows to compare all linear methods for PDE solving that use the same input data. This is done by writing them as linear recovery formulas for solution values as linear combinations of the input data, and these formulas are continuous linear functionals on Sobolev spaces. Calculating the norm of these functionals on a fixed Sobolev space will then serve as a quality criterion that allows a fair comparison of all linear methods with the same inputs, including standard, extended or generalized finite–element methods, finite–difference– and meshless local Petrov–Galerkin techniques. The error bound is computable and yields a sharp worst–case bound in the form of a percentage of the Sobolev norm of the true solution. In this sense, the paper replaces proven error bounds by calculated error bounds. A number of illustrative examples is provided. As a byproduct, it turns out that a unique error–optimal method exists. It necessarily outperforms any other competing technique using the same data, e.g. those just mentioned, and it is necessarily meshless, if solutions are written “entirely in terms of nodes” (Belytschko et. al. 1996 [6]). On closer inspection, it turns out that it coincides with *symmetric meshless collocation* carried out with the kernel of the Hilbert space used for error evaluation, e.g. with the kernel of the Sobolev space used. This technique is around since at least 1998, but its optimality properties went unnoticed, so far. Examples compare the optimal method with several others listed above.

Keywords Sobolev spaces · Finite Elements · Meshless methods · Finite Differences · Error Bounds · Radial Basis Functions · Kernels · Partial Differential Equations · Discretization

Mathematics Subject Classification (2000) 65N06 · 65N12 · 65N15 · 65N22 · 65N30 · 65N35 · 65N38 · 65D25

R. Schaback
Institut für Numerische und Angewandte Mathematik
Universität Göttingen
Lotzestrasse 16–18
D-37083 Göttingen
Germany
Tel.: ++49-+551-394501
Fax: ++49-+551-393944
E-mail: schaback@math.uni-goettingen.de

1 Introduction

For simplicity, consider a standard problem

$$\begin{aligned} Lu &= f \text{ in } \Omega \\ u &= g \text{ in } \Gamma := \partial\Omega \end{aligned} \tag{1}$$

with a linear elliptic differential operator L and Dirichlet boundary values, posed on a domain Ω . It will be clear in section 3 how this generalizes to other linear differential operators and other boundary conditions that are linear in u . Assume that the amount of available information is fixed and limited, namely to values

$$\begin{aligned} f(x_j) &\text{ in points } x_j \in \overline{\Omega}, \quad 1 \leq j \leq m, \\ g(y_k) &\text{ in points } y_k \in \Gamma := \partial\Omega, \quad 1 \leq k \leq n. \end{aligned} \tag{2}$$

Under all methods for solving such problems, we ask for the one with smallest error that uses this information and not more. To make this more precise, we fix a single point $x \in \overline{\Omega}$ and assume existence of a true solution u^* of the problem. Any method \mathcal{M} using the above information will produce some numerical solution $\tilde{u}_{\mathcal{M}}$, and we want to single out methods that make the error $|u^*(x) - \tilde{u}_{\mathcal{M}}(x)|$ small for all problems posed that way. We achieve this by looking at error bounds

$$|u^*(x) - \tilde{u}_{\mathcal{M}}(x)| \leq C_{\mathcal{M}} \|u^*\|_H \text{ for all } u^* \in H, \tag{3}$$

where the problems and their solutions are allowed to vary in such a way that the true solutions lie in some fixed reproducing kernel Hilbert space H of functions on Ω , e.g. a fixed Sobolev space. Note that the literature on partial differential equations provides theorems about existence and uniqueness for scales of Sobolev or Hölder spaces, see e.g. [18, 15].

In the sense of (3), the constant $C_{\mathcal{M}}$ describes the worst-case error behavior of the method \mathcal{M} on all problems with solutions in H using the same data, and the error is given as a percentage of $\|u^*\|_H$, which is the only unknown quantity here. We shall show how to evaluate $C_{\mathcal{M}}$ numerically, and this will allow fair comparisons between methods using the same data. A few examples including important methods like finite elements, generalized finite differences, and meshless collocation in various forms are provided at the end. Note that this paper does not define a new solver. It compares existing solvers, and it does not compete for efficiency. The comparison will not be cheaper than the solving.

But one can also ask for a method \mathcal{M} that makes $C_{\mathcal{M}}$ minimal over all linear methods using the same data. This problem is rewritten as one of optimal recovery of functions, and it is proven that the optimal solution exists uniquely. It can be calculated explicitly, takes the form of a recovery method in reproducing kernel Hilbert spaces, and is a *meshless method*. Since it is optimal in the above sense, it outperforms errorwise any other competing technique using the same data, may it use finite elements, finite differences, or local Petrov–Galerkin techniques. Upon closer inspection, it turns out to be nothing new, because it is a special case of *symmetric meshless collocation*, computed using the kernel that was used for error assessment. This method relies on Hermite–Birkhoff interpolation in reproducing kernel Hilbert spaces [31] and its application to PDE solving was first analyzed in [12, 13].

2 Recovery Problems

To keep the presentation simple, we stay with the problem (1). Any linear method that provides an approximate solution value $\tilde{u}(x)$ for a fixed point x and uses exclusively the data (2), must necessarily satisfy a formula of the form

$$\tilde{u}(x) = \sum_{j=1}^m r_j(x) f(x_j) + \sum_{k=1}^n s_k(x) g(y_k), \quad (4)$$

whatever the weights $r_j(x)$ and $s_k(x)$ are. We call this a *direct linear recovery* of the value $\tilde{u}(x)$ from the data on the right-hand side. The existence of (4) follows from linearity and the restriction to the admitted data, but in general it will need some effort to rewrite a classical method in this form. We come back to this in section 5 on specific methods.

Clearly, the error in (4) is

$$u^*(x) - \tilde{u}(x) = u^*(x) - \sum_{j=1}^m r_j(x) Lu^*(x_j) - \sum_{k=1}^n s_k(x) u^*(y_k),$$

where we now have connected the data functions f and g back to the true solution u^* . The map

$$\varepsilon_{x,r,s} : u^* \mapsto u^*(x) - \sum_{j=1}^m r_j(x) Lu^*(x_j) - \sum_{k=1}^n s_k(x) u^*(y_k) \quad (5)$$

is a linear functional in u^* , and if it is bounded on some Hilbert space H , we have an error bound

$$|u^*(x) - \tilde{u}(x)| \leq \|\varepsilon_{x,r,s}\|_{H^*} \|u^*\|_H \text{ for all } u^* \in H. \quad (6)$$

If the norm $\|\varepsilon_{x,r,s}\|_{H^*}$ is evaluated, it precisely describes the worst-case error behavior for all problems with solutions in H , because the inequality is sharp by definition of the norm of the functional. In other words, one can express the error explicitly in percent of $\|u^*\|_H$, and this is what we shall do in the rest of the paper. Note that error bounds of the form (6) need no special theoretical proof, just the calculation of the actual value of $\|\varepsilon_{x,r,s}\|_{H^*}$ by the computer. In that sense, finding error bounds of that form is not a matter of proof anymore, but a matter of explicit computation. The usual theoretical arguments like *consistency and stability imply convergence* are not relevant for this a-posteriori error evaluation, because the calculated value of the norm $\|\varepsilon_{x,r,s}\|_{H^*}$ precisely describes the actual error behavior, and it will be large if the design of the PDE solving method behind the recovery formula has flaws like instability or bad consistency, in whatever sense.

Furthermore, one can ask for a choice $\mathbf{r}^*(x)$ and $\mathbf{s}^*(x)$ of the coefficient vectors that make the norm minimal, and this will later lead to an optimal method for the given reconstruction problem.

Note that we do not consider the way numerical solutions within the considered methods are actually calculated. They are usually not obtained via a recovery formula (4), though the latter is hidden behind the scene. The amount of numerical integration needed for weak methods, the error committed in integration subroutines, and any multigrid solver techniques are contained implicitly in the discrete recovery formula, but not appearing explicitly. Different numerical strategies within the same class of methods, e.g. finite element solvers with

different ways of integration or with different element families will lead to variations of recovery formulas that can be compared directly. The only requirement is that the method is rewritten in direct linear recovery form (4), and then the total error is evaluated, containing all method-specific internal features of the technique in question.

3 Error Evaluation

Fortunately, it is easy to evaluate such norms in reproducing kernel Hilbert spaces, and Sobolev spaces are examples. In such spaces, the inner product and the kernel $K : \Omega \rightarrow \Omega$ have the properties

$$\begin{aligned} f(x) &= (f, K(x, \cdot))_H \text{ for all } f \in H, x \in \Omega \\ K(x, y) &= (K(x, \cdot), K(y, \cdot))_H \text{ for all } x, y \in \Omega \end{aligned}$$

and for all linear and continuous functionals $\lambda, \mu \in H^*$,

$$\begin{aligned} (\lambda, \mu)_{H^*} &= (\lambda^x K(x, \cdot), \mu^y K(y, \cdot))_H \\ &= \lambda^x \mu^y K(x, y), \end{aligned} \tag{7}$$

where the upper index at the functionals denotes the variable that is acted upon. Details are in the literature on kernels, with [30] being a fairly complete reference and [25, 26] being open access compilations for teaching purposes. A good source for MATLAB programs within kernel methods and their background is [11]. A somewhat more theoretical book is [8].

Global Sobolev spaces $W_2^m(\mathbf{R}^d)$ in d dimensions and for order $m > d/2$ in the standard Fourier transform definition have the radial *Matérn* reproducing kernel

$$K(x, y) = \frac{2^{1-m}}{\Gamma(m)} \|x - y\|_2^{m-d/2} K_{m-d/2}(\|x - y\|_2), x, y \in \mathbf{R}^d \tag{8}$$

with the modified Bessel function $K_{m-d/2}$ of the second kind. Local Sobolev spaces $W_2^m(\Omega)$ for domains $\Omega \subset \mathbf{R}^d$ are norm-equivalent to the global spaces, as long as domains are non-pathological, i.e. they satisfy a Whitney extension property or a have a piecewise smooth boundary with a uniform interior cone condition. We shall use the above kernels for evaluation of errors in Sobolev space.

To avoid double sums and to pave the way for generalizations, we shall calculate the error norms after rewriting (5) in terms of functionals as

$$\varepsilon_{x,c} = \delta_x - \sum_{k=1}^N c_k(x) \lambda_k \tag{9}$$

with

$$\lambda_j(u) = Lu(x_j), 1 \leq j \leq m, \text{ and } \lambda_{m+i}(u) = u(y_i), 1 \leq i \leq n, N = m + n \tag{10}$$

in our special case. It should be clear that other differential operators and other boundary conditions will just change the functionals here.

Then (7) implies that we can calculate the quadratic form

$$\begin{aligned} \|\varepsilon_{x,c}\|_{H^*}^2 &= K(x,x) - 2 \sum_{i=1}^N c_i(x) \lambda_i^z K(x,z) \\ &\quad + \sum_{j,i=1}^N c_i(x) c_j(x) \lambda_i^y \lambda_j^z K(y,z) \end{aligned} \quad (11)$$

explicitly, if the kernel is known and the functionals are continuous, in particular on Sobolev spaces of sufficient regularity. For instance, if L is a second-order elliptic operator, this poses the restriction $m - 2 > d/2$ on the Sobolev order m we can use. But this is no surprise, since we focus on methods that use $f = Lu^*$ pointwise, forcing f to be in W_2^k with $k > d/2$ and thus $u^* \in W_2^m$ with $m > 2 + d/2$, by the Sobolev embedding theorem. All methods that use these data are implicitly making this smoothness assumption, even if their users think that they are working in in less regular spaces. Plenty of papers and books miss this point. In particular, the standard FEM method is usually formulated and analyzed in low-regularity spaces like H^1 or H^2 , but when it comes to using data like (2), it needs H^m with $m > 3$ in \mathbf{R}^2 to let the data functionals be continuous. Since this argument seems to ignore that the FEM can perfectly handle problems stated in H^1 or H^2 , we explain the situation in more detail in section 7. Our numerical comparison technique works also in the limit $m = 3$ for dimension 2, as the examples in section 6 show.

All methods based on the data $\lambda_1(u^*), \dots, \lambda_N(u^*)$ and brought into the recovery form

$$\tilde{u}(x) = \sum_{i=1}^N c_i(x) \lambda_i(u^*)$$

generalizing (2) and (5) can now be plugged into (11) to show how good the reproduction quality at x is, because (6) generalizes accordingly. Note that there is no linear system to be solved once the recovery formula is known. But since the formula is inserted into the positive definite quadratic form (11), a small final value will necessarily contain quite some amount of numerical cancellation. The computational complexity for norm evaluation is $\mathcal{O}(N^2)$ for a total of N data.

This allows a fair comparison of all such methods using the same data, and we shall provide examples below. The comparison can be made pointwise, as we saw, but for small problems one can plot the function $x \mapsto \|\varepsilon_{x,c}\|_{H^*}^2$ to see where a method works badly and needs more data. This gives direct information for refinement of the discretization, for all kinds of methods.

4 Optimal Methods

Since (11) is a quadratic form with a positive semidefinite matrix, it can be minimized. The optimal solution coefficients $c_i^*(x)$ solve the system

$$\sum_{k=1}^N c_k^*(x) \lambda_j^y \lambda_k^z K(y,z) = \lambda_j^z K(x,z), \quad 1 \leq j \leq N, \quad (12)$$

and the system can be proven to be solvable [31]. Clearly, this choice of coefficients in the generalized discrete recovery formula (9) will then outperform all other competitors error-wise. We shall show examples later. The minimal value of (11) then is

$$\|\varepsilon_{x,c^*}\|_{H^*}^2 = K(x,x) - \sum_{i=1}^N c_i(x)^* \lambda_i^z K(x,z) \geq 0$$

and does not contain any matrix.

The system (12) reveals the nature of this method. Indeed, the functions c_k are necessarily linear combinations of the functions $\lambda_i^z K(\cdot, z)$, and application of λ_i^x to the above system shows the Lagrange property $\lambda_i(c_k) = \delta_{ik}$, $1 \leq i, k \leq N$. This means that the c_k are the Lagrange basis for general Hermite–Birkhoff interpolation of the given data by the functions $\lambda_i^z K(\cdot, z)$, and this is the well-known method of *symmetric meshless collocation* based on [31] and analyzed thoroughly in [12, 13]. The optimality of the technique in the sense of this paper should have been known at least since 1997, but it went unnoticed because [25, p.82, (4.2.2)] was not applied to PDE solving at that time. Note finally that the optimal method inherits any proof of any convergence rate for any method whatsoever using the same data.

5 Special Methods

We now provide some details on how to evaluate recovery errors for special PDE solution techniques. This will be useful for the examples in the final section, and we specialize to $L = -\Delta$ here, because generalizations are easy to work out.

5.1 Finite Elements

The simplest possible 2D finite–element code uses piecewise linear elements on the triangles of a triangulation of a domain with piecewise linear boundary, and requires f –values only at the barycenters of the triangles, if numerical integration is performed by the midpoint rule. These are the points x_j in (2) in the FEM version that we denote by FEMBary below. But since usually there are more triangles than vertices, one can also prescribe f –values at all interior and boundary vertices to calculate approximate values at the barycenters. This usually needs less f –values at the same order of accuracy, but the data points are closer to the evaluation points, which usually are the vertices of the triangulation that are not carrying Dirichlet data. We call this method FEMNode below. These two FEM variations have different recovery formulas (4) and different error functionals (5) to be compared, because their integration strategy is different.

To get the recovery formulas in the form needed here, users will have to check carefully what their FEM code does. We used the MATLAB `pdetool` setting, which does the following. It uses all N triangle vertices z_1, \dots, z_N for setting up the standard piecewise linear test and trial functions v_1, \dots, v_N with $v_j(z_k) = \delta_{jk}$ and builds the $N \times N$ stiffness matrix with entries $(\nabla v_i, \nabla v_j)_{L_2}$ in the usual way. The right–hand sides $(f, v_j)_{L_2}$, $1 \leq j \leq N$ are certain linear combinations of f –values either at triangle barycenters or at vertices of the neighboring triangles. These linear combinations form a sparse *integration matrix* \mathbf{B} with entries b_{jk}

such that the stiffness system without boundary conditions but with some form of numerical integration is

$$\sum_{i=1}^N (\nabla v_i, \nabla v_j)_{L_2} u(z_i) = \sum_{k=1}^N b_{jk} f(x_k), \quad 1 \leq j \leq N.$$

The FEMBary and FEMNode variations have different \mathbf{B} matrices and use f at different points x_k , but the stiffness matrices are the same. The study of errors induced by numerical integration [4] can be studied by using various integration matrices, for all methods using a weak approach to the PDE.

If we single out the set D of indices of the boundary vertices, the unknowns $u(z_i)$, $i \in D \subset \{1, \dots, N\}$ are known Dirichlet values, and thus

$$\sum_{i \notin D} (\nabla v_i, \nabla v_j)_{L_2} u(z_i) = \sum_{k=1}^N b_{jk} f(x_k) - \sum_{i \in D} (\nabla v_i, \nabla v_j)_{L_2} u(z_i), \quad j \notin D$$

is the system to be actually solved. Since the $u(z_i)$ with $i \in D$ are g -values, the system has the matrix–vector form

$$\mathbf{A}\mathbf{u} = \mathbf{B}\mathbf{f} + \mathbf{C}\mathbf{g} \quad (13)$$

under adequate notation, and each row of

$$\mathbf{u} = \mathbf{A}^{-1}\mathbf{B}\mathbf{f} + \mathbf{A}^{-1}\mathbf{C}\mathbf{g}$$

provides one instance of (4) for each of the points $x = x_i$, $i \notin D$. If we focus on the origin as one of these points, as we do later in our examples, we need just one row. Note that the sparsity of the stiffness matrix \mathbf{A} gets lost after the transition to a recovery formula.

If users want the error at a non–nodal point x , they have to add a piecewise linear interpolation on a triangle containing x , and then the discrete reconstruction formula is a linear combination of three rows of the above system. This illustrates how numerical integration and interpolatory post–processing both enter explicitly into our version of a complete and explicit error analysis. We applied this additional linear interpolation when preparing Figures 6 and 7, where we needed the error norms for evaluation on midpoints of edges.

5.2 Symmetric Kernel–Based Collocation

This method works on the two point sets X and Y from (2) and uses linear combinations

$$u(x) = \sum_{j=1}^m c_j \Delta K(x, x_j) + \sum_{k=1}^n d_k K(x, y_k) \quad (14)$$

of basis functions derived from a smooth kernel K . The argument in Section 4 shows that this yields optimal errors in the Hilbert spaces in which their kernels are reproducing. In case of Sobolev spaces, we thus get the error–optimal methods in Sobolev spaces this way.

But, again, the numerical process for solving the PDE is different from a recovery formula (4). The standard algorithm collocates these trial functions at Dirichlet and PDE vertices, forming the block system

$$\begin{aligned} \sum_{j=1}^m c_j \Delta^x \Delta^y K(x_i, x_j) + \sum_{k=1}^n d_k \Delta K(x_i, y_k) &= f(x_i), \quad 1 \leq i \leq m, \\ \sum_{j=1}^m c_j \Delta K(y_i, x_j) + \sum_{k=1}^n d_k K(y_i, y_k) &= g(y_i), \quad 1 \leq i \leq n. \end{aligned}$$

The inverse of the coefficient matrix recovers the coefficients c_j and d_k from the f and g data, and the numerical solution at x is just a linear combination (14) of those coefficients. Consequently, the discrete recovery (4) is furnished by the inverse of the above “stiffness” matrix, premultiplied by the row vector of the kernel values in (14).

Of course, one can use a special kernel K to calculate the discrete recovery, and then use another kernel, e.g. one generating a Sobolev space, for error evaluation on that space. We shall do this in the final section, for the methods called HOBary and HONode there, HO standing for “high order”.

5.3 Unsymmetric Kernel-Based Collocation

In contrast to the previous section, this class of methods takes an additional set $Z = \{z_1, \dots, z_N\}$ of usually $N = m + n$ points and works on linear combinations

$$u(x) = \sum_{k=1}^N d_k K(x, z_k) \quad (15)$$

of basis functions, while still using collocation in the point sets X and Y from (2). This method dates back to early papers of Ed Kansa [16, 17] and was called MLSQ2 as a variation of the Meshless Local Petrov–Galerkin method [1, 2] of S.N. Atluri and collaborators. The linear system for the coefficients now is

$$\begin{aligned} \sum_{k=1}^n d_k \Delta K(x_i, z_k) &= f(x_i), \quad 1 \leq i \leq m, \\ \sum_{k=1}^n d_k K(y_i, z_k) &= g(y_i), \quad 1 \leq i \leq n, \end{aligned}$$

and the inverse of the coefficient matrix (if it exists, see [14] for a counterexample), premultiplied by the vector of values $K(x, z_k)$ of (15) will yield a row vector for the discrete recovery formula (4). If N is chosen larger than $m + n$ to increase stability, a pseudoinverse of the system coefficient matrix can replace the inverse. This was done in the examples of the final section, for the methods called KansaBary and KansaNode there.

5.4 Meshless Lagrange Methods

Here, a set $Z = \{z_1, \dots, z_N\}$ of trial nodes is chosen, and there are *shape functions* u_1, \dots, u_N such that *trial functions*

$$u(x) = \sum_{k=1}^N u_k(x) u(z_k)$$

can be written “entirely in terms of nodes” [6]. Usually, this implies Lagrange conditions $u_j(z_k) = \delta_{jk}$, and in many cases the shape functions are defined via Moving Least Squares. We do not care here for details, and allow such techniques to come in weak or strong form. The strong case collocates in sets X and Y like above, forming a system

$$\begin{aligned} \sum_{k=1}^N \Delta u_k(x_j) u(z_k) &= f(x_j), \quad 1 \leq j \leq m, \\ \sum_{k=1}^N u_k(y_i) u(z_k) &= g(y_i), \quad 1 \leq i \leq n \end{aligned}$$

and the weights of the discrete recovery at z_k will be a row of the pseudoinverse of the coefficient matrix of this system.

The weak cases form stiffness matrices and right-hand sides like in the FEM situation, and then we get the coefficients of the discrete recovery in the same way, involving a special matrix \mathbf{B} caring for the numerical integration.

We have omitted examples for these methods and postpone them to a follow-up paper. It might also include the effect of numerical integration [3] on such techniques.

5.5 Generalized Finite-Difference Methods

Here, there are no trial functions, but everything is still expressed in terms of values at nodes $Z = \{z_1, \dots, z_N\}$. In the strong situation, PDE operator values are approximated by formulas like

$$\Delta u(x_j) \approx \sum_{i=1}^N \alpha_{jk} u(z_k), \quad (16)$$

with localized weights α_{jk} , and this can be done with minimal error in a reproducing kernel Hilbert space using the logic of section 4. See [9,27] for more details. The linear system then is

$$\begin{aligned} f(x_j) &= \sum_{i=1}^N \alpha_{jk} u(z_k), \\ f(x_j) - \sum_{i \in D} \alpha_{jk} u(z_k) &= \sum_{i \notin D} \alpha_{jk} u(z_k), \end{aligned}$$

if we use a subset D of Dirichlet nodes like in the FEM case. This is of the form (13) and we already know how to derive the recovery formulas in such a case. It is interesting to see that the matrix with coefficients α_{jk} plays the role of a stiffness matrix here, and it is the place where sparsity can be implemented to yield local methods with sparse matrices. This was very successfully done in various application papers of C.S. Chen, B. Sarler, and G.M. Yao [23,33,32,34,29]. We shall present a simple numerical example in the final section.

The Direct Meshless Local Petrov Galerkin methods of [19,20] are weak cases of this approach, with other functionals than (16) being directly approximated in terms of values at nodes, and with integrations involving \mathbf{B} matrices again, like in all other weak methods. We leave this for a future paper dealing with all variants of Atluri’s Meshless Local Petrov Galerkin technique, and comparing them to FEM and kernel methods, optimal or not, sparse or not.

5.6 General Methods

If a linear PDE solver of unknown type and unknown source code is at hand as an executable program, one can retrieve its hidden recovery formula by sufficiently many runs. In the context of (4), this needs $m + n$ runs of the black–box algorithm on Kronecker data, i.e. setting all input data to zero except for one datum set to 1, and do this for all $m + n$ input data in turn. If the evaluation point x is kept fixed, the results of the runs will yield the values $r_j(x)$, $1 \leq j \leq m$ and $s_k(x)$, $1 \leq k \leq n$. Then the black–box algorithm can be compared error–wise to others. Though computationally inefficient, this method may be used to test doubtful solvers provided by dubious sources, even if the code is concealed.

6 Numerical Examples

To avoid overloading this expository paper, we present a simple series of examples. They all work on the unit disk for simplicity, and in order to include finite elements, we have to use triangulations, even if they are not needed for meshless methods. We start with the standard discretization of the unit disk roughly into 8 triangles meeting at the origin, followed by three standard finite–element refinement steps halving the edges. The problem (1) is posed with $L = -\Delta$, and Dirichlet boundary values are always provided in the boundary vertices, which are the y_k in (2). In all cases, the non–boundary vertices of the triangles are the vertices where we want to know the solution, but in the sense of (5) and for simplicity we only evaluate the recovery error at the origin $x = 0$.

As the FEM variations show, one can work with f values either in barycenters of triangles or in vertices of the triangulation. We shall evaluate all examples in both situations, denoting the methods by either *Bary or *Node. Details on the unit disk discretizations we call C0 to C4 are in Table 1:

- n : number of Dirichlet boundary data points for g values,
- m_{Bary} : number of triangles and barycentric data points for f values,
- m_{Node} : number of vertices and vertex data points for f values, including the n Dirichlet boundary vertices,
- DOF: degrees of freedom = number of unknowns = $m_{Node} - n$,
- h : *fill distance* in the sense of kernel discretizations, describing the maximal distance of an arbitrary point of the domain to one of the vertices.

Case	n	m_{Bary}	m_{Node}	DOF	h
C0	8	8	9	1	0.2706
C1	16	32	25	9	0.1515
C2	32	128	81	49	0.0768
C3	64	512	289	225	0.0389
C4	128	2048	1089	961	0.0197

Table 1 Discretization data for the examples on the unit disk

The plots in Figures 1 to 5 show the errors of various methods in Sobolev space of order 3 to 7, with order 3 being somewhat out of the theoretical bounds mentioned in section 3, and

the data for the fine discretization C4 being polluted by ill-conditioning in various cases. In each figure, the Sobolev space for error evaluation is fixed, and thus also the kernel in that Sobolev space, called the “evaluation kernel” in what follows. But kernel-based meshless methods use their own “construction kernels” to implement their solution procedure. These are independent from the “evaluation kernels” tied to the Sobolev spaces in which the comparison takes place. However, the optimal methods in the sense of section 4 must use the evaluation kernel as their construction kernel.

The methods are

- FEMBary: piecewise linear FEM with f data in barycenters
- FEMNode: same FEM with f data in nodes
- KansaBary: Unsymmetric collocation with f data in barycenters, using the order 7 Sobolev kernel at scale 0.5 for construction
- KansaNode: same with node data, at scale 1
- HOBary: symmetric high-order collocation with f data in barycenters, using the order 7 Sobolev construction kernel at scale 1
- HONode: same with data in vertices. These two coincide with the optimal methods, if evaluated on Sobolev space of order 7, see Figure 5, because then the construction and evaluation kernels coincide
- OptBary: optimal method in the Sobolev space used for error evaluation, with f data in barycenters
- OptNode: same with data in nodes
- LocNode: a bandwidth 15 method like in section 5.5, constructed using the order 7 Sobolev kernel, data in nodes, with scale 1.

There are serious instabilities in the figures that need explanation. They could have been avoided in all cases by choosing a smaller kernel scale for the construction kernels, usually at the expense of a larger error norm. The evaluation kernels cannot be scaled without changing the Sobolev norm, and thus the kernel scale for Sobolev space error evaluation was fixed at 1.0 throughout. This does not seriously affect the error evaluation, because the latter just consists of a calculation of a quadratic form.

But instability affects the calculation of recovery formulas by kernel methods, including the optimal ones. In particular, increasing the smoothness of the construction kernel will increase the instability, if no precautions like preconditioning [5, 7] are taken. We chose the KansaNode and the HO* kernel methods to work with the kernel of Sobolev space of order 7 at scale 1, and the scale of the kernel for KansaBary to be 0.5, but smaller scales would have been more stable. Users can easily try different kernels and scales for construction and then evaluate in a fixed Sobolev space at a fixed scale to see which construction scale gives best results.

The numerical results support quantitatively what is known from experience and partly supported by theory. The piecewise linear FEM technique is adapted to low-regularity problems, and it shows its second-order convergence in all applicable Sobolev spaces from order 4 on. Of course, it is somewhat unfair to use only a h -type FEM with linear elements here, but since the p -FEM is hard to implement, we have to leave comparisons to the p -FEM to a follow-up paper, which should also deal with the MLPG technique for higher p .

Due to being limited to piecewise linear elements, the FEM is clearly inferior when evaluated in all higher-order Sobolev spaces, but it is surprisingly close to the optimum when

evaluated in H^4 . The error-optimal method is symmetric collocation, and the difference to the FEM must get larger for increasing Sobolev order, because the optimal method increases its convergence rate automatically with the Sobolev order. This effect is well-known and proven in the interpolation case, and a proof for the PDE case is not necessary, because the method outperforms any p -FEM technique error-wise. Unfortunately, the optimal method suffers from severe ill-conditioning if no precautions are taken, and it does not allow sparsity. However, it should serve as a standard reference to evaluate error performances of all other methods using the same data.

We now list some observations concerning comparisons. The optimal method HO* for a fixed high Sobolev order performs well also for lower Sobolev order. It adapts automatically to lower regularity. Again, this effect is well-known in the approximation and interpolation case [24, 21]. Other competing methods, like Kansa* for unsymmetric collocation, behave in the same way. Our results for the Kansa method can very probably be improved by playing around with other construction kernels and scales, but we wanted to stick to Sobolev kernels at scale 1 as far as possible. If sparsity is enforced by localizing meshless collocation [23, 33, 32, 34, 29], one gets a competitive method LocNode that shares sparsity with FEM techniques, but also yields high convergence orders depending on the bandwidth chosen. Since the bandwidth limits the attainable order in such cases, one can sacrifice order-adaptivity and use a variable scale for construction, in this case $10h$ for fill distance h . At this point, it should be noted that meshless methods depend on various parameters, e.g. the kernel choice, the scaling, and the point distribution. Follow-up papers should use the comparison tool of this paper to investigate the effect of changes in the parameters in much more detail.

We now leave the situation where we compare methods with respect to their error at a single point. As was mentioned at the end of section 3, one can plot the error norm as a function. Figures 6 and 7 show the error on Sobolev space of order 4 for the FEM working on the C2 discretization, and the error evaluated on the C3 discretization of Table 1. Such functions can be used for deciding about refinement, and they provide strict error bounds rather than error estimators.

The optimal method on Sobolev spaces can be run in a *greedy* way. If the method is considered on a fixed data set, one can evaluate the error norm on a fine discrete point set and select a new data point on the boundary or the interior where the error norm is maximal. This was started with just one point on the boundary and the interior, and run up to a total of 120 data points, with results in Figures 8 and 9 for Sobolev order 4. The plot of the error decrease in Figure 9 as a function of the total number n of data was done in double logarithmic scale, but as a function of \sqrt{n} to show a fixed convergence rate for a 2D problem. This is a generalization of the method described in [10] to PDE data. The comparison method of this paper opens the quest for finding optimal data locations for PDE solving. While the above technique works independent of the data and provides good discretizations for all problems in the same setting, there also is a data-dependent greedy technique [28] designed for interpolation. It should be extended to PDE solving by a forthcoming paper.

MATLAB programs for generating the examples can be downloaded from the author's website.

7 Extension to weak data

One of the referees correctly pointed out that Dirichlet problems on domains in 2D with incoming corners do not seem to fit into the framework of this paper, since the regularity of the solution is of order smaller than 3. In particular, on the L-shaped domain $\Omega = (-1/2, 1/2) \times (-1/2, 1, 2) \setminus [0, 1/2) \times [0, 1/2)$, the solution to a Laplace problem is given by $u(r, \phi) = r^{3/2} \sin((2\phi - \pi)/3)$ in polar coordinates. This function is clearly continuous on the boundary of Ω , the right-hand side of $-\Delta u = 0$ is clearly a C^∞ function, but the solution belongs only to $H^s(\Omega)$ with $s < 1 + 3/2$. We use this example as a starting point for dealing with problems working on weak input data.

The comparison tool of this paper relies on the fact that the admitted solvers work on data f_j that are interpreted as $f_j = \lambda_j(u) := -\Delta u(x_j)$ with the functionals λ_j being continuous on the Sobolev space used for comparison. This is clearly not satisfied if $H^s(\Omega)$ with $s < 1 + 3/2$ is chosen, and consequently all methods using these “strong” data in this way have recovery formulas that are unbounded on that space, no matter how they are obtained by the linear algebra formulas of the method’s definition, and including all linear FEM codes that work on these data, e.g. via numerical integration. Note that this is a statement about linear PDE solver codes based on data (2) and yielding a recovery formula of the type (4), not a statement about a single example.

But, on the contrary, it is well-known that the FEM can handle the example, and problems of low regularity in general. This seems to be a contradiction, but it isn’t, and it can be resolved by looking at what the FEM does in case of low regularity, and by properly defining what the input data are and how the error is measured.

The FEM handles problems of low regularity by going over to a weak formulation. The input data, at least in theory, are not pointwise values of f , but inner products $(f, v_j)_2$ of f against (not necessarily piecewise linear or continuous) *test functions* $v_j \in H^1(\Omega)$, and they have the semantics $(f, v_j)_2 = (\nabla u, \nabla v_j)_{2, \text{supp}(v_j)}$, not the semantics of $(f, v_j)_2 = (-\Delta u, v_j)_2$ when played back to the solution u . Thus the FEM works with data $f_j := \lambda_j(u) = (f, v_j)_2$ and with functionals $\lambda_j(u) := (\nabla u, \nabla v_j)_{2, \text{supp}(v_j)}$ that are continuous on $H^1(\Omega)$. If defined in that form, the FEM belongs to a class of methods that is different from the one considered so far in this paper, because it uses different input data and yields different recovery formulas.

We shall now have a closer look at all linear PDE solvers based on weak data $\lambda_j(u) := (\nabla u, \nabla v_j)_{2, \text{supp}(v_j)}$ plus boundary data. The theory of this paper in its general form based on (9) applies to all cases in a Hilbert space, as long as the functionals are continuous. The above functionals are continuous on $H^1(\Omega)$. If the data on the boundary are given pointwise, this restricts us to H^s with $s > 1$ by the trace theorem in 2D, and this will cover the example. The kernel on H^s in 2D is $r^{s-1}K_{s-1}(r)$, and the optimal method will again be symmetric collocation using that kernel, now using the new functionals which in turn depend on pre-selected test functions v_j . In particular, the trial functions of the optimal method will partly be

$$w_j(y) = \lambda_j^x K(x, y) = (\nabla^x (\|x - y\|_2^{s-1} K_{s-1}(\|x - y\|_2)), \nabla v_j)_2.$$

Summarizing, this means that the FEM at H^s regularity with $s > 1$ can be treated with the methods of this paper, but with different data functionals.

The lowest possible regularity, however, arises when the FEM works in $H_0^1(\Omega)$ with zero boundary conditions. Then the FEM calculates values of an approximation \tilde{u} to the solution at the vertices of the triangulation, and these results are values of continuous functionals on $H_0^1(\Omega)$, using recovery formulas that are given by the rows of the inverse of the stiffness matrix, like in (13), but just using $\mathbf{u} = \mathbf{A}^{-1}\mathbf{f}$, where now the \mathbf{f} vector contains the weak data

$$\lambda_j(u) = (f, v_j)_2 = (\nabla u, \nabla v_j)_{2, \text{supp}(v_j)} =: f_j, \quad 1 \leq j \leq N.$$

The calculation of the values of \tilde{u} at the triangulation vertices thus is a continuous map on the data. But the value $u^*(x_j)$ of the true solution u^* at an inner vertex x_j is undefined for $u^* \in H_0^1(\Omega)$ in 2D. Thus one cannot write down a pointwise error bound, not even on the vertices, though the FEM provides $H_0^1(\Omega)$ -continuous result values there.

All of this is in line with the usual FEM theory. The standard setting is in $H_0^1(\Omega)$, and there the method is optimal because it realizes a Hilbert space projector, but it has only L_2 or H^1 convergence and consequently no pointwise error bound like (6) in 2D. For the latter one has to go to H^s for $s > 1$.

This discussion shows the peculiarities when dealing with low regularity. Any method must work on data supplied by continuous functionals, i.e. the data functionals λ_j must match the low regularity, forcing to pose weak problems instead of strong problems. If the data functionals are continuous, the general theory of this section applies, but only for evaluation functionals δ_x that are continuous as well.

In a similar way problems can be treated that need higher derivatives of the solution, e.g. elasticity problems. If derivative values are to be recovered, these derivative evaluations must be continuous on the Hilbert space used for evaluation, ruling out problems with extremely low regularity.

However, if numerical integration based on pointwise evaluation is used for providing values $f_j \approx (f, v_j)_2$, we are back to the old situation requiring higher regularity of f than the minimum H^{-1} regularity of the standard FEM.

8 Conclusion and Outlook

The paper provides a tool that allows an explicit and fully computational assessment of the error behavior of all linear solvers for all linear PDE problems based on a finite and fixed set of input data. The exact solution can be unknown, and the error is expressed as a factor of the unknown Sobolev norm of the true solution. This tool should be applied in many more circumstances, e.g. on special and awkward domains, for more general differential operators including those of Computational Mechanics, and for many other linear solvers, e.g. generalized or extended finite elements, p -finite element techniques, MLPG methods and boundary-oriented approaches like the DRM [22]. This paper is experimental in the sense that it deals with only a very restricted number of examples so far, but the main result consists of the general comparison tool based on the concept of recovery formulas. Any application-oriented paper can, in principle, apply this technique and thus provide a strict pointwise worst-case error bound in terms of the Sobolev norm of the true solution. Examples of single cases with known solutions can never be completely satisfactory, but they are the usual practice in application papers, unfortunately.

The comparison method itself just inserts a given recovery formula into a quadratic form. This evaluation needs no linear system to be solved, and thus is relatively stable, in particular if compared to solving PDEs with the optimal method. But it needs a reliable evaluation of the coefficients of the recovery problem, which is part of the method to be compared, not of the comparison technique. Its only source of instability is the numerical cancellation in the quadratic form, if the resulting norm is very small.

The comparison method is not intended to replace a solver or to compete for numerical effectivity. It only compares existing solvers error-wise, at an expense that is not smaller than the computational expense of the solvers themselves. Furthermore, the pointwise form of the error comparison is rather a feature than a bug. Users can and should pick an evaluation point or an evaluation functional that can be expected to have a large error that needs control, and this opens the way to localized error control and adaptivity.

Methods based on smooth kernels are adaptive with respect to the regularity of the problem and they perform well error-wise on small and regular problems, but they need preconditioning [5, 7] to enhance stability. They usually do not use sparsity unless when working with compactly supported kernels, but they can be localized, and then they combine sparsity and order-adaptivity.

The unusual way of writing a PDE solver as a recovery formula should be investigated further, in particular towards using it at different scales to mimic multigrid methods.

Finally, this paper shows that there is a method that always realizes the optimal error, but it needs further work to enhance numerical stabilization and computational efficiency. All other methods should be compared to it error-wise, and it is an interesting research challenge to see how close one can come to the optimal method under sparsity and efficiency restrictions.

Acknowledgement

Special thanks go to two reviewers and Dr. Davoud Mirzaei, University of Isfahan, Iran, for several helpful comments.

References

1. S. N. Atluri. *The meshless method (MLPG) for domain and BIE discretizations*. Tech Science Press, Encino, CA, 2005.
2. S. N. Atluri and T.-L. Zhu. A new meshless local Petrov-Galerkin (MLPG) approach in Computational Mechanics. *Computational Mechanics*, 22:117–127, 1998.
3. I. Babuska, U. Banerjee, J.E. Osborn, and Q. Zhang. Effect of numerical integration on meshless methods. *Comput. Methods Appl. Mech. Engrg.*, 198:27–40, 2009.
4. Ivo Babuška, John R. Whiteman, and Theofanis Strouboulis. *Finite elements*. Oxford University Press, Oxford, 2011. An introduction to the method and error estimation.
5. R.K. Beatson, J.B. Cherrie, and C.T. Mouat. Fast fitting of radial basis functions: Methods based on preconditioned GMRES iteration. *Advances in Computational Mathematics*, 11:253–270, 1999.
6. T. Belytschko, Y. Krongauz, D.J. Organ, M. Fleming, and P. Krysl. Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering, special issue*, 139:3–47, 1996.

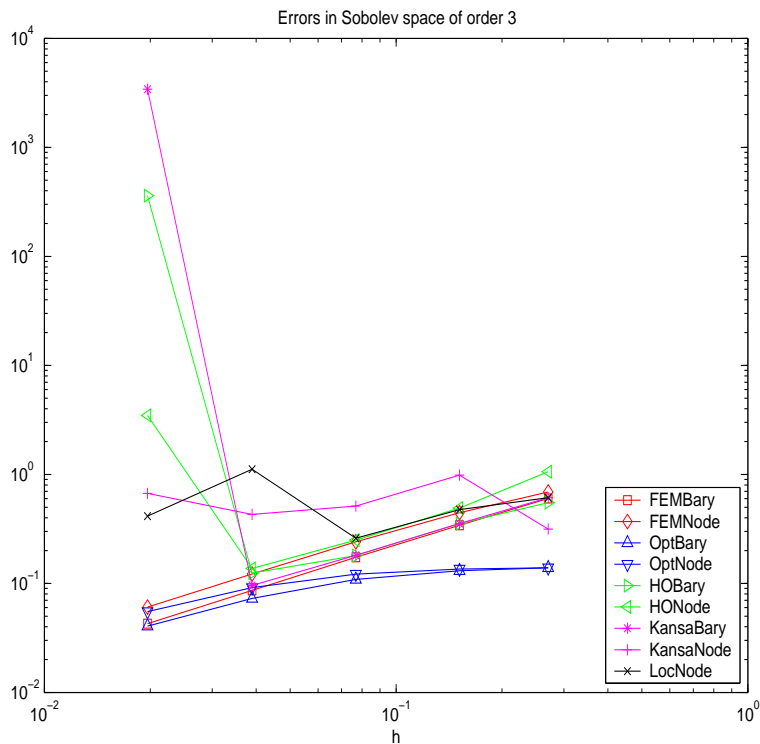


Fig. 1 Errors for Sobolev order 3

7. D. Brown, L. Ling, E.J. Kansa, and J. Levesley. On approximate cardinal preconditioning methods for solving PDEs with radial basis functions. *Engineering Analysis with Boundary Elements*, 19:343–353, 2005.
8. M.D. Buhmann. *Radial Basis Functions, Theory and Implementations*. Cambridge University Press, 2003.
9. O. Davydov and R. Schaback. Error bounds for kernel-based numerical differentiation. Draft, 2012.
10. Stefano De Marchi, R. Schaback, and H. Wendland. Near-optimal data-independent point locations for radial basis function interpolation. *Adv. Comput. Math.*, 23(3):317–330, 2005.
11. G. F. Fasshauer. *Meshfree Approximation Methods with MATLAB*, volume 6 of *Interdisciplinary Mathematical Sciences*. World Scientific Publishers, Singapore, 2007.
12. C. Franke and R. Schaback. Convergence order estimates of meshless collocation methods using radial basis functions. *Advances in Computational Mathematics*, 8:381–399, 1998.
13. C. Franke and R. Schaback. Solving partial differential equations by collocation using radial basis functions. *Appl. Math. Comp.*, 93:73–82, 1998.
14. Y.C. Hon and R. Schaback. On unsymmetric collocation by radial basis functions. *J. Appl. Math. Comp.*, 119:177–186, 2001.
15. Jürgen Jost. *Partial differential equations*, volume 214 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2002. Translated and revised from the 1998 German original by the author.
16. E. J. Kansa. Application of Hardy’s multiquadric interpolation to hydrodynamics. In *Proc. 1986 Simul. Conf., Vol. 4*, pages 111–117, 1986.
17. E. J. Kansa. Multiquadrics - a scattered data approximation scheme with applications to computational fluid-dynamics - I: Surface approximation and partial derivative estimates. *Comput. Math. Appl.*, 19:127–145, 1990.
18. J.L. Lions and Magenes E. *Problèmes aux limites non homogènes at applications vol. 1*. Travaux et recherches mathématiques. Dunod, 1968.

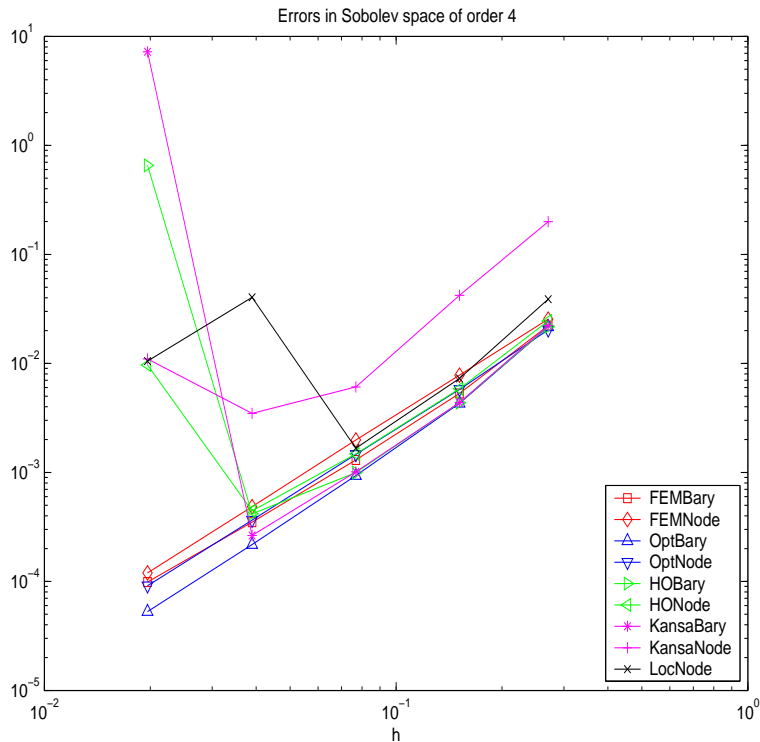


Fig. 2 Errors for Sobolev order 4

19. D. Mirzaei and R. Schaback. Direct Meshless Local Petrov-Galerkin (DMLPG) method: A generalized MLS approximation. *Applied Numerical Mathematics*, <http://dx.doi.org/10.1016/j.apnum.2013.01.002>, 2013.
20. D. Mirzaei, R. Schaback, and M. Dehghan. On generalized moving least squares and diffuse derivatives. *IMA J. Numer. Anal.*, 32, No. 3:983–1000, 2012. doi: 10.1093/imanum/drr030.
21. F. J. Narcowich, J. D. Ward, and H. Wendland. Sobolev bounds on functions with scattered zeros, with applications to radial basis function surface fitting. *Mathematics of Computation*, 74:743–763, 2005.
22. P.W. Partridge, C.A. Brebbia, and L.C. Wrobel. *The Dual Reciprocity Boundary Element Method*. CMP/Elsevier, 1992.
23. Božidar Šarler. From global to local radial basis function collocation method for transport phenomena. In *Advances in meshfree techniques*, volume 5 of *Comput. Methods Appl. Sci.*, pages 257–282. Springer, Dordrecht, 2007.
24. R. Schaback. Approximation by radial basis functions with finitely many centers. *Constructive Approximation*, 12:331–340, 1996.
25. R. Schaback. Reconstruction of multivariate functions from scattered data. Manuscript, available via <http://www.num.math.uni-goettingen.de/schaback/research/group.html>, 1997.
26. R. Schaback. Kernel-based meshless methods. Lecture Note, Göttingen, <http://num.math.uni-goettingen.de/schaback/teaching/AV2.pdf>, 2011.
27. R. Schaback. Direct discretizations with applications to meshless methods for PDEs. submitted, <http://www.num.math.uni-goettingen.de/schaback/research/group.html>, 2013.
28. R. Schaback and H. Wendland. Adaptive greedy techniques for approximate solution of large RBF systems. *Numer. Algorithms*, 24(3):239–254, 2000.
29. Robert Vertnik and Božidar Šarler. Local collocation approach for solving turbulent combined forced and natural convection problems. *Adv. Appl. Math. Mech.*, 3(3):259–279, 2011.
30. H. Wendland. *Scattered Data Approximation*. Cambridge University Press, 2005.

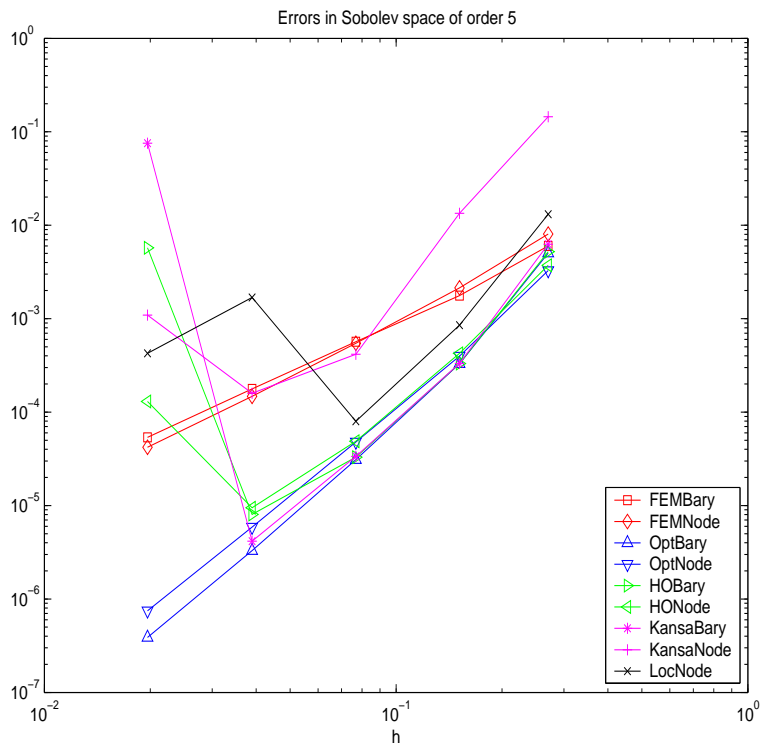


Fig. 3 Errors for Sobolev order 5

31. Z. Wu. Convergence of interpolation by radial basis functions. *Chinese Ann. Math. Ser. A*, 14:480–486, 1993.
32. Guangming Yao, Božidar Šarler, and C. S. Chen. A comparison of three explicit local meshless methods using radial basis functions. *Eng. Anal. Bound. Elem.*, 35(3):600–609, 2011.
33. Guangming Yao, Siraj ul Islam, and Božidar Šarler. A comparative study of global and local meshless methods for diffusion-reaction equation. *CMES Comput. Model. Eng. Sci.*, 59(2):127–154, 2010.
34. Guangming Yao, Siraj ul Islam, and Božidar Šarler. Assessment of global and local meshless methods based on collocation with radial basis functions for parabolic partial differential equations in three dimensions. *Eng. Anal. Bound. Elem.*, 36(11):1640–1648, 2012.

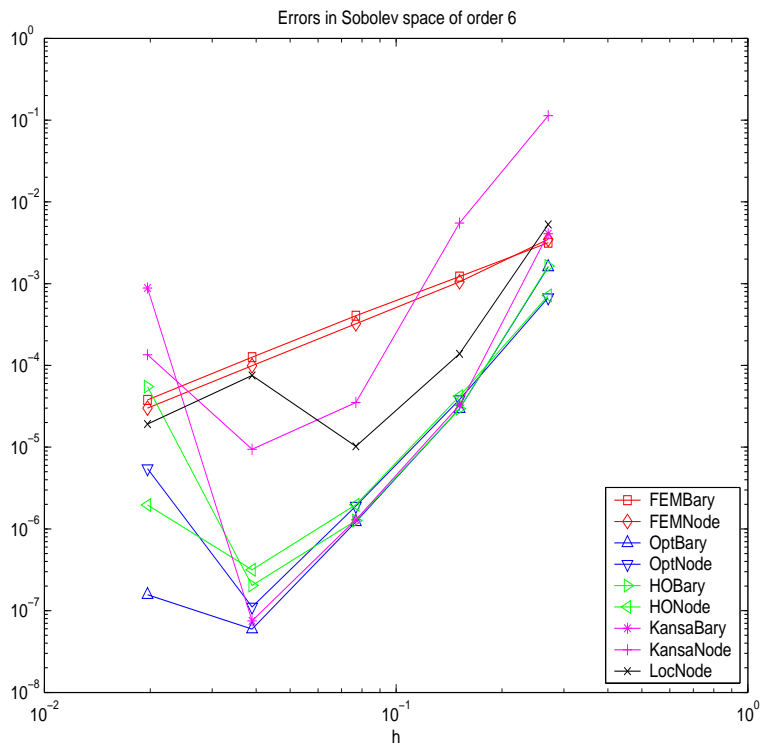


Fig. 4 Errors for Sobolev order 6

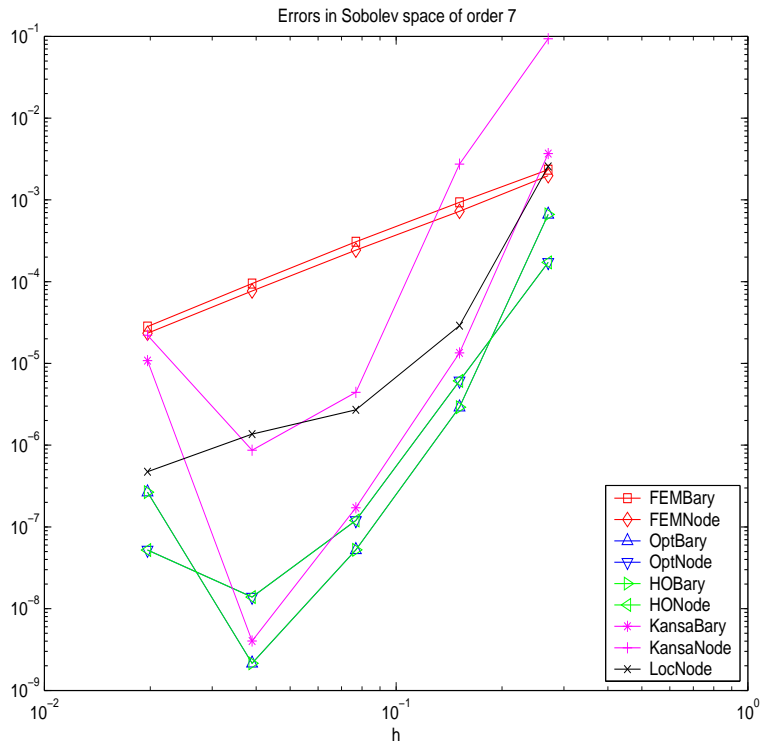


Fig. 5 Errors for Sobolev order 7

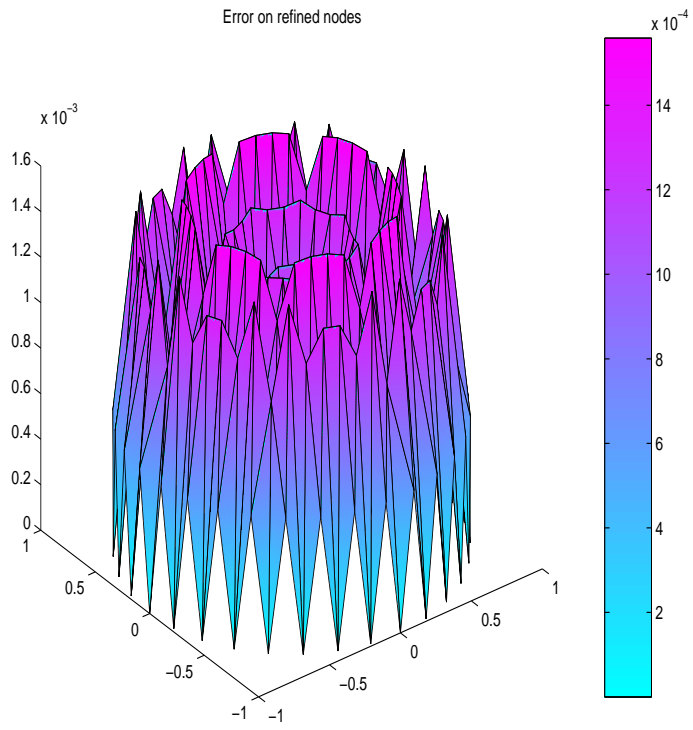


Fig. 6 FEM Errors on refined nodes

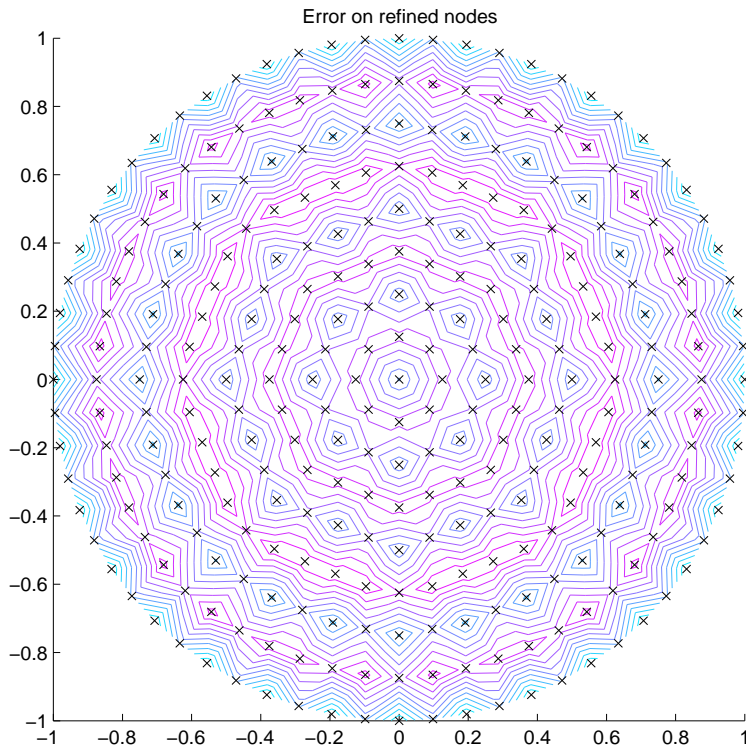


Fig. 7 FEM Errors on refined nodes

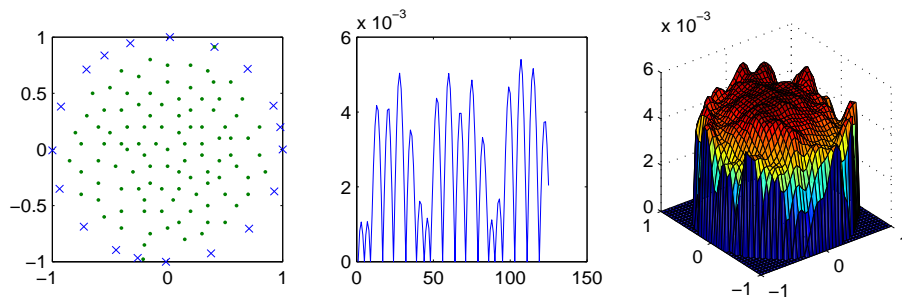


Fig. 8 Greedy optimal method in Sobolev space of order 4. Selected points, error on boundary and error in the interior

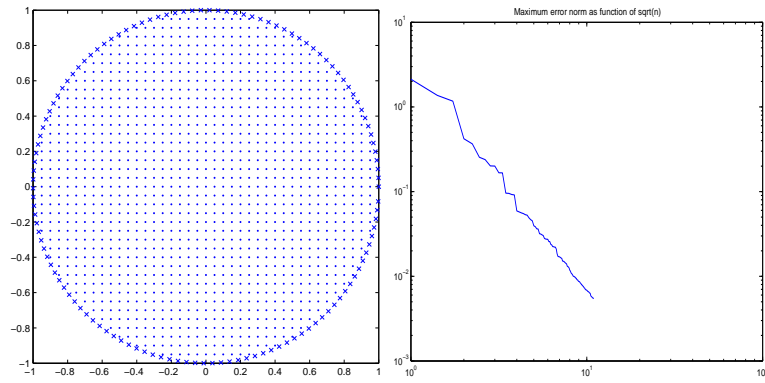


Fig. 9 Greedy optimal method in Sobolev space of order 4. Points offered for selection, and error decrease