MATLAB Implementation of Hp Finite Elements on Rectangles Using Hierarchical Basis Functions

Alexej Moskovka[1] and Jan Valdman[2,3](✉)

[1] Department of Mathematics, Faculty of Applied Sciences, University of West Bohemia, Technická 8, 30100 Plzeň, Czech Republic
[2] Department of Computer Science, Faculty of Science, University of South Bohemia, Branišovská 31, 37005 České Budějovice, Czech Republic
[3] The Czech Academy of Sciences, Institute of Information Theory and Automation, Pod vodárenskou věží 4, 18208, Prague 8, Czech Republic
jan.valdman@utia.cas.cz

Abstract. A MATLAB implementation of hierarchical shape functions on 2D rectangles is explained and available for download. Global shape functions are ordered for a given polynomial degree according to the indices of the nodes, edges, or elements to which they belong. For a uniform p-refinement, the hierarchical structure enables an effective assembly of mass and stiffness matrices. A solution to a boundary value problem is approximated for various levels of uniform h and p refinements.

Keywords: MATLAB vectorization · finite elements · mass and stiffness matrices · uniform hp-refinement · boundary value problem

1 Introduction

hp-FEM is a numerical method for solving partial differential equations based on piecewise polynomial approximations that employ elements of variable size (h) and degree of the polynomial (p). The origins of hp-FEM date back to the work of Ivo Babuška and his coauthors in the early 1980 s s (e.g. [11,12]) who discovered that the finite element method converges exponentially fast when the mesh is refined using a suitable combination of h-refinements (dividing elements into smaller ones) and p-refinements (increasing their polynomial degree). Many books (e.g. [3,4,6,9]) have been written explaining the methodology of hp-FEM accompanied by software codes [13,14] in C++. Implementing hierarchical shape functions, particularly in the case of hp adaptivity, is not straightforward, and

A. Moskovka was supported by the MSMT CR project 8J21AT001 Model Reduction and Optimal Control in Thermomechanics. J. Valdman announces the support of the Czech Science Foundation (GACR) through the GF21-06569K grant Scales and shapes in continuum thermomechanics.

R. Wyrzykowski et al. (Eds.): PPAM 2022, LNCS 13827, pp. 287–299, 2023.
https://doi.org/10.1007/978-3-031-30445-3_24

special data structures are needed [5,10]. A recent MATLAB contribution [8] provides an object-oriented approach to implement hp-FEM on triangles with adaptive h-refinement.

Our focus is on a simple hp-FEM implementation on rectangles directly based on [4]. We provide eight examples that demonstrate the basics of hp-FEM assemblies, including:

- constructions of basis functions and their isoparametric transformations to general quadrilaterals (Sect. 2),
- the ordering of global shape functions using indexing matrices (Sect. 3),
- assemblies of the mass and stiffness matrices (Sect. 4),
- solution of a particular diffusion-reaction boundary value problem using uniform h and p refinements (Sect. 5).

A complementary software for this paper is available at

https://www.mathworks.com/matlabcentral/fileexchange/111420

for download and testing. The codes for the evaluation of the shape functions were provided by Dr. Sanjib Kumar Acharya (Mumbai). The assemblies of FEM matrices are partially based on vectorization techniques of [1,2]. The names of most of the mesh attributes and the domain triangulation algorithms are taken from [7].

2 Hierarchic Shape Functions

We consider the basis functions for the dimensions of space $d \in \{1, 2\}$ (see [4]). For a reference element $T_{ref} = [-1, 1]^d$ and $p \in \mathbb{N}$ we denote by

$$S^p(T_{ref}) \tag{1}$$

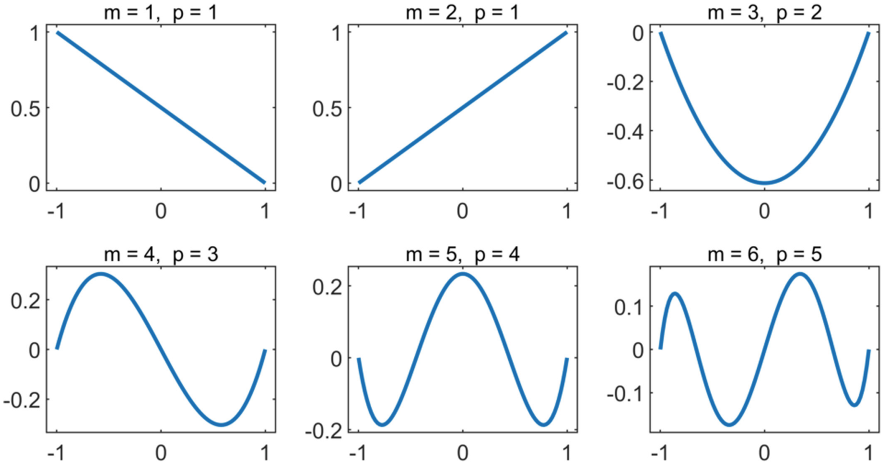

Fig. 1. The hierarchic shape basis functions $N_m(\xi)$, $m = 1, \ldots, 6$, where p is the corresponding polynomial degree.

the space of polynomials of degree p defined on T_{ref}. The basis functions that span the space are called shape functions. We define them using Legendre polynomials for $x \in [-1,1]$:

$$\begin{aligned} &P_0(x) = 1\,, \quad P_1(x) = x\,, \\ &P_{n+1}(x) = \frac{(2n+1)\,x\,P_n(x) - n\,P_{n-1}(x)}{n+1}\,, \qquad n \geq 1\,. \end{aligned} \tag{2}$$

Hierarchic shape functions on $\boldsymbol{T_{ref} = [-1,1]}$ are functions $N_m(\xi) : T_{ref} \to \mathbb{R}$, $m \in \mathbb{N}$ defined using (2) as:

$$\begin{aligned} &N_1(\xi) = \frac{1-\xi}{2}\,, \quad N_2(\xi) = \frac{1+\xi}{2}\,, \\ &N_m(\xi) = \frac{1}{\sqrt{2(2m-3)}}\big(P_{m-1}(\xi) - P_{m-3}(\xi)\big)\,, \qquad m \geq 3\,. \end{aligned} \tag{3}$$

All $N_m(\xi)$, $m \geq 3$ vanishes at the endpoints of T_{ref}.

Example 1. The first hierarchical shape functions are shown in Fig. 1 and the pictures can be reproduced by the script

```
example1_draw_hp_basis_1D
```

2.1 Hierarchic Shape Functions on $\boldsymbol{T_{ref} = [-1,1]^2}$

For $p \in \mathbb{N}$ we define the trunk space $S^p(T_{ref})$ spanned by polynomials $\xi^i \eta^j$, where $i,j \in \mathbb{N}_0$ satisfies $i + j \leq p$, supplemented by the polynomial $\xi\eta$ for $p = 1$ and the polynomials $\xi^p\eta$, $\xi\eta^p$ for $p \geq 2$. Its dimension is given by

$$n_{p,ref} = \dim(S^p(T_{ref})) = \begin{cases} 4p\,, & p \leq 3 \\ 4p + (p-2)(p-3)/2\,, & p \geq 4\,. \end{cases} \tag{4}$$

There are three types of 2D shape functions: nodal (Q1), edge, and bubble (sometimes called internal). The nodal shape functions that span the space $S^1(T_{ref})$ are defined as follows:

$$\begin{aligned} N_1(\xi,\eta) &= \frac{1}{4}(1-\xi)(1-\eta)\,, & N_2(\xi,\eta) &= \frac{1}{4}(1+\xi)(1-\eta)\,, \\ N_3(\xi,\eta) &= \frac{1}{4}(1+\xi)(1+\eta)\,, & N_4(\xi,\eta) &= \frac{1}{4}(1-\xi)(1+\eta)\,. \end{aligned} \tag{5}$$

The function of the i-th nodal shape is equal to one in the i-th node of T_{ref} and vanishes in other nodes. The edge shape functions are constructed by multiplying one-dimensional shape functions $N_m(\xi)$, $m \geq 3$ from (3) by linear blending functions. We define $\phi_p(x) = N_{p+1}(x)$, $p \geq 2$, and the edge shape functions by

$$\begin{aligned} N_p^{(1)}(\xi,\eta) &= \frac{1}{2}(1-\eta)\,\phi_p(\xi)\,, & N_p^{(2)}(\xi,\eta) &= \frac{1}{2}(1+\xi)\,\phi_p(\eta)\,, \\ N_p^{(3)}(\xi,\eta) &= \frac{1}{2}(1+\eta)\,\phi_p(-\xi)\,, & N_p^{(4)}(\xi,\eta) &= \frac{1}{2}(1-\xi)\,\phi_p(-\eta)\,. \end{aligned} \tag{6}$$

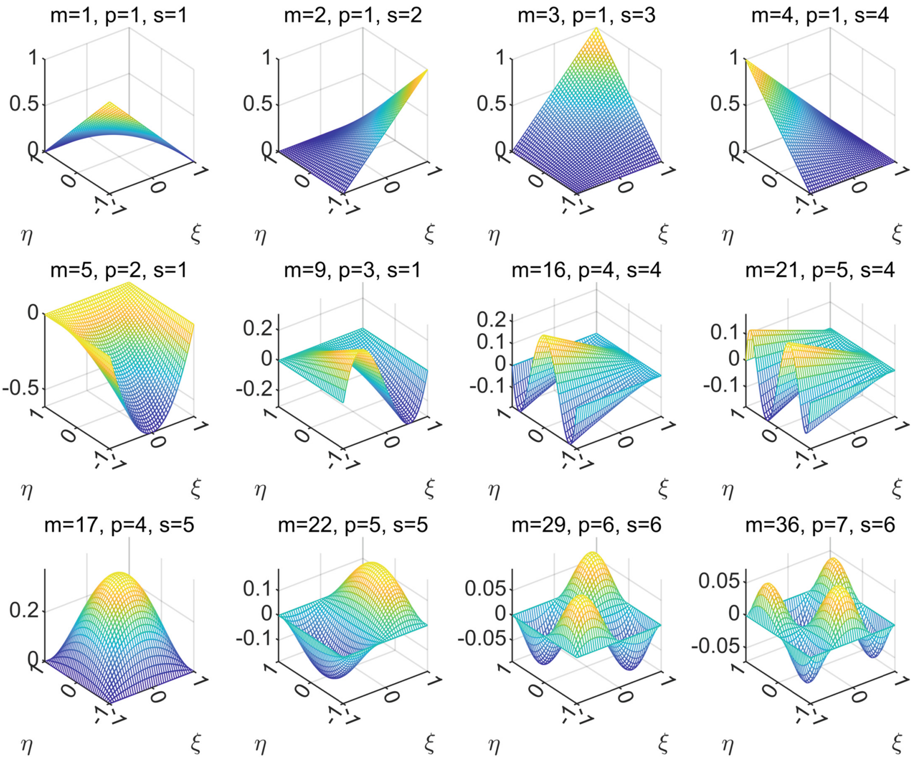

Fig. 2. Examples of nodal (the top row), edge (the middle row) and bubble (the bottom row) shape functions. Here, m denotes the index of the shape function, p its polynomial degree and s is the local index.

For $j \in \{1,2,3,4\}$, the restriction of $N_p^{(j)}$ on the j-th edge is equal to the corresponding one-dimensional edge shape function of the p-th degree, and it vanishes along the other edges. The bubble functions are defined as

$$N_p^{\beta}(\xi,\eta) = \phi_{p-(\beta+1)}(\xi)\,\phi_{\beta+1}(\eta)\,, \qquad 1 \le \beta \le p-3\,, \quad p \ge 4 \tag{7}$$

and any of them attains zero values on all edges. Table 1 shows the number of shape functions in T_{ref} for $1 \le p \le 7$.

Local Indexing. The shape functions of the p th degree in T_{ref} are ordered by a unique index $m \in \mathbb{N}$ given by

$$m = \begin{cases} 4(p-1)+s\,, & \text{for } \ p \le 4\,, \\ 4(p-1)+(p-3)(p-4)/2+s\,, & \text{for } \ p \ge 5\,, \end{cases} \tag{8}$$

where for $p = 1$: s is the index of a node $i \in \{1,2,3,4\}$,
for $p \ge 2$: s is the index of an edge $j \in \{1,2,3,4\}$,

Table 1. The numbers of shape functions.

polynomial degree p	# of nodal functions	# of edge functions	# of bubble functions	# of all functions
1	4	0	0	4
2	4	4	0	8
3	4	8	0	12
4	4	12	1	17
5	4	16	3	23
6	4	20	6	30
7	4	24	10	38

for $p \geq 4$: $s = 4 + \beta$, where β is the local index of a bubble function (7).

Example 2. Several shape functions are depicted in Fig. 2 and can be reproduced by the script

```
example2_draw_hp_basis_2D
```

The degree of the polynomial p and the local index s are evaluated by the function `[s,p] = shapeindx(m)`.

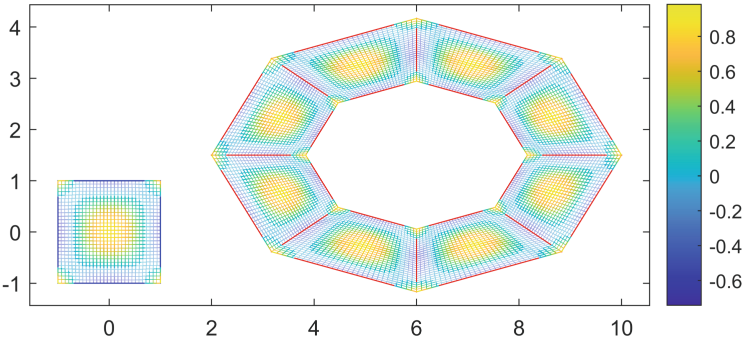

Fig. 3. The isoparametric transformation of T_{ref} indicated by the blue frame (bottom-left) to eight quadrilaterals indicated by red frames (right) and the transformation of the function $\cos\left(\frac{3\pi}{4}\xi\right)\cos\left(\frac{3\pi}{4}\eta\right)$ approximated for $p = 4$. (Color figure online)

Mapping From T_{ref} to a quadrilateral T. Transformation of a reference element T_{ref} to a quadrilateral T is performed by the isoparametric mapping $Q : T_{ref} \rightarrow T$ defined as $(x, y)(\xi, \eta) = Q(\xi, \eta)$, where

$$Q(\xi, \eta) = \Big(\sum_{i=1}^{4} X_i \, N_i(\xi, \eta) \, , \sum_{i=1}^{4} Y_i \, N_i(\xi, \eta) \Big) \, , \tag{9}$$

and (X_i, Y_i), $i \in \{1, 2, 3, 4\}$ are the coordinates of the i-th node of T. For a given $p \in \mathbb{N}$, we denote by

$$S^p(T) \tag{10}$$

the space of functions spanned by $N_m\big(Q^{-1}(x, y)\big)$, where $N_m \in S^p(T_{ref})$.

Example 3. The transformation of T_{ref} into eight different quadrilaterals forming a flattened annulus is shown in Fig. 3 and can be reproduced using the script

```
example3_isoparametric_transformation
```

It also visualizes the approximation of the function $f(\xi,\eta) = \cos\left(\frac{3\pi}{4}\xi\right)\cos\left(\frac{3\pi}{4}\eta\right)$, $(\xi,\eta) \in T_{ref}$ for $p = 4$ and its transformation into quadrilaterals.

3 Global Shape Functions

A domain $\Omega \subset \mathbb{R}^2$ is approximated by a triangulation $\mathcal{T}$ into closed elements (quadrilaterals). We denote by $\mathcal{N}, \mathcal{E}$ and $\mathcal{T}$ the sets of nodes, edges, and elements, respectively, and by $|\mathcal{N}|, |\mathcal{E}|$ and $|\mathcal{T}|$ their sizes. For a given $p \in \mathbb{N}$ we define it by

$$S^p(\mathcal{T}) \tag{11}$$

the space of all global shape functions on $\mathcal{T}$ and by n_p its dimension given by

$$n_p = \begin{cases} |\mathcal{N}| + (p-1)\,|\mathcal{E}|\,, & p \le 3\,, \\ |\mathcal{N}| + (p-1)\,|\mathcal{E}| + \frac{1}{2}(p-2)(p-3)\,|\mathcal{T}|\,, & p \ge 4\,. \end{cases} \tag{12}$$

We denote by $N_m^{(g)}$, $1 \le m \le n_p$ the m-th global shape function defined by its restrictions on elements $T_k \in \mathcal{T}$, $1 \le k \le |\mathcal{T}|$ in the following way:

$N_m^{(g)}$ is a nodal shape function corresponding to the i-th node: If T_k is adjacent to the i-th node, then $N_m^{(g)}\big|_{T_k} = \tilde{N}_{l,k}$, where $\tilde{N}_{l,k}$ is the l-th local nodal shape function on T_k which is equal to one in the i-th node. Otherwise, $N_m^{(g)} = 0$.

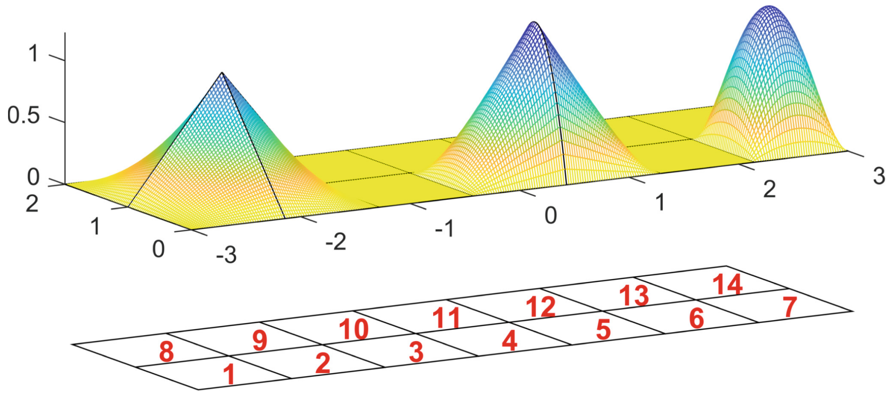

Fig. 4. Function $u \in S^p(\mathcal{T}) \in$ of (13) and the underlying rectangular mesh with indices of elements.

$N_m^{(g)}$ is an edge shape function corresponding to the j-th edge: If T_k is adjacent to the j-th edge, then $N_m^{(g)}\big|_{T_k} = \tilde{N}_{l,k}$, where $\tilde{N}_{l,k}$ is the l-th local edge shape function on T_k whose restriction on the j-th edge is the corresponding edge shape function in 1D. Otherwise, $N_m^{(g)} = 0$.

$N_m^{(g)}$ is a bubble shape function corresponding to the k-th element: $N_m^{(g)}\big|_{T_k} = \tilde{N}_{l,k}$, where $\tilde{N}_{l,k}$ is the corresponding l-th local bubble shape function on the k-th element. Otherwise, $N_m^{(g)} = 0$.

Example 4. We assume a triangulation $\mathcal{T}$ of $\Omega = (-3,3) \times (0,2)$ with $|\mathcal{N}| = 24$, $|\mathcal{E}| = 37$, $|\mathcal{T}| = 14$, $n_p = 149$ and the function $u \in S^4(\mathcal{T})$ defined as

$$u(x,y) = N_{10}^{(g)}(x,y) - 2N_{34}^{(g)}(x,y) - 2N_{142}^{(g)}(x,y)\,, \qquad (x,y) \in \Omega \tag{13}$$

shown in Fig. 4. The nodal function $N_{10}^{(g)}$ corresponds to the node adjacent to T_1, T_2, T_8, T_9, the edge function $N_{34}^{(g)}$ to the edge adjacent to T_4, T_5, and the bubble function $N_{142}^{(g)}$ is defined in T_7. Figure 4 is generated by the script

```
example4_draw_hp_basis_2D_global
```

3.1 Global Indexing

The relation between the topology of $\mathcal{T}$ and the global shape function indices is represented by three essential matrices.

Table 2. The matrix $B(\mathcal{T}, p)$ for $\mathcal{T}$ with $|\mathcal{N}| = 4$, $|\mathcal{E}| = 4$, $|\mathcal{T}| = 1$ and $p = 5$. Zero values are replaced by symbol $'-'$.

p	the global node index	the global edge index	the global element index	the local bubble index
1	1	–	–	–
1	2	–	–	–
1	3	–	–	–
1	4	–	–	–
2	–	1	–	–
2	–	2	–	–
2	–	3	–	–
2	–	4	–	–
3	–	1	–	–
3	–	2	–	–
3	–	3	–	–
3	–	4	–	–
4	–	1	–	–
4	–	2	–	–
4	–	3	–	–
4	–	4	–	–
4	–	–	1	1
5	–	1	–	–
5	–	2	–	–
5	–	3	–	–
5	–	4	–	–
5	–	–	1	1
5	–	–	1	2

A Matrix $\boldsymbol{B(\mathcal{T},p)}$ is of size $n_p \times 5$ and stores the key attributes of the global shape functions $N_m^{(g)} \in S^p(\mathcal{T})$, $1 \leq m \leq n_p$ which are uniquely determined by: the degree of $N_m^{(g)}$ (the first column of $B(\mathcal{T},p)$), the type of $N_m^{(g)}$ (nodal, edge or bubble) specified by the global index of the respective node (the 2nd column), edge (the 3rd column), or element (the 4th column). Additionally, the type of bubble requires a local index of a bubble (the 5th column). The key advantage of this approach is that for the same $\mathcal{T}$ and $1 \leq p_1 < p_2$ the first n_{p_1} rows of both matrices $B(\mathcal{T},p_1)$ and $B(\mathcal{T},p_2)$ are the same.

A Matrix $\boldsymbol{C(\mathcal{T},p)}$ of size $n_{p,ref} \times |\mathcal{T}|$ collects for individual elements the indices of the corresponding global functions. In particular, $C_{l,k}(\mathcal{T},p) = m$ means that $N_m^{(g)}\big|_{T_k}$ corresponds to the l-th local shape function on the k-th element.

A Matrix $\boldsymbol{S(\mathcal{T},p)}$ of size $n_{p,ref} \times |\mathcal{T}|$ for the l-th row and the k-th column returns the sign of the l-th local function on the k-th element. For edges adjacent to two elements, the corresponding local edge functions of odd degrees have to be assigned opposite signs to ensure the continuity of the corresponding global edge functions.

Example 5. We assume a triangulation $\mathcal{T}$ with $|\mathcal{N}| = 4$, $|\mathcal{E}| = 4$, and $|\mathcal{T}| = 1$. Tab. 2 depicts for $p = 5$ the corresponding matrix $B(\mathcal{T},p)$ with 23 (the value of n_p) rows and 5 columns that can be generated by the script

```
example5_B_matrix
```

Example 6. We assume a triangulation $\mathcal{T}$ with $|\mathcal{N}| = 6$, $|\mathcal{E}| = 7$ and $|\mathcal{T}| = 2$. Tab. 3 depicts for $p = 3$ the corresponding matrices $C(\mathcal{T},p)$ and $S(\mathcal{T},p)$ with 12 (the value of $n_{p,ref}$) rows and 2 (the value of $\mathcal{T}$) columns that can be generated by script

```
example6_C_S_matrices
```

Fig. 5 shows the global edge function $N_{17}^{(g)}$ of the 3rd degree. The left part exploits the right orientation with opposite signs providing continuity, and the right part exploits the wrong orientation leading to discontinuity.

Table 3. Matrices $C(\mathcal{T},p)$ (left) and $S(\mathcal{T},p)$ (right) of Example 6.

l	T_1	T_2
1	1	2
2	2	3
3	5	6
4	4	5
5	7	9
6	10	11
7	12	13
8	8	10
9	14	16
10	**17**	18
11	19	20
12	15	**17**

l	T_1	T_2
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
6	1	1
7	1	1
8	1	1
9	1	1
10	-1	1
11	1	1
12	1	**1**

4 Mass and Stiffness Matrices

4.1 The Reference Mass and Stiffness Matrices

are for a given $p \in \mathbb{N}$ matrices of size $n_{p,ref} \times n_{p,ref}$ defined by

$$M_{i,j}^{ref} = \int_{T_{ref}} N_i N_j \mathrm{d}\mathbf{x}\,, \qquad K_{i,j}^{ref} = \int_{T_{ref}} \nabla N_i \cdot \nabla N_j \mathrm{d}\mathbf{x}\,. \tag{14}$$

Functions

```
mass_matrixQp_2D_reference(p)

stiffness_matrixQp_2D_reference(p)
```

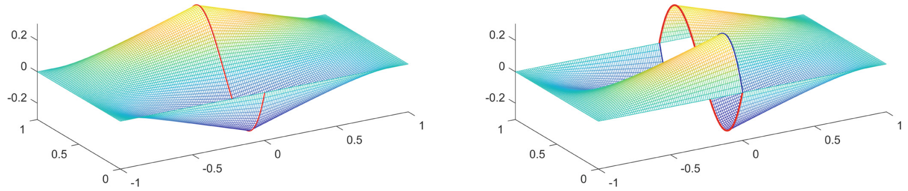

Fig. 5. The right (left) and wrong (right) orientation of $N_{17}^{(g)}$ of Example 6.

evaluate the corresponding reference mass and stiffness matrices using the Gaussian quadrature rule. For a given p, the function `[X,W] = intrec_hp(p)` returns the Gauss points $X \in T_{ref}$ together with the corresponding weights stored in a vector W.

4.2 The Global Mass and Stiffness Matrices

are for a specific p and $\mathcal{T}$ matrices of size $n_p \times n_p$ defined by

$$M_{i,j} = \int_{\mathcal{T}} N_i^{(g)} N_j^{(g)} \mathrm{d}\mathbf{x}\,, \qquad K_{i,j} = \int_{\mathcal{T}} \nabla N_i^{(g)} \cdot \nabla N_j^{(g)} \mathrm{d}\mathbf{x} \tag{15}$$

and assembled by adding the contributions of local mass and stiffness matrices $M(T_k)$ and $K(T_k)$ of size $n_{p,ref} \times n_{p,ref}$ to the corresponding entries. In particular, $M_{i,j}(T_k)$ and $K_{i,j}(T_k)$ contribute to the $\boldsymbol{c}_i^k$-th row and the $\boldsymbol{c}_j^k$-th column of M and K, respectively, where $\boldsymbol{c}^k$ is the k-th column vector of $C(\mathcal{T}, p)$.

For any $T_k \in \mathcal{T}$, $1 \leq k \leq |\mathcal{T}|$ the local mass matrix $M(T_k)$ is given by

$$M(T_k) = \frac{|T_k|}{|T_{ref}|} M^{ref} = \frac{|T_k|}{4} M^{ref}\,, \tag{16}$$

however, this formula cannot be applied to the assembly of the local stiffness matrix $K(T_k)$. Instead, we apply the chain rule to evaluate

$$\begin{aligned} K_{i,j}(T_k) &= \int_{T_k} \nabla \tilde{N}_i(x,y) \cdot \nabla \tilde{N}_j(x,y) \mathrm{d}\mathbf{x} = \\ &= \int_{T_k} \nabla N_i\big(Q^{-1}(x,y)\big) \cdot \nabla N_j\big(Q^{-1}(x,y)\big) \mathrm{d}\mathbf{x}\,, \end{aligned} \tag{17}$$

where $\tilde{N}_i$ and N_i, $1 \le i \le n_{p,ref}$ are the i-th local function on T_k and T_{ref}, respectively. Using the chain rule, one can write

$$\nabla \tilde{N}_i\big(Q^{-1}(x,y)\big) = \Big(\frac{\partial \tilde{N}_i}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial \tilde{N}_i}{\partial \eta}\frac{\partial \eta}{\partial x}, \ \frac{\partial \tilde{N}_i}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial \tilde{N}_i}{\partial \eta}\frac{\partial \eta}{\partial y}\Big)\,, \tag{18}$$

where

$$\Big(\frac{\partial \xi}{\partial x}, \frac{\partial \eta}{\partial x}\Big) = \frac{\partial Q^{-1}}{\partial x}(x,y)\,, \quad \Big(\frac{\partial \xi}{\partial y}, \frac{\partial \eta}{\partial y}\Big) = \frac{\partial Q^{-1}}{\partial y}(x,y).$$

Additionally, we apply derivative of the formula of inverse function to evaluate

$$\begin{pmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{pmatrix} = \nabla Q^{-1}(x,y) = \big(\nabla Q(\xi,\eta)\big)^{-1} = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{pmatrix}^{-1}. \tag{19}$$

Table 4. Mesh properties (left) of the uniform mesh refinements of Example 7 and numbers of global shape functions for different polynomial orders (right).

level	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{T}\rvert$
2	$2.5 \cdot 10^1$	$4.0 \cdot 10^1$	$1.6 \cdot 10^1$
3	$8.1 \cdot 10^1$	$1.4 \cdot 10^2$	$6.4 \cdot 10^1$
4	$2.9 \cdot 10^2$	$5.4 \cdot 10^2$	$2.6 \cdot 10^2$
5	$1.1 \cdot 10^3$	$2.1 \cdot 10^3$	$1.0 \cdot 10^3$
6	$4.2 \cdot 10^3$	$8.3 \cdot 10^3$	$4.1 \cdot 10^3$
7	$1.7 \cdot 10^4$	$3.3 \cdot 10^4$	$1.6 \cdot 10^4$
8	$6.6 \cdot 10^4$	$1.3 \cdot 10^5$	$6.6 \cdot 10^4$
9	$2.6 \cdot 10^5$	$5.3 \cdot 10^5$	$2.6 \cdot 10^5$

level	n_1	n_2	n_3	n_4	n_5
2	$2.5 \cdot 10^1$	$6.5 \cdot 10^1$	$1.1 \cdot 10^2$	$1.6 \cdot 10^2$	$2.3 \cdot 10^2$
3	$8.1 \cdot 10^1$	$2.3 \cdot 10^2$	$3.7 \cdot 10^2$	$5.8 \cdot 10^2$	$8.5 \cdot 10^2$
4	$2.9 \cdot 10^2$	$8.3 \cdot 10^2$	$1.4 \cdot 10^3$	$2.2 \cdot 10^3$	$3.2 \cdot 10^3$
5	$1.1 \cdot 10^3$	$3.2 \cdot 10^3$	$5.3 \cdot 10^3$	$8.4 \cdot 10^3$	$1.3 \cdot 10^4$
6	$4.2 \cdot 10^3$	$1.3 \cdot 10^4$	$2.1 \cdot 10^4$	$3.3 \cdot 10^4$	$5.0 \cdot 10^4$
7	$1.7 \cdot 10^4$	$5.0 \cdot 10^4$	$8.3 \cdot 10^4$	$1.3 \cdot 10^5$	$2.0 \cdot 10^5$
8	$6.6 \cdot 10^4$	$2.0 \cdot 10^5$	$3.3 \cdot 10^5$	$5.3 \cdot 10^5$	$7.9 \cdot 10^5$
9	$2.6 \cdot 10^5$	$7.9 \cdot 10^5$	$1.3 \cdot 10^6$	$2.1 \cdot 10^6$	$3.2 \cdot 10^6$

Example 7. For $\Omega = T_{ref} = [-1,1]^2$ the script

```
example7_M_K_matrices_times
```

runs a nested loop on different p and levels of uniform refinements of Ω. The mass and stiffness matrices are assembled by the functions `mass_matrixQp_2D(mesh)` and `stiffness_matrixQp_2D(mesh)`, respectively. Tables 4 and 5 contain the properties of the mesh and the corresponding assembly times. Assembly times were obtained on a MacBook Air (M1 processor, 2020) with 16 GB memory running MATLAB R2022a.

Table 5. Assembly times of mass and stiffness matrices in Example 7 measured in seconds.

	$p = 1$		$p = 2$		$p = 3$		$p = 4$		$p = 5$	
level	M [s]	K [s]	M [s]	K [s]	M [s]	K [s]	M [s]	K [s]	M [s]	K [s]
2	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01
3	0.00	0.01	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.01
4	0.00	0.01	0.00	0.00	0.01	0.02	0.01	0.02	0.01	0.03
5	0.00	0.01	0.00	0.01	0.01	0.02	0.01	0.05	0.02	0.09
6	0.00	0.02	0.01	0.04	0.02	0.08	0.03	0.16	0.07	0.33
7	0.01	0.08	0.03	0.15	0.07	0.32	0.25	0.71	0.45	1.46
8	0.04	0.31	0.15	0.66	0.48	1.44	0.98	3.14	2.00	6.20
9	0.22	1.29	0.91	2.86	1.94	6.01	5.26	12.43	11.17	27.31

5 Solving Partial Differential Equation in 2D

We solve a diffusion-reaction boundary value problem

$$-\varDelta u + \nu\, u = f \quad \text{in } \varOmega\,, \qquad \frac{\partial u}{\partial n} = 0 \quad \text{on } \partial\varOmega \tag{20}$$

by applying the hp-FEM method to the weak formulation of (20) given by

$$\int_{\mathcal{T}} \nabla u \cdot \nabla N_m^{(g)} \mathrm{d}\mathbf{x} \,+\, \nu \int_{\mathcal{T}} u\, N_m^{(g)} \mathrm{d}\mathbf{x} = \int_{\mathcal{T}} f\, N_m^{(g)} \mathrm{d}\mathbf{x}\,, \quad \forall\, N_m^{(g)} \in S^p(\mathcal{T})\,. \tag{21}$$

It leads to an algebraic system of linear equations in the form of

$$(K + \nu\, M)\, \tilde{u}_n = b\,, \tag{22}$$

where M and K are global mass and stiffness matrices, u_n is the numerical solution of (21) represented by the vector $\tilde{u}_n \in \mathbb{R}^{n_p}$ of coefficients in the corresponding hp basis and the vector $b \in \mathbb{R}^{n_p}$ is given by $b_m = \int_{\mathcal{T}} f N_m^{(g)} \mathrm{d}\mathbf{x}$. We assume the domain $\varOmega = T_{ref} = [-1, 1]^2$ and the parameter $\nu = 0.1$. It is easy to show that

$$u(x, y) = (1 - x^2)^2\, (1 - y^2)^2$$

represents the solution of (20) corresponding to the function

$$f(x, y) = \nu\, u(x, y) - 4\big(-2 + 5y^2 - y^4 + x^4(-1 + 3y^2) + x^2(5 - 12y^2 + 3y^4)\big)$$

for $(x, y) \in \varOmega$. To study the convergence of hp approximations, we take several levels of uniform refinements of $\varOmega$ defined by $|T|$ squares of same size, where $|T| = 4^{level}, level = 1, \ldots, 7$ and solve (22) for different polynomial orders p, $1 \le p \le p_{max} = 5$. The exact solution u is approximated in $S^{\tilde{p}}(\mathcal{T})$, $\tilde{p} = p_{max} + 2$ by the vector $\tilde{u}$. The corresponding error e in the energy norm is given by

$$e^2 = \int_{\mathcal{T}} \big(\|\nabla u - \nabla u_n\|^2 + (u - u_n)^2\big) \mathrm{d}\mathbf{x} \,\approx\, (\tilde{u} - \tilde{u}_n)^T (K + M)(\tilde{u} - \tilde{u}_n)\,. \tag{23}$$

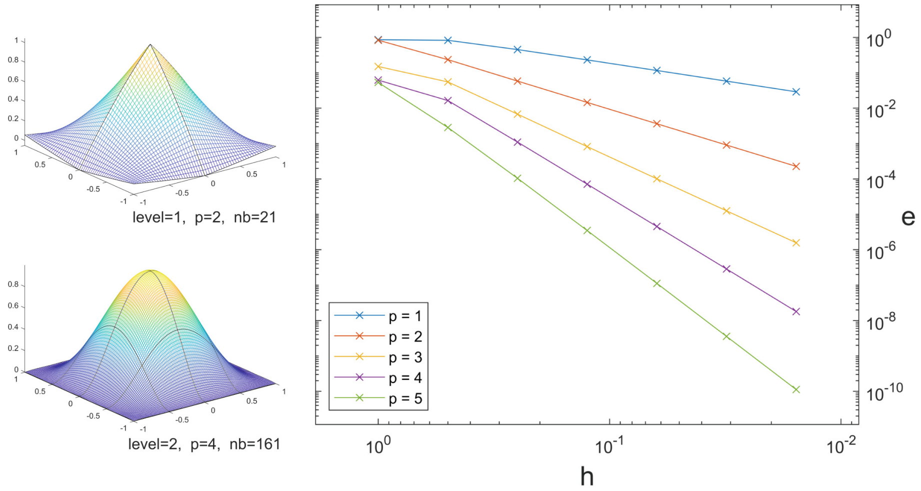

Fig. 6. Examples of solutions of (21) and convergence in the energy norm.

The script **example8_diffusion_reaction_BVP** utilizes a nested for loop on p (inside) and mesh refinement levels (outside). Two particular numerical solutions are shown in Fig. 6 (left). The corresponding errors (23) are shown in Fig. 6 (right), where the crosses on different lines correspond to the mesh refinement levels and both x and y labels are log-scaled. The calculation confirms a theoretical expectation $e \approx h^p$, where p is a chosen polynomial order and h is a chosen square size satisfying $|\mathcal{T}| = 4\,h^{-2}$.

References

1. Anjam, I., Valdman, J.: Fast MATLAB assembly of FEM matrices in 2D and 3D: Edge elements. Appl. Math. Comput. **267**, 252–263 (2015)
2. Rahman, T., Valdman, J.: Fast MATLAB assembly of FEM matrices in 2D and 3D: Nodal elements. Appl. Math. Comput. **219**, 7151–7158 (2013)
3. Szabó, B., Babuška, I.: Finite Element Analysis. Wiley-Interscience, New York (1991)
4. Szabó, B., Babuška, I.: Introduction to Finite Element Analysis, John Wiley & Sons (2011)
5. Bangerth, W., Kayser-Herold, O.: Data structures and requirements for hp finite element software. ACM Trans. Math. Softw. (TOMS) **36**(1), 1–31 (2009)
6. Šolín, P., Segeth, K., Doležel, I.: Higher-Order Finite Element Methods, Chapman & Hall/CRC (2004)
7. Moskovka, A., Valdman, J.: Fast MATLAB evaluation of nonlinear energies using FEM in 2D and 3D: nodal elements. Appl. Math. Comput. **424**, 127048 (2022)
8. Innerberger, M., Praetorius, D.: MooAFEM: an object oriented Matlab code for higher-order adaptive FEM for (nonlinear) elliptic PDEs. Appl. Math. Comput. **442**, 127731 (2023)

9. Demkowicz, L.: Computing with hp-ADAPTIVE FINITE ELEMENTS, Volume 1, Chapman & Hall/CRC (2007)
10. Demkowicz, L., Oden, J.T., Rachowicz, W., Hardy, O.: Toward a Universal h-p Adaptive Finite Element Strategy. Part 1. Constrained Approximation and Data Structure. Comput. Methods Appl. Mech. Eng. **77**(1–2), 79–112 (1989)
11. Babuška, I., Szabó, B., Katz, I.: The p-version of the finite element method. SIAM J. Num. Anal. **18**(3), 515–545 (1981)
12. Babuška, I., Guo, B.Q.: The h-p version of the finite element method. Comput. Mech. **1**, 21–41 (1986)
13. Schöberl, J.: C++11 Implementation of Finite Elements in NGSolve. Vienna University of Technology, Institute for Analysis and Scientific Computing (2014)
14. Šolín, P., Korous, L., Kus, P.: Hermes2D, a C++ library for rapid development of adaptive hp-FEM and hp-DG solvers. J. Comput. Appl. Math. **270**, 152–165 (2014)