

# Balancing truncation and round-off errors in practical FEM: one-dimensional analysis

Jie Liu<sup>a,\*</sup>, Matthias Möller<sup>a</sup>, Henk M. Schuttelaars<sup>a</sup>

<sup>a</sup>*Delft Institute of Applied Mathematics  
Delft University of Technology  
Van Mourik Broekmanweg 6, 2628 XE Delft, The Netherlands*

---

## Abstract

In finite element methods (FEMs), the accuracy of the solution cannot increase indefinitely because the round-off error increases when the number of degrees of freedom (DoFs) is large enough. This means that the accuracy that can be reached is limited. A priori information of the highest attainable accuracy is therefore of great interest. In this paper, we devise an innovative method to obtain the highest attainable accuracy. In this method, the truncation error is extrapolated when it converges at the analytical rate, for which only a few primary  $h$ -refinements are required, and the bound of the round-off error is provided through extensive numerical experiments. The highest attainable accuracy is obtained by minimizing the sum of these two types of errors. We validate this method using a one-dimensional Helmholtz equation in space. It shows that the highest attainable accuracy can be accurately predicted, and the CPU time required is much less compared with that using the successive  $h$ -refinement.

*Keywords:* Finite Element Method (FEM), error estimation, optimal number of degrees of freedom,  $hp$ -refinement strategy.

---

## 1. Introduction

Many problems in engineering sciences and industry are modelled mathematically by initial-boundary value problems comprising systems of coupled, nonlinear partial and/or ordinary differential equations. These problems often consider complex geometries, with initial and/or boundary conditions that depend on measured data [1]. In some applications, not only the solution, but also its derivatives are of interest [1, 2]. For many problems of practical interest, analytical or semi-analytical solutions are not available, and hence one has to resort to numerical solution methods, such as the finite difference, finite volume, and finite element methods. The latter will be adopted throughout this paper and applied to one-dimensional boundary value problems.

---

\*Corresponding author

*Email addresses:* j.liu-5@tudelft.nl (Jie Liu), m.moller@tudelft.nl (Matthias Möller), h.m.schuttelaars@tudelft.nl (Henk M. Schuttelaars)

The accuracy of the numerically obtained solution is influenced by many sources of errors [3]: firstly, errors in the set-up of the models, such as the simplification of the domain and governing equations and the approximation of the initial and boundary conditions; next, truncation errors due to the discretization of the computational domain and the use of basis functions for the function spaces defined on it; then, the iteration error resulting from the artificially controlled tolerance of iterative solvers; finally, the round-off error due to the adoption of finite-precision computer arithmetics, rather than exact arithmetics. One tacitly assumes that most errors are well-balanced and/or negligibly small. In particular, the round-off error is often ignored based on the argument that it will be ‘sufficiently small’ if just IEEE-754 double-precision floating-point arithmetics [4] are adopted. In this paper, the focus is on the overall discretization error due to truncation and round-off. In particular, we will show that the latter might very well have a significant influence on the overall accuracy and propose a practical strategy to balance both error contributors.

The discretization error strongly depends on the number of degrees of freedom (“DoFs”), denoted by  $N_h^{(p)}$ , which is a function of the mesh width  $h$  and the approximation order  $p$ . The truncation error, denoted by  $E_T$ , dominates the discretization error only when  $N_h^{(p)}$  is not too large, and it decreases with increasing mesh resolution and element degree as it can be expected from finite element theory [5]. Based on this, the commonly used approaches to reduce the truncation error are to reduce the mesh width ( $h$ -refinement), increase the approximation order ( $p$ -refinement), or apply both strategies simultaneously ( $hp$ -refinement) [6]. The round-off error, denoted by  $E_R$ , is, however, only negligible for moderately small values of  $N_h^{(p)}$  and dominates the overall discretization error if more and more DoFs are employed [7]. Consequently, for a particular approximation order  $p$ , by performing  $h$ -refinement, the best accuracy is obtained at the break-even point where the discretization error is the smallest. We denote the highest accuracy by  $E_{\min}^{(p)}$  and the optimal number of DoFs by  $N_{\text{opt}}^{(p)}$ .

While  $N_{\text{opt}}^{(p)}$  is typically impractically large if low(est)-order approximations are used, it can be very small if high-order approximations are adopted, which are nowadays becoming more and more popular, and make the results more prone to be polluted by round-off errors. Despite this alarming observation, to the authors’ best knowledge, only very few publications address the impact of accumulated round-off errors on the overall accuracy of the final solution [8, 9] or take them into account explicitly in the error-estimation procedure [10, 11]. The general rule of thumb is still to perform as many  $h$ -refinements as possible considering the available computer hardware.

The aim of this paper is to systematically analyze the influence of the round-off error on the discretization error, for the solution, and its first and second derivative, and propose a practical approach for obtaining  $E_{\min}^{(p)}$ . The scope is restricted to one-dimensional model problems, i.e. Poisson, diffusion and Helmholtz equations, for which both the standard finite element method (FEM) and the mixed FEM[12] are considered. To assess the general applicability of the aforementioned approach, the following factors are investigated: the element degree over a wide range, first and second derivative of the solution, type of boundary conditions

and method of implementing them, choice and configuration of the linear system solver, order of magnitude of the solution and its derivatives, and equation type.

The paper is organized as follows. The model problem, finite element formulation and numerical implementation are described in Section 2. The general behavior of the discretization error and the approach to predict  $E_{\min}^{(p)}$  are discussed in Section 3. Numerical results for determining the offset of the round-off error are shown in Section 4. The algorithm for realizing the approach is put forward in Section 5, followed by its validation by a Helmholtz problem in Section 6. The conclusions are drawn in Section 7.

## 2. Model problem, finite element formulation and numerical implementation

### 2.1. Model problem

Consider the following one-dimensional second-order differential equation:

$$-(d(x)u_x)_x + r(x)u(x) = f(x), \quad x \in I = (0, 1), \quad (1)$$

with  $u$  denoting the unknown variable, which can either be real or complex,  $f(x) \in L^2(I)$  a prescribed right-hand side, and  $d(x)$  and  $r(x)$  continuous coefficient functions. By choosing  $d(x) = 1$  and  $r(x) = 0$ , Eq. (1) reduces to the Poisson equation; for  $d(x) > 0$  and not constant, when  $r(x) = 0$ , the diffusion equation is found, and when  $r(x) \neq 0$ , we obtain the Helmholtz equation. The boundary conditions are  $u(x) = g(x)$  on  $\Gamma_D$  and  $d(x)u_x = h(x)$  on  $\Gamma_N$ . Here,  $\Gamma_D$  and  $\Gamma_N$  are the boundaries where, respectively, Dirichlet and Neumann boundary conditions are imposed. In this paper, for all the equations investigated, the existence of the second derivative is guaranteed in the weak sense, i.e.  $u \in H^2(I)$ .

### 2.2. Finite element formulation

For convenience, we introduce the two inner products:

$$(f_1(x), f_2(x)) = \int_I f_1(x)f_2(x) dx, \quad (2a)$$

$$(g_1(x), g_2(x))_\Gamma = g_1(x_0)g_2(x_0), \quad (2b)$$

where  $f_1(x)$ ,  $f_2(x)$ ,  $g_1(x)$  and  $g_2(x)$  are continuous functions defined on the unit interval  $I$ ,  $\Gamma$  denotes the boundary of  $I$ , and  $x_0$  denotes the value of  $x$  on  $\Gamma$ .

### 2.2.1. The standard FEM

The weak form of Eq. (1) is derived in Appendix A.1. Imposing the Dirichlet boundary conditions strongly, the weak form reads:

Weak form 1

Find  $u \in H_D^1(I)$  such that:

$$(\eta_x, du_x) + (\eta, ru) = (\eta, f) + (\eta, hn)_{\Gamma_N} \quad \forall \eta \in H_{D0}^1(I),$$

with

$$H_D^1(I) = \{t \mid t \in H^1(I), t = g \text{ on } \Gamma_D\},$$

$$H_{D0}^1(I) = \{t \mid t \in H^1(I), t = 0 \text{ on } \Gamma_D\},$$

where  $n$  is 1 at  $x = 1$ , and  $-1$  at  $x = 0$ .

(3)

By imposing the Dirichlet boundary conditions in the weak sense[13], the weak form reads:

Weak form 2

Find  $u \in H^1(I)$  such that:

$$(\eta_x, du_x) + (\eta, ru) - (\eta, du_x n)_{\Gamma_D} + (\eta_x, un)_{\Gamma_D} - (\eta, \rho un)_{\Gamma_D}$$

$$= (\eta, f) + (\eta, hn)_{\Gamma_N} + (\eta_x, gn)_{\Gamma_D} - (\eta, \rho gn)_{\Gamma_D} \quad \forall \eta \in H^1(I),$$

where  $\rho$  is a positive value that serves as the penalty parameter.

(4)

Note that, the terms in the right-hand sides of Eqs. (3)–(4) consist of information of the Neumann boundary conditions, and hence, if no Neumann boundary conditions are prescribed, these terms vanish. We use Weak form 1 if not stated otherwise. Next, we approximate the exact solution  $u_{\text{exc}}$  by a linear combination of a finite number of basis functions:

$$u_{\text{exc}} \approx u_h^{(p)} = \sum_{i=1}^m u_i \varphi_i^{(p)}. \quad (5)$$

Here,  $\varphi_i^{(p)}$  are  $C^0$ -continuous Lagrange basis functions of degree  $p$ , denoted as  $P_p$ , with Gauss-Lobatto support points  $x_j$ , which feature the Kronecker-delta property, i.e.  $\varphi_i^{(p)}(x_j) = \delta_{ij}$ . The coefficients  $u_i$  are the values of  $u_h^{(p)}$  at the DoFs, as a direct consequence of the Kronecker-delta property of  $\varphi_i^{(p)}$ . The number of DoFs of  $u_h^{(p)}$ , denoted by  $m$ , equals  $p \times t + 1$ , where  $t$  is the total number of the grid cells. Finally, taking the test function  $\eta$  equal to  $\varphi_k^{(p)}$ ,  $k = 1, 2, \dots, m$ , we obtain

$$AU = F, \quad (6)$$

where  $A$  is the stiffness matrix,  $F$  the right-hand side and  $U$  the discrete solution, i.e. the vector of the coefficients  $u_i$ .

### 2.2.2. The mixed FEM

As a first step, we introduce the auxiliary variable

$$v(x) = -d(x)u_x, \quad (7a)$$

allowing Eq. (1) to be rewritten as

$$-v_x + r(x)u(x) = f(x). \quad (7b)$$

Unlike the standard FEM, for the mixed FEM, the essential boundary conditions are imposed on  $\Gamma_N$ , and the natural boundary conditions on  $\Gamma_D$ . The weak form of Eq. (1) using the mixed FEM, derived in Appendix A.2, is given by:

Weak form 3

Find  $v \in H_N^1(I)$  and  $u \in L^2(I)$  such that:

$$(w, d^{-1}v) - (w_x, u) = -(w, gn)_{\Gamma_D} \quad \forall w \in H_{N0}^1(I), \quad (8a)$$

$$-(q, v_x) + (q, ru) = (q, f) \quad \forall q \in L^2(I), \quad (8b)$$

with

$$H_N^1(I) = \{t \mid t \in H^1(I), t = -h \text{ on } \Gamma_N\},$$

$$H_{N0}^1(I) = \{t \mid t \in H^1(I), t = 0 \text{ on } \Gamma_N\}.$$

Next, we approximate the exact gradient  $v_{\text{exc}}$  and the exact solution  $u_{\text{exc}}$  by a linear combination of a finite number of basis functions:

$$v_{\text{exc}} \approx v_h^{(p)} = \sum_{i=1}^n v_i \varphi_i^{(p)}, \quad (9a)$$

$$u_{\text{exc}} \approx u_h^{(p-1)} = \sum_{j=1}^p u_{cj} \psi_j^{(p-1)} \text{ in cell } c, \text{ for } c = 1, 2, \dots, t. \quad (9b)$$

where  $\varphi_i^{(p)}$  are of the same type of basis functions used in Eq. (5), with coefficients  $v_i$  the associated values of  $v_h^{(p)}$  at the DoFs;  $\psi_j^{(p-1)}$  are discontinuous Lagrange basis functions of degree  $p-1$ , denoted as  $P_{p-1}^{\text{disc}}$ , with coefficients  $u_{cj}$  the associated values of  $u_h^{(p-1)}$  at the DoFs. This pair of elements will be referred to as  $P_p/P_{p-1}^{\text{disc}}$ . Since the use of discontinuous basis functions, there are two independent  $u_{cj}$  at cell interfaces. The number of DoFs for  $v_h^{(p)}$ , denoted by  $n$ , equals  $p \times t + 1$ , and the number of DoFs for  $u_h^{(p-1)}$  equals  $p \times t$ . Finally, replacing the test functions  $w$  and  $q$  by  $\varphi_k^{(p)}$ ,  $k = 1, 2, \dots, p \times t + 1$ , and  $\psi_e^{(p-1)}$ ,  $e = 1, 2, \dots, p \times t$ , respectively, the resulting coupled linear system of equations that has to be solved reads:

$$\begin{bmatrix} M & B \\ B^\top & 0 \end{bmatrix} \begin{bmatrix} V \\ U \end{bmatrix} = \begin{bmatrix} G \\ H \end{bmatrix}, \quad (10)$$

where the mass matrix  $M$ , the discrete gradient operator  $B$ , and its transpose, the discrete divergence operator  $B^\top$ , are the components of the discrete left-hand side of Eqs. (8a)–(8b),  $G$  and  $H$  are the components of the right-hand side, and  $V$  and  $U$  are the discrete first derivative and solution, i.e. the vectors of the coefficients  $v_i$  and  $u_{cj}$ , respectively.

For the sake of readability, we will drop the superscript  $(p)$ , whenever the approximation order is clear from the context.

### 2.3. Numerical implementation

In what follows, we demonstrate how to obtain the numerical solution for Eq. (1) with specific coefficients and assess its quality. For the latter, both the error, obtained using the analytical solution or the finer numerical solution, and the order of convergence are investigated.

#### 2.3.1. Solution technique

Unless stated otherwise, all results are computed in IEEE-754 double precision [4] using the deal.II finite element code [14] that provides subroutines for creating the computational grid, building and solving the system of equations, and computing the error norms.

The computational mesh is obtained by globally refining a single element that covers the interval  $I$ , and the Dirichlet boundary conditions are imposed strongly unless stated otherwise. The former means that, when the solution is real valued, using the standard FEM, the number of DoFs equals  $2^{REF} \times p + 1$  at the  $REF$ th refinement; using the mixed FEM, the number of DoFs equals  $2 \times 2^{REF} \times p + 1$  at the  $REF$ th refinement. For complex-valued problems, the above numbers double since deal.II does not provide native support for complex-valued problems and, hence, all components need to be split into their real and imaginary parts.

To compute the occurring integrals, sufficiently accurate Gaussian quadrature formulas are used. Furthermore, unless stated otherwise, to solve the matrix equation, the UMFPACK solver [15], which implements the multi-frontal LU factorization approach, is used as it results in relatively fast computations of the problems considered in this paper, and prevents the iteration errors for the iterative solvers.

#### 2.3.2. Error estimation

For the numerical results  $var_h$ , where  $var$  can be  $u$ ,  $u_x$  and  $u_{xx}$ , the discretization error measured in the  $L_2$  norm is used. This measure contains all types of errors, for example the truncation error, round-off error, etc. It is defined as

$$E_h = \|var_h - var_{\text{exc}}\|_2 \quad (11a)$$

when the exact solution  $var_{\text{exc}}$  is available, or [16]

$$\widetilde{E}_h = \|var_h - var_{h/2}\|_2 \quad (11b)$$

otherwise, where  $var_{h/2}$  is the numerical solution computed on a mesh with grid size  $h/2$ . The derivatives, which are  $u_{h,x}$  and  $u_{h,xx}$  in the standard FEM and only  $u_{h,xx}$  in the mixed FEM, are computed in the classical finite element manner, e.g.  $u_{h,x}^{(p-1)} = \sum_{i=1}^m u_i \varphi_{i,x}^{(p)}$  yields an approximation to  $u_x$  using standard FEM. Note that, each differentiation decreases the element degree by one.

### 2.3.3. Convergence of the solution

When the number of DoFs is relatively large, but the round-off error does not exceed the truncation error, the discretization error converges at a fixed rate  $\beta_T$  theoretically[5, Theorem 5.I]. The value of  $\beta_T$  is one order higher than the element degree, and decreases one for each differentiation. In practice, it can be calculated from either

$$\beta_T = \log_2 \left( \frac{E_h}{E_{h/2}} \right) \quad (12a)$$

using Eq. (11a), or

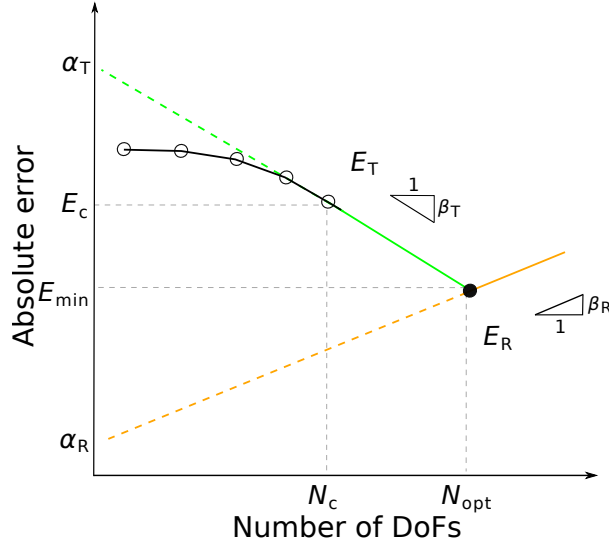
$$\beta_T = \log_2 \left( \frac{\widetilde{E_h}}{\widetilde{E_{h/2}}} \right) \quad (12b)$$

using Eq. (11b).

### 3. General behaviour of the discretization error and approach to predict the highest attainable accuracy

In this section, based on [7, 17], we illustrate the general behaviour of the discretization error  $E_h$  for Eq. (1) as a function of the number of DoFs  $N_h$ , and provide an approach to predict the highest attainable accuracy.

The discretization error for one variable using one  $p$  is illustrated in Fig. 1, where log-log axes are used.



**Fig. 1.** Conceptual sketch of the dependency of the discretization error on the number of DoFs.

As can be seen, the change of  $E_h$  with  $N_h$  can be divided into three phases according to  $N_c$  and  $N_{opt}$ , for which the former is  $N_h$  where  $E_h$  begins showing the expected asymptotic convergence behavior, and the latter is  $N_h$  where  $E_h$  begins increasing, i.e. where the highest attainable accuracy is obtained. The features of  $E_h$  in each phase are shown in Table 1.

**Table 1** Features of  $E_h$  in different phases.

	1. $N_h < N_c$	2. $N_c \leq N_h < N_{opt}$	3. $N_{opt} \leq N_h$
Description	Decreasing but not converging at slope $\beta_T$	Decreasing and converging at slope $\beta_T$ , with the offset $\alpha_T$	Increasing and converging at slope $\beta_R$ , with the offset $\alpha_R$
Formula	-	$E_h = \alpha_T N_h^{-\beta_T}$	$E_h = \alpha_R N_h^{\beta_R}$
Dominant error	Truncation error		Round-off error

As we will prove in section 4, the values of  $\alpha_R$  and  $\beta_R$  can be relatively fixed. Therefore, the round-off



error can be assessed before solving the problem. Moreover, since  $\alpha_T$  can be inverted by using

$$\alpha_T = E_c / N_c^{-\beta_T}, \quad (13)$$

at the beginning of phase 2, where  $E_c$  is the value of  $E_h$  corresponding to  $N_c$ , we can forecast  $E_T$  afterwards.

Obviously,  $N_{\text{opt}}$  happens when  $E_T + E_R$  is the smallest. By solving

$$\frac{d(E_T + E_R)}{dN} = 0, \quad (14)$$

we can predict

$$N_{\text{opt}} = \left( \frac{\alpha_T \beta_T}{\alpha_R \beta_R} \right)^{\frac{1}{\beta_T + \beta_R}}, \quad (15a)$$

and hence, the highest attainable accuracy

$$E_{\min} = \alpha_T N_{\text{opt}}^{-\beta_T} + \alpha_R N_{\text{opt}}^{\beta_R}. \quad (15b)$$

#### 4. Numerical quantification of the round-off error

In this section, we assess the general values for  $\alpha_R$  and  $\beta_R$  for variables  $u$ ,  $u_x$  and  $u_{xx}$ , using both the standard FEM and the mixed FEM. We start with the preliminary results obtained from three benchmark equations, see Table 2, and then investigate the following factors: solution strategy, order of magnitude and boundary condition.

**Table 2** Settings of the benchmark Poisson, diffusion and Helmholtz equations.

	“Poisson”	“diffusion”	“Helmholtz”
$d(x)$	1	$1 + x$	$(1 + i)e^{-x}$
$r(x)$	0	0	$2e^{-x}$
$f(x)$	$-e^{-(x-1/2)^2} (4x^2 - 4x - 1)$	$-2\pi \cos(2\pi x) + 4\pi^2 \sin(2\pi x)(x + 1)$	0
$\ f(x)\ _2$	1.60	42.99	0.00
Boundary conditions	$u(0) = e^{-1/4}$	$u(0) = 0$	$u(0) = 1$
	$u(1) = e^{-1/4}$	$u_x(1) = 2\pi$	$u_x(1) = 0$
Analytical solution $u_{\text{exc}}$	$e^{-(x-1/2)^2}$	$\sin(2\pi x)$	$ae^{(1+i)x} + (1-a)e^{-ix},$ $a = 1/((1-i)e^{1+2i} + 1)$
$\ u_{\text{exc}}\ _2$	0.92	0.71	1.26

##### 4.1. Preliminary results

Element degrees  $p$  range from 1 to 5.

#### 4.1.1. Benchmark equations

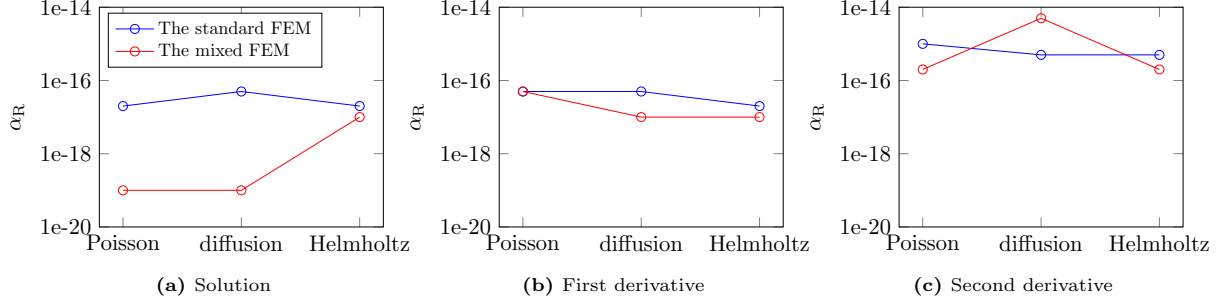
The evolution of the order of convergence is shown in Table 3, where only  $p = 2$  is considered. It shows that the theoretical order of convergence  $\beta_T$  is reached within 5 refinements.

The offsets  $\alpha_R$  for the benchmark equations are shown in Fig. 2, and the slopes  $\beta_R$  using the standard FEM is 2, and that using the mixed FEM is 1. These values are independent of the element degree. Notably, the values of the former are of order  $10^{-16}$ , which is as expected when using double precision. It also shows that  $\alpha_R$  tends to increase slightly with increasing order of derivative.

**Table 3** Evolution of order of convergence.

# of refinements	Solution		
	Poisson	diffusion	Helmholtz
1	-	-	-
2	2.96(2.95)	0.92(1.15)	3.01(3.03)
3	2.99(2.99)	2.96(2.96)	3.00(3.01)
4	3.00(3.00)	2.99(2.99)	3.00(3.01)
5	3.00(3.00)	3.00(3.00)	3.00(3.00)
# of refinements	First derivative		
	Poisson	diffusion	Helmholtz
1	-	-	-
2	2.97(2.96)	3.35(3.28)	2.73(2.75)
3	2.99(2.99)	2.97(3.00)	2.98(2.99)
4	3.00(3.00)	2.99(3.00)	3.00(3.00)
5	3.00(3.00)	3.00(3.00)	3.00(3.00)
# of refinements	Second derivative		
	Poisson	diffusion	Helmholtz
1	-	-	-
2	3.01(3.00)	0.46(2.29)	2.98(2.98)
3	3.00(3.00)	2.98(2.95)	3.00(3.00)
4	3.00(3.00)	3.00(2.99)	3.00(3.00)
5	3.00(3.00)	3.00(3.00)	3.00(3.00)

Summarizing this section, for the two FEM methods and different equation types,  $\beta_T$  can be quickly reached,  $\alpha_R$  tend to be fixed,  $\beta_R$  is only dependent on the FEM method. In what follows, we will take  $\beta_R$  as constant if not stated otherwise.



**Fig. 2.**  $\alpha_R$  for the benchmark equations.

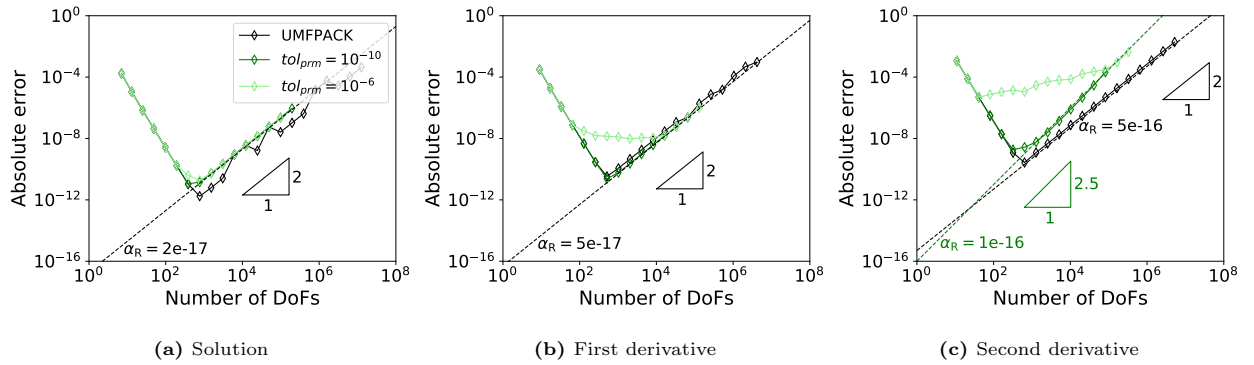
#### 4.2. Sensitivity analysis

We focus on the benchmark Poisson equation, and the approximation order for each variable is 3.

##### 4.2.1. Solution strategy

The alternative solution method is the iterative Conjugate Gradient (CG) method [19], which is applied when the left-hand side is symmetric and positive definite. In the standard FEM, it can be applied directly, while in the mixed FEM, since the left-hand side of Eq. (10) is indefinite, it is applied after segregating Eq. (10) based on the Schur complement. The tolerance of the CG solver is set to be the product of a parameter, denoted by  $tol_{prm}$ , and the  $L_2$  norm of the discrete right-hand side.

*The standard FEM.* The comparison of the absolute errors using the two solution approaches are made in Fig. 3, in which  $tol_{prm} = 10^{-10}$  and  $10^{-6}$  are investigated for the CG solver.



**Fig. 3.** Comparison of the errors using the CG solver and the UMFPACK solver.

When  $tol_{prm}$  is adequately small, i.e.  $tol_{prm} = 10^{-10}$ , the CG solver produces basically the same error with the UMFPACK solver except for the second derivative, for which the round-off error increases faster when using the CG solver. When  $tol_{prm}$  is too large, i.e.  $tol_{prm} = 10^{-6}$ , the error contribution due to the iterative solver dominates the discretization error. Furthermore, it is also found that the errors of higher-order elements are more easier to be affected by larger  $tol_{prm}$ .

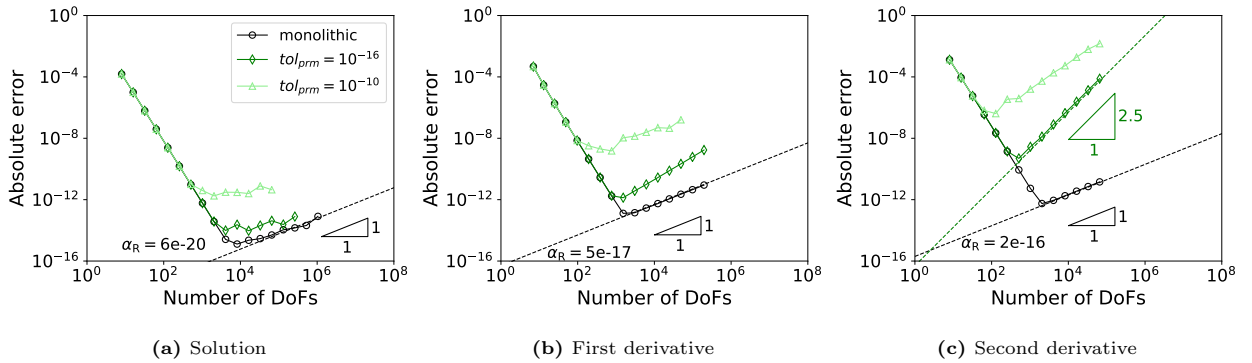
The mixed FEM. The resulting system of equations after segregating Eq. (10) reads

$$B^\top M^{-1} B U = B^\top M^{-1} G - H, \quad (16a)$$

$$M V = G - B U, \quad (16b)$$

for which Eq. (16a) is solved in the first place to obtain  $U$ , which is then substituted into Eq. (16b) to obtain  $V$ .

We first investigate the influence of  $tol_{prm}$  of the CG solver on the solution accuracy when the left-hand side is  $B^\top M^{-1} B$  (Schur complement). In this case, the UMFPACK solver is used to solve the system of equations for the left-hand side being  $M$ . For  $tol_{prm}$  being  $10^{-16}$  and  $10^{-10}$ , the errors are shown in Fig. 4, in comparison with that obtained from solving the monolithic Eq. (10) directly. It shows that the monolithic solution approach yields by far the most accurate solution and derivative values. Remarkably, the round-off error for  $v_x$  increases fastest using the Schur complement approach even though  $tol_{prm}$  is sufficiently small, i.e.  $tol_{prm} = 10^{-16}$ . When  $tol_{prm}$  is less strict, i.e.  $tol_{prm} = 10^{-10}$ , the iteration error dominates the discretization error before the round-off error.



**Fig. 4.** Influence of the CG solver on the solution accuracy using the mixed FEM when the left-hand side is the Schur complement.

Next, we investigate the influence of  $tol_{prm}$  of the CG solver when the left-hand side is  $M$ . In this case, the CG solver with  $tol_{prm}$  being  $10^{-16}$  is used to solve the system of equations with the left-hand side being  $B^\top M^{-1} B$ . For the same tolerances, the errors are basically the same with that shown in Fig. 4.

In summary, when  $tol_{prm}$  is strict enough, for the standard FEM, the CG solver gives the same accuracy for  $u$  and  $u_x$  as the UMFPACK solver, while the UMFPACK solver produces higher accuracy for  $u_{xx}$ ; for the mixed FEM, the accuracy for all the three variables is the highest when using the monolithic approach. When  $tol_{prm}$  is less strict, the applications of the CG solver on both the standard and mixed FEM methods introduce iteration errors.

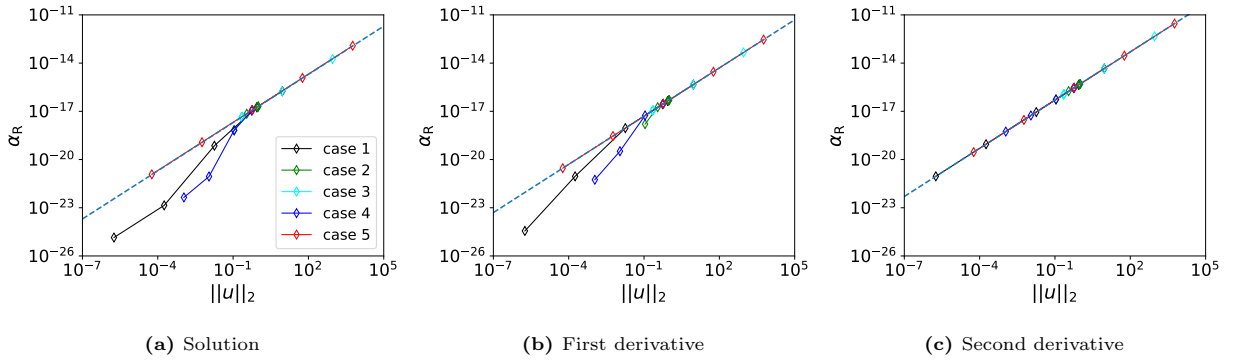
#### 4.2.2. Order of magnitude

For the benchmark equations,  $\|u_{\text{exc}}\|_2$  is of order 1. In this section, we investigate the influence of  $\|u_{\text{exc}}\|_2$  on the offset  $\alpha_R$ , and also  $\|u_{x,\text{exc}}\|_2$  when using the mixed FEM. To cover a wide range of scenarios for both  $\|u_{\text{exc}}\|_2$  and  $\|f\|_2$ , we choose 5 cases shown in Table 4. Each case contains a coefficient  $c_i$ ,  $i = 1, 2, \dots, 5$ , which is varied over several orders of magnitude. Moreover, Dirichlet boundary conditions are imposed at both ends.

**Table 4** Setting of the Poisson equation with different right-hand sides.

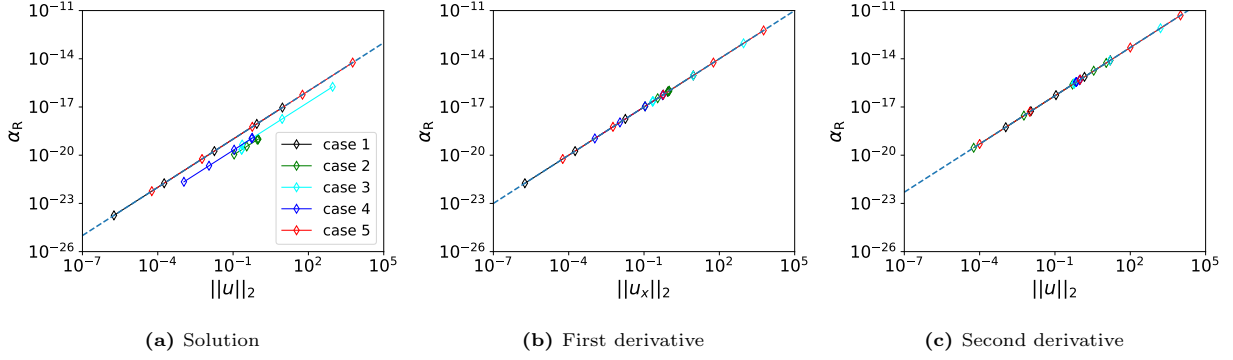
Case	$f(x)$	$u_{\text{exc}}(x)$
1	$\sin(2\pi c_1 x)$	$(2\pi c_1)^{-2} \sin(2\pi c_1 x)$
2	$-e^{-c_2(x-1/2)^2} \cdot (4c_2^2(x-1/2)^2 - 2c_2)$	$e^{-c_2(x-1/2)^2}$
3	$\sin(2\pi c_3 x) + 1$	$(2\pi c_3)^{-2} \sin(2\pi c_3 x) - \frac{x^2}{2}$
4	$2\pi c_4 \sin(2\pi c_4 x)$	$(2\pi c_4)^{-1} \sin(2\pi c_4 x)$
5	0	$c_5^{-1} x$

For case 1, the results are given below, and that of other cases can be found in the data repository, which shows qualitatively the same behavior.



**Fig. 5.** Summary of the offsets when using the standard FEM.

*The standard FEM.* This is because the number of accurate significant digits that the double-precision floating-point format can hold is 17 at most, and hence, more significant digits in the fractional part will be rounded with increasing  $\|u\|_2$ . To eliminate this influence factor, we scale the  $L_2$  norm of  $u$  to 1, which is achieved by dividing the right-hand side  $F$  of the linear system of equations (6) by  $\|u\|_2$ . The scaling scheme can be found in the second row of Table 5, which is denoted as  $S$ . Note that, the scaling factor is approximated from the numerical solution through an a posteriori algorithm presented in Section 5.



**Fig. 6.** Summary of the offsets when using the standard FEM.

Using scheme  $S$  for Case 1, the absolute errors are depicted in the data repository. It shows that  $\alpha_R$  for different  $c_1$  converge to common values, which are  $2 \times 10^{-17}$ ,  $5 \times 10^{-17}$  and  $5 \times 10^{-16}$  for  $u$ ,  $u_x$  and  $u_{xx}$ , respectively. These values also apply to Cases 2~5 when using scheme  $S$ .

**Table 5** Scaling schemes.

Scheme	Left-hand side		Solution	Right-hand side	
$S$	$A$		$\frac{1}{\ u\ _2} U$	$\frac{1}{\ u\ _2} F$	
$M_1$	$M$	$\frac{\ u\ _2}{\ v\ _2} B$	$\frac{1}{\ v\ _2} V$	$\frac{1}{\ v\ _2}$	$G$
	$B^T$	$0$	$\frac{1}{\ u\ _2} U$		$H$
$M_2$	$M$	$B$	$\frac{1}{\ u\ _2}$	$\frac{1}{\ u\ _2}$	$G$
	$B^T$	$0$	$U$		$H$

*The mixed FEM.* The outcome of the numerical experiments performed with the mixed-FEM formulation Eq. (10) are presented in the data repository. Like with the standard FEM, the offsets  $\alpha_R$  for  $u$  and  $u_x$  increase whenever  $\|u\|_2$  and  $\|u_x\|_2$  are increased.

Instinctively, to mitigate the influence of the magnitude of the solution  $u$  and the first derivative  $v$  on the offset  $\alpha_R$ , one would scale the  $L_2$  norm of  $u$  and  $v$  to 1. This can be achieved by dividing the right-hand sides  $G$  and  $H$  by the  $L_2$  norm of the first derivative  $\|v\|_2$  and multiplying the discrete first derivative operator  $B$  by  $\frac{\|u\|_2}{\|v\|_2}$ , see scheme  $M_1$  shown in Table 5. Using this scheme, the absolute errors of  $u$ ,  $v$  and  $v_x$  are shown in the data repository. As expected, the offsets  $\alpha_R$  for  $u$  and  $v_x$  converge, but that for  $v$  only converge when  $c_1 < 1$ . For  $c_1 > 1$ , no convergence of  $\alpha_R$  is seen for  $v$ . It further indicates that we need smaller scaling factor for  $v$  when  $c_1 > 1$ .

Given that  $\|u\|_2$  is of the same order with  $\|v\|_2$  when  $c_1 < 1$ , while it is smaller than  $\|v\|_2$  when  $c_1 > 1$ , we scale both  $u$  and  $v$  by  $\|u\|_2$ . This scaling scheme, which divides both the right-hand sides  $G$  and  $H$  by  $\|u\|_2$ , is denoted as scheme  $M_2$  as shown in Table 5. The absolute errors obtained by using this scheme are shown in the data repository, where the offsets  $\alpha_R$  for both  $u$  and  $v$  converge. However, not for  $v_x$ .

Therefore, scheme  $M_2$  is preferable if  $u$  and  $v$  are of primary interest, and scheme  $M_1$  is more suitable when  $v_x$  is of interest. If all three quantities need to be computed with required accuracy, both schemes  $M_1$  and  $M_2$  need to be applied side by side. The generalized values of  $\alpha_R$  using the mixed FEM are  $1 \times 10^{-19}$ ,  $5 \times 10^{-17}$  and  $5 \times 10^{-16}$  for  $u$ ,  $v$  and  $v_x$ , respectively. These two scaling schemes also work for Cases 2~5, but the resulting  $\alpha_R$  are slightly different. After being amended by Cases 2~5,  $\alpha_R$  become  $1 \times 10^{-18}$ ,  $1 \times 10^{-16}$  and  $5 \times 10^{-16}$ .

Generalizing  $\alpha_R$  using both the standard FEM and the mixed FEM for the Poisson equations, we obtain  $2.0 \times 10^{-17}$ ,  $5.0 \times 10^{-17}$  and  $5.0 \times 10^{-16}$  for  $u$ ,  $u_x$  and  $u_{xx}$ , respectively. Using the above scaling schemes for the benchmark diffusion and Helmholtz equations, we obtain values  $2.0 \times 10^{-17}$ ,  $2.0 \times 10^{-17}$  and  $1.0 \times 10^{-15}$ . Generalizing these two sets of values, we obtain  $\alpha_R$  shown in Table 6.

**Table 6** Generalized values of  $\alpha_R$  for Eq. (1).

	$u$	$u_x$	$u_{xx}$
$\alpha_R$	2e-17	5e-17	1e-15

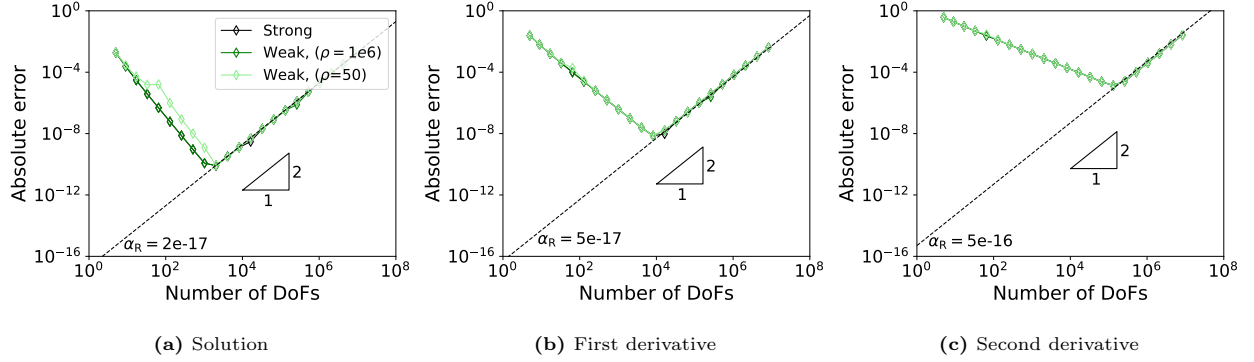
Summarizing this section, to mitigate the influence of the order of magnitude of the different variables on  $\alpha_R$ , we have proposed and validated three different scaling schemes  $S$ ,  $M_1$  and  $M_2$ , resulting in the common values for  $\alpha_R$ . This is an essential prerequisite for our a posteriori refinement strategy to be robust and generally applicable.

#### 4.2.3. Boundary conditions

In this section, two aspects of the influence of the boundary conditions on the round-off error are investigated: first the method of implementing the Dirichlet boundary conditions, and secondly types of boundary conditions.

For the first aspect, using Weak form 2 for  $\rho = 50$  and  $10^6$ , the discretization errors are depicted in Fig. 7, in comparison with that using Weak form 1. As can be seen, both weak and strong imposition of the Dirichlet boundary condition yield the same trend line for the round-off error for the solution and its derivatives, and the magnitude of the penalty parameter in the weak imposition makes no difference. In addition, small penalty parameters might lead to larger truncation errors for  $u$ , but the difference diminishes when the penalty parameter is large enough.

To construct the problem for the second aspect, the Dirichlet boundary condition at the left boundary

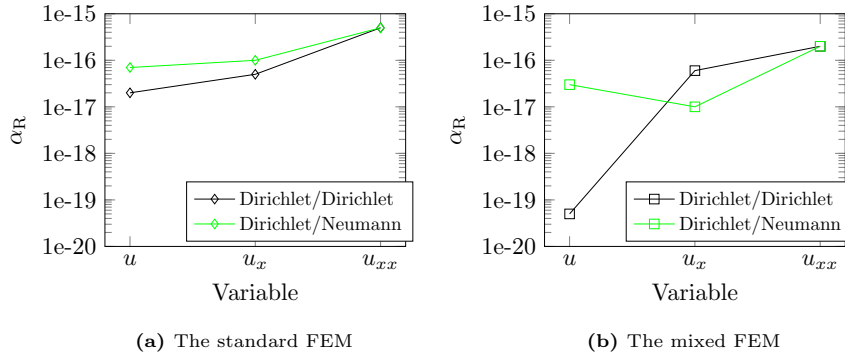


**Fig. 7.** Comparison of the errors for imposing the Dirichlet boundary condition strongly and weakly.

( $x = 0$ ) is kept while the Dirichlet boundary condition at the right boundary ( $x = 1$ ) has been replaced by the Neumann boundary condition  $u_x(1) = -e^{-1/4}$ , leading to the same solution and derivative profiles.

*The standard FEM.* Using the standard FEM, the offsets  $\alpha_R$  for the two types of boundary conditions are depicted in Fig. 8(a). For the Dirichlet/Neumann boundary condition, the offsets  $\alpha_R$  for  $u$  and  $u_x$  are slightly larger than that for the Dirichlet/Dirichlet boundary condition by a factor of 3.5 and 2, respectively. The offsets  $\alpha_R$  for  $u_{xx}$  are identical for the two types of boundary conditions.

*The mixed FEM.* Using the mixed FEM, the offsets  $\alpha_R$  for the two types of boundary conditions are depicted in Fig. 8(b). As can be seen, the type of boundary conditions plays a more important role for  $\alpha_R$  for the solution than  $\alpha_R$  for other variables.



**Fig. 8.** Comparison of the errors for imposing Dirichlet/Dirichlet and Dirichlet/Neumann boundary conditions.

In summary,  $\alpha_R$  are relatively independent of the variations in the type of boundary conditions and the method Dirichlet boundary conditions are implemented, which is an important prerequisite for our a posteriori refinement strategy to be applicable for a wide range of problems.

To conclude the sections on sensitivity analysis, the factors that cannot be mitigated are the tolerances for the iterative linear solver, that can be mitigated are the order of magnitude, and that are relatively



irrelevant are the boundary conditions.

#### 4.2.4. Equation type

### 5. A posteriori algorithm for finding the optimal number of degrees of freedom

Based on the validation experiments from the previous section, we introduce a novel a posteriori algorithm for determining  $E_{\min}$  for the solution and its first and second derivative without performing the brute-force mesh refinement. Table 7 gives the default settings and the required custom input of the algorithm.

**Table 7** Settings of the algorithm.

Item	Default	Custom
Problem	-	<ul style="list-style-type: none"> <li>the differential equation to be solved</li> <li>its associated boundary conditions</li> </ul>
Grid	<ul style="list-style-type: none"> <li>initial number of vertices: 2</li> <li>the vertices are equidistant</li> </ul>	-
FEM	<ul style="list-style-type: none"> <li>the maximum <math>N_h</math>, denoted by <math>N_{\max}</math>, : <math>10^8</math></li> <li>Dirichlet boundary conditions are imposed strongly</li> </ul>	<ul style="list-style-type: none"> <li>standard or mixed formulation</li> <li>an ordered array of element degrees <math>\{p_{\min}, \dots, p_{\max}\}</math></li> </ul>
Computer precision	IEEE-754 double precision	-
Solver	UMFPACK	-
$var$	-	<ul style="list-style-type: none"> <li>chosen from <math>\{u, u_x, u_{xx}\}</math></li> <li>error tolerance <math>tol_{var}</math></li> </ul>

Furthermore, we use the following coefficients in the algorithm:

- a minimal number of  $h$ -refinements before ‘*NORMALIZATION*’ and carrying out ‘*PREDICTION*’, denoted by  $REF_{\min}$ , with the following default values:

$$REF_{\min} = \begin{cases} 9 - p & \text{for } p < 6, \\ 4 & \text{otherwise.} \end{cases} \quad (17)$$

We choose this parameter mainly because the error might increase, or decrease faster than the theoretical order of convergence for coarse refinements, especially for lower-order elements.

- a stopping criterion  $c_s$  for seeking the scaling factor  $\|var_{\text{exc}}\|_2$  in Table 5, its value is 0.001 by default. We choose this parameter because the analytical solution does not exist for most practical problems.

- a relaxation coefficient  $c_r$  for seeking the theoretical order of convergence, with the following default values:

$$c_r = \begin{cases} 0.9 & \text{for } p < 4, \\ 0.7 & \text{for } 4 \leq p < 10, \\ 0.5 & \text{otherwise.} \end{cases} \quad (18)$$

- the offset  $\alpha_R$ , see Table 6 for the default values.

The procedure of our algorithm consists of four steps, which are explained below:

*Step-1. ‘INPUT’.* In this step, the custom input has to be provided.

*Step-2. ‘NORMALIZATION’.* The function of this step is to find the scaling factor to normalize problems of different orders of magnitude for the variable. The specific procedure can be found in Algorithm 1, where elements of degree  $p_{\min}$  are used.

---

**Algorithm 1: NORMALIZATION**

---

```

1 while  $N_h < N_{\max}$  do
2   if  $\left| \frac{\|var_h\|_2 - \|var_{2h}\|_2}{\|var_h\|_2} \right| < c_s$  then
3      $\|var_{\text{exc}}\|_2 \leftarrow \|var_h\|_2$ ;
4     break;
5   else
6      $h \leftarrow h/2$ ;
7     calculate  $\|var_h\|_2$  using Eq. (11a) without scaling;
8   end
9 end
```

---

*Step-3. ‘PREDICTION’.* This step finds  $E_{\min}$  for each  $var$  and  $p$  of interest, as illustrated in Fig. 1. The procedure for carrying out this step can be found in Algorithm 2.

---

**Algorithm 2: PREDICTION**


---

```

1 while  $\widetilde{E}_h > E_R$  and  $N_h < N_{\max}$  do
2    $\widetilde{Q} \leftarrow \log_2 \left( \widetilde{E}_{2h} / \widetilde{E}_h \right);$ 
3   if  $\widetilde{Q} \geq \beta_T \times c_r$  then
4      $N_c \leftarrow N_h;$ 
5      $E_c \leftarrow \widetilde{E}_h;$ 
6      $\alpha_T \leftarrow E_c / N_c^{-\beta_T};$ 
7      $N_{\text{opt}} \leftarrow \left( \frac{\alpha_T \beta_T}{\alpha_R \beta_R} \right)^{\frac{1}{\beta_R + \beta_T}};$ 
8      $E_{\min} \leftarrow \alpha_T N_{\text{opt}}^{-\beta_T} + \alpha_R N_{\text{opt}}^{\beta_R};$ 
9   else
10     $h \leftarrow h/2;$ 
11    calculate  $\widetilde{E}_h$  using Eq. (11b) with proper scaling schemes;
12  end
13 end

```

---

*Step-4. ‘OUTPUT’.* In this step, we output  $E_{\min}$  obtained from *Step-3*.

## 6. Validation

In what follows, we validate the strategy discussed in Section 3 by using the following Helmholtz problem:

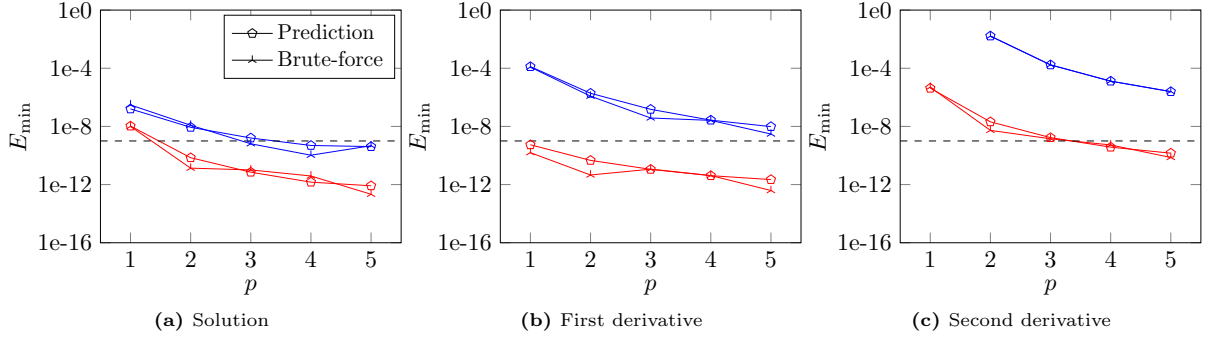
$$((0.01 + x)(1.01 - x)u_x)_x - (0.01i)u(x) = 1.0, \quad x \in I = (0, 1), \quad (19)$$

with homogeneous Dirichlet and Neumann boundary conditions imposed as follows:  $u(0) = 0$  and  $u_x(1) = 0$ .

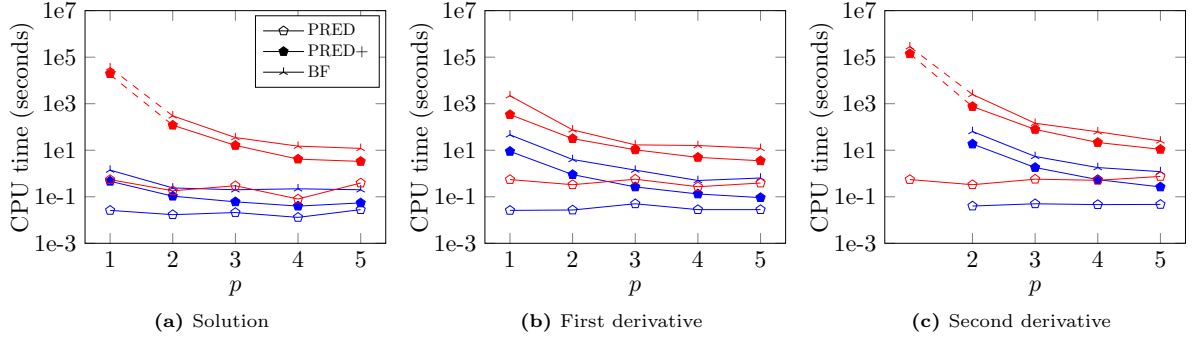
Both the standard FEM and the mixed FEM are investigated, and the element degree  $p$  has a range of  $\{1, 2, \dots, 5\}$ . Variables  $u$ ,  $u_x$  and  $u_{xx}$  are all investigated, for which  $tol_{var}$  is set to be  $10^{-9}$ .

Using the prediction approach and the brute-force approach,  $E_{\min}$  are compared in Fig. 9. As can be seen,  $E_{\min}$  can be predicted correctly.

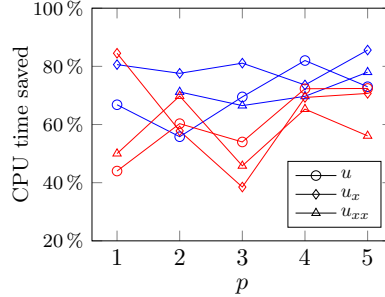
The CPU time required by the prediction approach (PRED) and the brute-force approach (BF) is shown in Fig. 10. Next to time PRED, and the computation time for the optimal grid (PRED+) using the prediction approach is also given. As can be seen, both time BF and time PRED+ decrease with increasing element degree. Time PRED+ is much smaller compared to time BF, see Fig. 11 for the percentage of the CPU time saved by PRED+, which shows a saving of the CPU time basically more than 60% and 40% for the standard FEM and the mixed FEM, respectively. Last but not least, time PRED is negligible compared to time PRED+.



**Fig. 9.** Comparison of  $E_{\min}$  for Eq. (19) using the algorithm and the brute-force refinement. The blue color denotes the standard FEM, and the red color denotes the mixed FEM.



**Fig. 10.** Comparison of the CPU time to obtain  $E_{\min}$  for Eq. (19) using the algorithm and the brute-force refinement. The blue color denotes the standard FEM, and the red color denotes the mixed FEM.



**Fig. 11.** Percentage of CPU time saved using the algorithm. The blue color denotes the standard FEM, and the red color denotes the mixed FEM.

Furthermore, the dashed line indicating the desired error tolerance in Fig. 9 cannot be reached using the standard FEM, whereas it can be reached using the mixed FEM with  $P_4/P_3^{\text{disc}}$  or better. When using  $P_4/P_3^{\text{disc}}$ ,  $N_{\text{opt}}$  for  $u$ ,  $u_x$  and  $u_{xx}$  are predicted to be 6042, 9812 and 123486, respectively.

## 7. Conclusions

A novel approach is presented to predict the highest attainable accuracy for second-order ordinary differential equations using the finite element methods. In contrast to the brute-force approach, which uses successive  $h$ -refinements, this approach uses only a few coarse grid refinements. This approach is viable for the solution and its first and second derivative, for both the standard FEM and the mixed FEM, and different element degrees. The algorithm for implementing the approach shows that the highest attainable accuracy can be accurately predicted and the CPU time is significantly reduced. To compute the solution of the highest attainable accuracy using our approach, the CPU time can be saved more than 60% for the standard FEM and 40% for the mixed FEM.

Future research will focus on the validation of the approach for 2D second-order problems, where the influence of the linear system solver, local mesh refinement and boundary conditions might be significantly different from 1D problems.

## Appendix A. Derivation of the weak form

### Appendix A.1. The standard FEM

Multiply Eq. (1) by a test function  $\eta \in H^1(I)$ , and integrate it over  $I$  yields

$$(\eta, -(du_x)_x + ru) = (\eta, f). \quad (\text{A.1})$$

By applying Gauss's theorem for the first term of the left-hand side of Eq. (A.1), we obtain

$$(\eta_x, du_x) + (\eta, ru) = (\eta, f) + (\eta, du_x n)_{\Gamma_N}. \quad (\text{A.2})$$

Therefore, omitting the boundary conditions, the weak form reads

<p>Find <math>u \in H^1(I)</math> such that:</p> $(\eta_x, du_x) + (\eta, ru) = (\eta, f) + (\eta, du_x n)_{\Gamma_N} \quad \forall \eta \in H^1(I),$ <p>where <math>n</math> is 1 at <math>x = 1</math>, and -1 at <math>x = 0</math>.</p>
---

(A.3)

Imposing the original Dirichlet boundary conditions on  $u$  and the corresponding homogeneous Dirichlet boundary conditions on  $\eta$  in Eq. (A.3), which is called the strong imposition of the Dirichlet boundary conditions, the weak form can be found in Eq. (3). Instead of imposing the Dirichlet boundary conditions directly on the variables  $u$  and  $\eta$  in Eq. (A.3), by adding auxiliary terms, which is called the weak imposition of the Dirichlet boundary conditions, we obtain the weak form Eq. (4).

*Appendix A.2. The mixed FEM*

To obtain the weak form of Eq. (7), Eq. (7a) is multiplied by a test function of  $v$ , i.e.  $w \in H_{N0}^1(I)$ , and integrated over  $I$ , yielding

$$(d^{-1}v + u_x, w) = 0, \quad (\text{A.4a})$$

and Eq. (7b) is multiplied by a test function of  $u$ , i.e.  $q \in L^2(I)$ , and integrated over  $I$ , yielding

$$-(q, v_x) + (q, ru) = (q, f). \quad (\text{A.4b})$$

By applying Gauss's theorem and imposing the natural boundary condition  $u(x) = g(x)$  on  $\Gamma_D$ , Eq. (A.4a) becomes

$$(w, d^{-1}v) - (w_x, u) = -(w, gn)_{\Gamma_D}, \quad (\text{A.4c})$$

which results in Eq. (8a).

## References

- [1] Mohit Kumar, Henk M. Schuttelaars, Pieter C. Roos, and Matthias Möller. Three-dimensional semi-idealized model for tidal motion in tidal estuaries. *Ocean Dynamics*, 66(1):99–118, 2016.
- [2] GF Carey. Derivative calculation from finite element solutions. *Computer Methods in Applied Mechanics and Engineering*, 35(1):1–14, 1982.
- [3] Joel H Ferziger and Milovan Peric. *Computational methods for fluid dynamics*. Springer Science & Business Media, 2012.
- [4] Dan Zuras, Mike Cowlshaw, Alex Aiken, Matthew Applegate, David Bailey, Steve Bass, Dileep Bhandarkar, Mahesh Bhat, David Bindel, Sylvie Boldo, et al. IEEE standard for floating-point arithmetic. *IEEE Std 754-2008*, pages 1–70, 2008.
- [5] Mark S Gockenbach. *Understanding and implementing the finite element method*, volume 97. Siam, 2006.
- [6] B Guo and I Babuška. The hp version of the finite element method. *Computational Mechanics*, 1(1):21–41, 1986.
- [7] Ivo Babuska and Gustaf Söderlind. On roundoff error growth in elliptic problems. *ACM Transactions on Mathematical Software*, 44(3):1–22, 2018.
- [8] Fuyun Ling and J Proakis. Numerical accuracy and stability: Two problems of adaptive estimation algorithms caused by round-off error. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84.*, volume 9, pages 571–574. IEEE, 1984.
- [9] Shan-Cong Mou, Yu-Xuan Luan, Wen-Tao Ji, Jian-Fei Zhang, and Wen-Quan Tao. An example for the effect of round-off errors on numerical heat transfer. *Numerical Heat Transfer, Part B: Fundamentals*, 72(1):21–32, 2017.
- [10] Mark Ainsworth and J Tinsley Oden. A procedure for a posteriori error estimation for hp finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 101(1-3):73–96, 1992.
- [11] DW Kelly, De SR Gago, OC Zienkiewicz, I Babuska, et al. A posteriori error analysis and adaptive processes in the finite element method: Part I – error analysis. *International Journal for Numerical Methods in Engineering*, 19(11):1593–1619, 1983.
- [12] Daniele Boffi, Franco Brezzi, Michel Fortin, et al. *Mixed finite element methods and applications*, volume 44. Springer, 2013.
- [13] Jouni Freund and Rolf Stenberg. On weakly imposed boundary conditions for second order problems. In *Proceedings of the Ninth Int. Conf. Finite Elements in Fluids*, pages 327–336. Venice, 1995.
- [14] Giovanni Alzetta, Daniel Arndt, Wolfgang Bangerth, Vishal Boddur, Benjamin Brands, Denis Davydov, Rene Gassmöller, Timo Heister, Luca Heltai, Katharina Kormann, et al. The deal.II library, version 9.0. *Journal of Numerical Mathematics*, 26(4):173–183, 2018.
- [15] Timothy A Davis. Algorithm 832: UMFPACK V4.3 – an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software (TOMS)*, 30(2):196–199, 2004.
- [16] Olof Runborg. Lecture notes in numerical solutions of differential equations (dn2255): Verifying numerical convergence rates, 2012.
- [17] Meshing considerations for linear static problems. <https://www.comsol.com/blogs/meshing-considerations-linear-static-problems/>. Accessed: 2019-12-9.
- [18] W Kahan. Floating-point tricks to solve boundary-value problems faster. *University of California@ Berkeley*, 2013.
- [19] Theo Ginsburg. The conjugate gradient method. *Numer. Math.*, 5(1):191–200, December 1963.