Turkish Journal of Mathematics

http://journals.tubitak.gov.tr/math/

Research Article

Turk J Math
(2021) 45: 2269 – 2281
© TÜBİTAK
doi:10.3906/mat-2102-18

A performance assessment of an HDG method for second-order Fredholm integro-differential equation: existence-uniqueness and approximation

Mehmet Fatih KARAASLAN*
Department of Statistics, Faculty of Arts and Sciences, Yıldız Technical University, İstanbul, Turkey

Received: 05.02.2021 • **Accepted/Published Online:** 22.08.2021 • **Final Version:** 16.09.2021

Abstract: In this paper, we obtain the existence–uniqueness of solution to the second-order linear Fredholm integro-differential equation (FIDE) with Dirichlet boundary condition by hybridizable discontinuous Galerkin (HDG) method. A key property of these methods is to eliminate all internal degrees of freedom and to construct a linear system that only includes globally coupled unknowns at the element interfaces. After designing and implementing HDG algorithm, we provide some necessary and sufficient conditions based on the stabilization parameter and kernel function to guarantee the existence-uniqueness of the approximate solution. Then, some numerical examples are carried out to assess the performance of the present method. When comparing with existing some methods in literature, the experimental studies verify the reliability and feasibility of the HDG method for the problem under consideration.

Key words: Hybridization, Fredholm integro-differential equation, stabilization parameter, boundary value problem

1. Introduction

The origins of the work about integral and integro-differential equations (IDEs) may be traced to the study of Abel, Lotka, Fredholm, Malthus, Verhulst, and Volterra on problems that contain mechanics, mathematical biology and economics. Outstanding names are associated with the problem of applying these equations to describe leg effects, but one of the first and most prominent figures is Vito Volterra. Integro–differential equations (IDEs) appear naturally when studying discontinuous stochastic processes. These equations arise in many different disciplines such as neutron diffusion, electromagnetic theory, ocean circulations, dispersive waves, electric circuit problems, heat and mass diffusion processes, etc. (see [14, 15] and references therein). It is usually difficult to solve these equations analytically due to the integral of the finite range in the equation. Therefore, it is most natural a growing potential tendency in the study to investigate these problems in the recent years. Many different numerical methods have been tried so far to find the approximate solution of different types of IDEs such as multiscale Galerkin method [6], compact finite difference method [24], differential transform method [8], homotopy perturbation method and sine-cosine wavelet method [12], Taylor polynomial method [1], B-spline collocation method [13], Legendre polynomial method, and variational iteration method [3], Monte Carlo method [10], reproducing kernel Hilbert space method [2], Pell–Lucas collocation method [18], operational matrix method [19], exponential method [20], a Galerkin-like method [21], Bessel collocation method [22], and Bernstein collocation method [23]. FIDEs are a class of IDEs that contain an unknown function appears under

*Correspondence: mfatih@yildiz.edu.tr
2010 *AMS Mathematics Subject Classification:* 45B05, 65L10, 65L60

the sign of integration and derivatives functional arguments of it.

The main purpose of this paper is to investigate the performance of the HDG method for second order FIDEs of the form:

$$u''(x) = g(x) + \int_0^1 \kappa(x,t)u(t)dt, \quad x \in \Omega = [0,1]\,, \tag{1.1}$$

subject to boundary conditions

$$u(x) = u_D(x), \quad x \in \partial\Omega = \{0,1\}, \tag{1.2}$$

where $u : [0,1] \to \mathbb{R},\ g : [0,1] \to \mathbb{R}$ are continuous functions, the kernel function $\kappa(x,t)$ is sufficiently smooth continuous function, and u_D is a boundary value function such that

$$u_D(0) = \mu_1,\ u_D(1) = \mu_2,\ \mu_1, \mu_2 \in \mathbb{R}. \tag{1.3}$$

Similar problems are considered in the literature by Chen et al. [5], Jalius and Abdul–Majid [11], and Xue et al. [16].

The HDG methods were introduced in [7] on the framework of second order elliptic problems. Scientists have successfully applied these methods to a wide range of partial differential equations arising in various areas of science, technology, and engineering. The standard approach of the HDG method is to first compute the local solver independently on each finite element and then assemble the global linear equation system. The outstanding property of the method is that the number of the globally-coupled degrees of freedom defined on the element boundaries is significantly reduced. The approximate solutions can be expressed in an element-by-element fashion in terms of an approximate trace satisfying a global weak formulation. Therefore, they are easy to implement. As an indication of HDG approximation, we obtain existence–uniqueness theory of solution to the problem (1.1)–(1.2) and also confirm this phenomena by condition numbers of the global stiffness matrix that has simetric, positive definite, and tridiagonal structure. The performance of the HDG method is compared with a number of numerical methods to provide the accuracy of proposed method. To the best of our knowledge, this is the first time that HDG methods are used for a second-order FIDE.

This paper is organized as follows. Section 2 provides some necessary notations, approximation spaces, and HDG formulation for FIDE. Section 3 presents existence-uniqueness theorems of the approximate solution. Section 4 briefly explain the realization of the algorithm process. Section 5 gives some numerical examples to obtain a good agreement with the effectiveness of the HDG method.

2. Definition of HDG method

In this section, we introduce some notations, approximate spaces, and define the formulation of the HDG method for problem (1.1)–(1.2).

2.1. Notations and approximate spaces

Let Ω_h denote the partition of $\Omega = [0,1]$ for a given positive integer elements N as follows:

$$\Omega_h := \{I_i = (x_{i-1}, x_i)|\ 0 = x_0 < x_1 < \cdots < x_{N-1} < x_N = 1\}.$$

We define the nodes and interior nodes as $\xi_h := \{x_0, x_1, \ldots, x_N\}$ and ${\xi_h}^0 := \xi_h \backslash \partial\Omega$, respectively. We introduce the set of $h := \max_{K\in\Omega_h}\{h_K\}$ where h_K is the length of K.

Now, we define $P^s(K)$ as the set of all polynomials of degree $s \geq 0$ on K. The piecewise polynomial space on Ω is given as

$$V_h^s := \{v \in L^2(\Omega) : \; v|_K \in P^s(K) \text{ for all } K \in \Omega_h\}.$$

We also introduce

$$L_0^2(\xi_h) := \{z \in L^2(\xi_h) : z = 0 \text{ on } \partial\Omega\},$$

where $L^2(\xi_h)$ is the set of vectors $z = (z_1, z_2, \; . \; . \; . \; , z_{N+1}) \in \mathbb{R}^{N+1}$ such that $\sum\limits_{j=1}^{N+1} {z_j}^2 < \infty$.

2.2. HDG formulation

We now consider second-order FIDE with Dirichlet boundary condition (1.1)-(1.2). The standard basic procedure of the formulation of HDG method is designed. For this purpose, we introduce an auxiliary unknown variable q and an operator L_κ acting on the integral terms of (1.1) as follows:

$$q(x) = -u'(x),$$

$$L_\kappa[u(x)] = \int\limits_0^1 \kappa(x,t)u(t)dt,$$

and rewrite (1.1) as a first-order system of equations as

$$q(x) + u'(x) = 0, \qquad\qquad \text{in } \Omega, \tag{2.1a}$$

$$q'(x) + L_\kappa[u(x)] = -g(x), \text{ in } \Omega, \tag{2.1b}$$

$$u(x) = u_D(x), \text{ on } \partial\Omega. \tag{2.1c}$$

The above formulation provides an advantage for obtaining the existence-uniqueness of the HDG approximation. The next step is to multiplying both sides of (2.1a)–(2.1b) by test functions $(v_1, v_2) \in [V_h^s]^2$ and using integration by parts formula over Ω_h. Then, we implement the HDG method that seeks an approximate solution $(q_h, u_h, \hat{u}_h)$ to the exact solution $(q, u, u|_{\varepsilon_h})$ in the finite dimensional space $[V_h^s]^2 \text{x} L^2(\xi_h)$ such that

$$(q_h, v_1)_{\Omega_h} - (u_h, v_1')_{\Omega_h} + < v_1, \hat{u}_h n >_{\partial\Omega_h} = 0, \tag{2.2a}$$

$$-(v_2', q_h)_{\Omega_h} + < v_2, \hat{q}_h n >_{\partial\Omega_h} + (L_\kappa[u_h], v_2)_{\Omega_h} = -(g, v_2)_{\Omega_h}, \tag{2.2b}$$

$$< \hat{q}_h n, \mu >_{\partial\Omega_h} = 0. \tag{2.2c}$$

The last equation is so-called conservativity condition for $\mu \in V_h^s$. The numerical trace $\hat{q}_h$ is expressed in terms of the approximate solution as follows:

$$\hat{q}_h = q_h + \tau(u_h - \hat{u}_h)n, \tag{2.2d}$$

where τ is a given stabilization function throughout the interval. The outward unit normal vectors are given as $n(x^{\mp}) := \pm 1$ for $x \in \xi_h$. The boundary condition (2.1c) is rewritten as

$$\hat{u}_h = u_D \text{ on } \partial\Omega.$$

We define two inner products used in the weak formulation (2.2) of the HDG method. The boundary inner product is introduced as

$$< u_1, u_2 n >_{\partial\Omega_h} := \sum_{K\in\Omega_h} < u_1, u_2 n >_{\partial K},$$

where $< u_1, u_2 n >_{\partial K} := u_1(x_j^-)u_2(x_j^-) + u_1(x_{j-1}^+)u_2(x_{j-1}^+)$, $u_1(x^\pm) := \lim_{\epsilon\downarrow 0} u_1(x\pm\epsilon)$ for $x \in \xi_h$, and the volume inner product is given as

$$(u_1, u_2)_{\Omega_h} := \sum_{K\in\Omega_h} (u_1, u_2)_K \quad \text{where} \quad (u_1, u_2)_K := \int_K u_1(x) u_2(x)\, dx.$$

This completes the description of the HDG method. We require and use numerical quadrature at each step by HDG method.

2.3. Local solvers

To efficiently implement HDG method, we introduce two local solvers within the local domain $K \in \Omega_h$ (for an arbitrary K). They are called local because they are defined on a single element K and they are called solvers because they form an approximate solution to the problem on K.

The local solvers are defined on the element $K \in \Omega_h$ as mapping

$$(w, g) \in L^2(\partial K) \text{ x } L^2(K) \to ({q_h}^{w,g}, {u_h}^{w,g}) \in [P^s(K)]^2.$$

We are now ready to rewrite the HDG formulation (2.2a), (2.2b), and (2.2d) in terms of the local solvers as follows:

$$({q_h}^{w,g}, v_1)_K - ({u_h}^{w,g}, {v_1}')_K = - < w, v_1.n >_{\partial K}, \tag{2.3a}$$

$$-({v_2}', {q_h}^{w,g})_K + < v_2, \hat{q}_h^{w,g}.n >_{\partial K} + (L_\kappa {u_h}^{w,g}, v_2)_K = -(g, v_2)_K, \tag{2.3b}$$

where $v_1, v_2 \in P^s(K)$ and numerical trace is given as

$$\hat{q}_h^{w,g} = {q_h}^{w,g} + \tau({u_h}^{w,g} - w).n \quad \text{on } \partial K. \tag{2.3c}$$

The motivation behind the choice of numerical trace (2.3c) is as follows:

- $\hat{q}_h^{w,g}$ is an approximation to ${q_h}^{w,g}$ assuming that the term $\tau({u_h}^{w,g} - w).n$ is small.
- Since ${u_h}^{w,g}$ approximates the function u on K, and that u satisfies the boundary condition (BC) $u = w$ on ∂K, we expect ${u_h}^{w,g}$ to be close to w on ∂K. The term $\tau({u_h}^{w,g} - w)$ weakly enforces the BC $u = w$ on ∂K.

We then define a new variable $\lambda_h \in L_0^2(\xi_h)$ as

$$\lambda_h := \begin{cases} \hat{u}_h, & \text{on } \partial\Omega_h \backslash \partial\Omega, \\ 0, & \text{on } \partial\Omega, \end{cases}$$

where $\hat{u}_h = \lambda_h + u_D$. Next, we present a characterization of the HDG approximation with respect to the local solvers.

Theorem 2.1 *The approximate solution* $(q_h, u_h) \in [V_h^s]^2$ *obtained via the HDG method can be written in terms of the local solvers as*

$$(q_h, u_h) = ({q_h}^{\lambda_h, u_D, g}, {u_h}^{\lambda_h, u_D, g}), \tag{2.4}$$

where the equation that determines $\lambda_h \in L_0^2(\varepsilon_h)$ *in* (2.2c) *is written as*

$$\left\langle \hat{q}_h^{\lambda_h, u_D, g} n, \mu \right\rangle_{\partial\Omega_h} = 0, \forall \mu \in L_0^2(\varepsilon_h). \tag{2.5}$$

We assemble the global system based on (2.5) *as*

$$a_h(\lambda_h, \mu) = b_h(\mu), \tag{2.6}$$

where

$$a_h(\lambda_h, \mu) = - < \hat{q}_h^{\lambda_h, 0, 0} n, \mu >_{\partial\Omega_h}, \tag{2.7a}$$

$$b_h(\mu) =< \hat{q}_h^{0, u_D, g} n, \mu >_{\partial\Omega_h} . \tag{2.7b}$$

Proof The proof of the theorem is bulky and nothing different from the proof of the corresponding theorem in [4, 7]. □

Theorem (2.1) (the characterization theorem) explains, first of all, how to implement the HDG methods, and secondly, why these methods are computationally effective. More specifically, this is the theorem that allows the elimination of interior degrees of freedom (DOF) which in turn gives rise to a much smaller and sparse linear system. Hence, computing λ_h means that we have computed q_h and u_h. This is known as the hybridization process, and is precisely why, the HDG methods get their name.

3. Existence-uniqueness theorem

Lemma 3.1 *Consider the HDG method defined by the weak formulation* (2.3)*. Then, the local solvers* $({q_h}^{w,g}, {u_h}^{w,g})$ *exist and are unique if and only if*

1. $\kappa(x, t) \geq 0$ *for* $(x, t) \in \Omega$*,*

2. $\tau > 0$ *on* ∂K*.*

Proof Using integrating by parts for (2.3b), we rewrite Equations (2.3a)–(2.3b) for $v_1, v_2 \in P^s(K)$ as follows:

$$({q_h}^{w,g}, v_1)_K - ({u_h}^{w,g}, {v_1}')_K = - < w, v_1.n>_{\partial K}, \tag{3.1a}$$

$$(v_2, ({q_h}^{w,g})')_K + < v_2, \tau {u_h}^{w,g}>_{\partial K} + (L_\kappa {u_h}^{w,g}, v_2)_K = -(g, v_2)_K + < v_2, \tau w>_{\partial K}. \tag{3.1b}$$

We will show that (3.1) has the only solution for $w = 0$ and $g = 0$ such that ${q_h}^{w,g} = 0$ and ${u_h}^{w,g} = 0$. Equation (3.1) is defined as a square system with linearity and finite dimensionality. By taking $v_1 = {q_h}^{w,g}$ and $v_2 = {u_h}^{w,g}$ and adding two equations, we obtain

$$({q_h}^{w,g}, {q_h}^{w,g})_K + < \tau {u_h}^{w,g}, {u_h}^{w,g}>_{\partial K} + \left(\int_0^1 \kappa(x, t)\ {u_h}^{w,g} dt, {u_h}^{w,g} \right)_K = 0. \tag{3.2}$$

This equation implies that ${q_h}^{w,g} = 0$ over the simplex K and ${u_h}^{w,g} = 0$ on ∂K since we assume that $\kappa(x,t) \geq 0$ for $(x,t) \in \Omega$ and $\tau > 0$ on ∂K. Thus, we obtain from (3.1a) that

$$(({u_h}^{w,g})', v_1)_K = 0, v_1 \in P^s(K). \tag{3.3}$$

As a consequence, ${u_h}^{w,g} = 0$ on K since ${u_h}^{w,g} = 0$ on ∂K. This completes the proof. □

Thus, we require conditions in Lemma 3.1 obtaining the existence-uniqueness and also stability of the approximate solution.

3.1. A condition for existence-uniqueness of λ_h

Next, we prove that Eq. (2.6) determines λ_h and, by (2.4) and Lemma (3.1), q_h and u_h.

Lemma 3.2 *We have, for all $\eta, m \in L^2(\xi_h)$, that*

$$a_h(\eta, m) =< {q_h}^{m,0}, {q_h}^{\eta,0} >_{\partial\Omega_h} + (L_\kappa {u_h}^{\eta,0}, {u_h}^{m,0})_{\Omega_h} + < {u_h}^{m,0} - m, \tau({u_h}^{\eta,0} - \eta) >_{\partial\Omega_h}. \tag{3.4}$$

Proof To prove the above result, we rewrite Equations (2.3a)–(2.3b) with $g = 0$ as

$$({q_h}^{m,0}, v_1)_{\Omega_h} - ({u_h}^{m,0}, {v_1}')_{\Omega_h} = - < m, v_1 n >_{\partial\Omega_h} \tag{3.5a}$$

$$(({q_h}^{\eta,0})', v_2)_{\Omega_h} + < v_2, \tau({u_h}^{\eta,0} - \eta) >_{\partial\Omega_h} + (L_\kappa {u_h}^{\eta,0}, v_2)_{\Omega_h} = 0 \tag{3.5b}$$

for all $(v_1, v_2) \in P^s(K)$ x $P^s(K)$. The first equation is obtained from (2.3a) by simply replacing g by 0 and summing over the elements. The second is obtained from (2.3b) by replacing m by η, g by 0, integrating by parts, summing the resulting equations over all elements and inserting the definition of the numerical trace (2.3c).

Thus we have

$$a_h(\eta, m) = - < m, \hat{q}_h^{\eta,0} n >_{\partial\Omega_h} = - < m, {q_h}^{\eta,0} n >_{\partial\Omega_h} - < m, \tau({u_h}^{\eta,0} - \eta) >_{\partial\Omega_h} \tag{3.6}$$

by (2.3c). Rearranging terms, we get by (3.5a) with $v_1 = {q_h}^{\eta,0}$,

$$a_h(\eta, m) = ({q_h}^{m,0}, {q_h}^{\eta,0})_{\Omega_h} - ({u_h}^{m,0}, ({q_h}^{\eta,0})')_{\Omega_h} - < m, \tau({u_h}^{\eta,0} - \eta) >_{\partial\Omega_h}. \tag{3.7}$$

Finally, by (3.5b) with $v_2 = {u_h}^{m,0}$, we obtain

$$a_h(\eta, m) =< {q_h}^{m,0}, {q_h}^{\eta,0} >_{\partial\Omega_h} + (L_\kappa {u_h}^{\eta,0}, {u_h}^{m,0})_{\Omega_h} + < {u_h}^{m,0} - m, \tau({u_h}^{\eta,0} - \eta) >_{\partial\Omega_h}. \tag{3.8}$$

This completes the proof. □

Theorem 3.3 *Assume that all conditions of Lemma (3.1) are satisfied. Then, the solution λ_h of the variational formulation (2.6) exists and is unique.*

Proof The existence and uniqueness of λ_h follows if we show that the only solution $\eta \in L_0^2(\varepsilon_h)$ of the problem

$$a_h(\eta, m) = 0, \ \forall m \in L_0^2(\varepsilon_h),$$

is $\eta = 0$. Taking $m = \eta$, by the previous lemma, we then have that

$$0 =< {q_h}^{\eta,0}, {q_h}^{\eta,0} >_{\partial\Omega_h} + \left(\int_0^1 \kappa(x,t)\ {u_h}^{\eta,0} dt, {u_h}^{\eta,0} \right)_{\Omega_h} + < {u_h}^{\eta,0} - \eta, \tau({u_h}^{\eta,0} - \eta >_{\partial\Omega_h}. \tag{3.9}$$

As a consequence, we conclude that ${q_h}^{\eta,0} = 0$ and $u_h^{\eta,0} = \eta$. After a simple integration by parts, Eq. (3.5a), with replaced by η , we obtain

$$(({u_h}^{\eta,0})', v_1)_{\Omega_h} = 0,\ \forall v_1 \in P^s(K).$$

This implies that ${u_h}^{\eta,0}$ is a constant on Ω. η is also a constant on Ω_h and, since $\eta \in M_h^0$, we must have that $\eta = 0$ on Ω_h. This completes the proof. □

4. Algorithm steps

- Compute the first local solver $({q_h}^g,\ {u_h}^g)$ that we only calculate for the given and fixed function g. The first local solver is written as

$$ {q_h}^g(x) = \sum_{j=1}^{s+1} q_j^K(x) Q_j^K(x), {u_h}^g(x) = \sum_{j=1}^{s+1} u_j^K(x) Q_j^K(x)$$

  , where $q_j^K(x)$, $u_j^K(x)$ are coefficients, and Q_j^K is the monomial basis polynomial. Thus, computing the first local solver $({q_h}^g, {u_h}^g)$ means computing $2(s+1)$ unknown coefficients. We write a loop in which we compute and store the first local solver $({q_h}^g, {u_h}^g)$ as follows:

```
for $K = 1 : N$
Compute DOF of ${q_h}^g$ and ${u_h}^g$ on the $K^{th}$ element
Store DOF of ${q_h}^g$ in the $K^{th}$ column of ${q_h}^g$
Store DOF of ${u_h}^g$ in the $K^{th}$ column of ${u_h}^g$
end
```

  This completes the computation of $({q_h}^g, {u_h}^g)$.

- Compute the second local solver $({q_h}^w, {u_h}^w)$. In this computation the situation is a little different. It is computed for the elements I_{l-1} and I_l, respectively and a loop is written as follows:

```
for $j = 1 : N + 1$
Compute the DOF of ${q_h}^{{w_j}^1}, {q_h}^{{w_j}^2}, {u_h}^{{w_j}^1}, {u_h}^{{w_j}^2}$
Store DOF of ${q_h}^{{w_j}^1}$ on the $j^{th}$ column of ${q_h}^{w1}$
Store DOF of ${u_h}^{{w_j}^1}$ on the $j^{th}$ column of ${u_h}^{w1}$
Store DOF of ${q_h}^{{w_j}^2}$ on the $j^{th}$ column of ${q_h}^{w2}$
Store DOF of ${u_h}^{{w_j}^2}$ on the $j^{th}$ column of ${u_h}^{w2}$
end
```

  This completes the planning of the coding of the local solvers which is the fundamental portion of an HDG code.

- Assemble the global linear system whose unknowns are DOF of λ_h. The linear system resulting form of Equation (2.6) is

$$A\lambda_h = b$$

where A is an $(N+1)$x$(N+1)$ matrix, λ_h is the vector of unknowns of size $N+1$, and b is the right hand side vector of known quantities also of size $N+1$. By solving this linear system, we complete the assembly of the global system.

- Construct the approximate solution (q_h, u_h) of the original problem as follows:

$$q_h = {q_h}^{\lambda_h} + {q_h}^{u_D} + {q_h}^{g}$$

$$u_h = {u_h}^{\lambda_h} + {u_h}^{u_D} + {u_h}^{g}$$

Hence, computing λ_h means that we have computed q_h and u_h which are approximations to q and u, respectively.

5. Numerical results

In this section, we evaluate the performance of the HDG method with respect to the results of some numerical examples. In Tables 1-4 and Table 6, for $\varphi = q, u$, we compute and show the error in the approximate solution (q_h, u_h) that is defined as

$$\|\varphi - \varphi_h\|_\infty = \max_{e\in\xi_h} |(\varphi - \varphi_h)(e)|.$$

The convergence order, p_i, at (mesh $=i$) is given as

$$p_i = \frac{\log(\frac{e_\varphi(i)}{e_\varphi(i-1)})}{\log(0.5)},$$

where $e_\varphi(i)$ denote the error with uniform mesh 2^i and it is calculated with the formula

$$\|\varphi - \varphi_h\|_\infty \le Ch^\eta,$$

where the constant C is independent of h and convergence order of the method is given with η. In the following examples, the stability parameter in the definition of the HDG method $\tau = 1$ on $\partial\Omega$. We start to compute with a mesh ($N = 4$) element and refine it up to a mesh ($N = 128$) element. We do not show the numerical results for $N = 4$ mesh since we start displaying the order column after the first refinement of the mesh. It is worth noting that the global linear system is not sparse due to presence of the integral term appearing on $[0, 1]$. The fact is that we do not obtain a convergence order for the numerical traces except of the polynomial function solution. We emphasize an interesting observation that the convergence order is independent of polynomial degree s. This is an unusual and surprising result for HDG method.

Example 5.1 *Consider the following FIDE [8, 9, 11]*

$$u''(x) = e^x - x + \int_0^1 xtu(t)dt,$$

$$u(0) = 1, \;\; u(1) = e,$$

The unique solution of this problem is given by $u(x) = e^x$*. We display the errors and convergence results of the HDG approximation in Table 1. For any given polynomial of degree* $s \geq 1$*, we observe that the convergence orders of* q_h *and* u_h *are equal to* 1 *and the accuracies are at most* 10^{-8} *and* 10^{-9} *for* $k = 6$*, respectively. In Table 2, for* $N = 10$*, we compared our method with three numerical techniques, and the results provided the superiority of HDG method. We used the following abbreviations for the approximation techniques:*

- *CFD2 (GE): Composite Simpson's 1/3 with second-order finite-difference (Gauss elimination) [11]*
- *B-spline: Cubic B-spline collocation method [9],*
- *DiffTrans: Differential transformation method [8].*

Table 1. The convergence result of Example (5.1).

		$\lVert u - u_h \rVert_\infty$		$\lVert q - q_h \rVert_\infty$	
s	N	error	order	error	order
1	8	3.35e-04	1.09	1.38e-03	0.59
	16	1.66e-04	1.01	8.21e-04	0.75
	32	8.28e-05	1.00	4.51e-04	0.86
	64	4.13e-05	1.00	2.37e-04	0.93
	128	2.06e-05	1.00	1.22e-04	0.96
	256	1.03e-05	1.00	6.15e-05	0.98
	512	5.16e-06	1.00	3.10e-05	0.99
2	8	2.59e-05	1.10	1.26e-04	0.81
	16	1.22e-05	1.08	6.48e-05	0.96
	32	5.95e-06	1.04	4.14e-05	0.65
	64	2.93e-06	1.02	2.32e-05	0.83
	128	1.46e-06	1.01	1.23e-05	0.92
	256	7.25e-07	1.01	6.32e-06	0.96
	512	3.62e-07	1.00	3.21e-06	0.98
3	8	6.68e-06	1.02	3.11e-05	0.62
	16	3.28e-06	1.02	1.95e-05	0.68
	32	1.63e-06	1.01	1.10e-05	0.82
	64	8.14e-07	1.00	5.90e-06	0.90
4	8	3.98e-07	1.07	2.32e-06	0.72
	16	1.94e-07	1.04	1.41e-06	0.65
	32	9.48e-08	1.03	9.28e-07	0.67
5	8	7.82e-08	1.11	4.06e-07	0.65
	16	3.84e-08	1.02	2.59e-07	0.65
6	4	8.07e-09	—	4.09e-08	—

Table 2. A numerical comparison between HDG (k=5), CFD2(GE), B-spline, and DiffTrans for Example (5.1) with mesh sizes ($h = 1/N$).

N	**HDG**	**CFD2–GE**	**B-spline**	**DiffTrans**
10	6.17e-08	7.94e-05	2.05e-02	4.85e-04

Example 5.2 *Consider a FIDE [5] as follows:*

$$u''(x) = -\frac{1}{\pi}(\pi^3 sin(\pi x) + 2x + 1) + \int_0^1 (x+t)u(t)dt, \ \Omega = (0,1)$$

$$u(0) = 0, \ u(1) = 0, \ \partial\Omega = \{0,1\}$$

where exact solution is given as $u(x) = sin(\pi x)$. The results diplayed in Table 3 validate a similar convergence order behavior to what we also obtained in Table 1. In Table 4, we conclude that our method provide significiantly better overall estimation performance than fast multiscale Galerkin (FMG) method in [5]. In Table 5, the condition number of the global stiffness matrix is in the order of h^{-1} independent of the polynomial degree s.

Table 3. Convergence result of Example (5.2).

		$\|u-u_h\|_\infty$		$\|q-q_h\|_\infty$	
	s	error	order	error	order
1	8	8.00e-03	1.13	3.42e-02	0.80
	16	3.79e-03	1.08	1.78e-02	0.94
	32	1.84e-03	1.04	9.00e-03	0.99
	64	9.04e-04	1.02	4.50e-03	1.00
	128	4.48e-04	1.01	2.25e-03	1.00
	256	2.23e-04	1.01	1.12e-03	1.00
	512	1.11e-04	1.00	5.62e-04	1.00
2	8	8.31e-04	0.01	6.77e-03	-0.83
	16	5.13e-04	0.69	5.54e-03	0.29
	32	2.89e-04	0.83	3.44e-03	0.69
	64	1.52e-04	0.92	1.91e-03	0.85
	128	7.82e-05	0.96	1.00e-03	0.93
	256	3.96e-05	0.98	5.14e-04	0.96
	512	1.99e-05	0.99	2.60e-04	0.98
	1024	1.00e-05	1.00	1.31e-04	0.99
	2048	5.01e-06	1.00	6.57e-05	1.00
	4096	2.51e-06	1.00	3.29e-05	1.00

Table 4. A numerical comparison ($\|u-u_h\|_\infty$) of two Galerkin method (HDG and FMG) for Example (5.2) with respect to $s=2$ and $N=4096$.

HDG		FMG	
error	order	error	order
2.51e-06	1.00	5.00e-04	1.00

Table 5. Condition numbers of the global stiffness matrix for Example (5.2).

N	$s=1$	$s=2$	$s=3$
8	3.20e+01	3.20e+01	3.20e+01
16	1.10e+02	1.08e+02	1.08e+02
32	4.38e+02	4.33e+02	4.33e+02
64	1.75e+03	1.73e+03	1.73e+03

Example 5.3 *Consider a FIDE as follows:*

$$\begin{aligned} &u''(x) = -\frac{41}{20} - \frac{1}{6}\left(x^2 - x\right) + \int_0^1 (x-t)^2 u(t)dt, \ \Omega = (0,1) \\ &u(0)=0, \ u(1)=0, \ \partial\Omega = \{\mathit{0,1}\} \end{aligned} \tag{5.1}$$

where exact solution is given as $u(x)=x(1-x)$. For $s=2$, what we obtain from Table 6 that the HDG method captures the exact solution with up to machine precision as the exact solution is a second order polynomial. This is the reason why we see errors that are practically zero, and that's why we do not display the numbers in the order column for this example.

Table 6. Convergence result of Example (5.3).

		$\|u-u_h\|_\infty$		$\|q-q_h\|_\infty$	
s	N	error	order	error	order
1	8	5.76e-04	1.07	2.83e-03	0.51
	16	2.86e-04	1.01	1.69e-03	0.75
	32	1.43e-04	1.00	9.24e-04	0.87
	64	7.14e-05	1.00	4.83e-04	0.93
	128	3.57e-05	1.00	1.25e-04	0.98
	256	1.78e-05	1.00	5.76e-04	1.07
	512	8.92e-06	1.00	6.28e-05	0.99
2	4	7.01e-17	—	2.41e-16	—

6. Conclusion

We investigate the performance analysis of HDG method for second–order FIDE with Dirichlet boundary condition. We observe the efficiency and reliability of the HDG approximation by the results of three numerical examples. Empirically, HDG method consistently outperforms other methods (CFD2 (GE), B-spline, DiffTrans) as shown in Example 1. For a specific value of polynomial degree s and N, in Example 2, we report that our method has better results than the Fast Galerkin method. We also show that the HDG method captures the exact solution up to the machine accuracy for a polynomial function in Example 3. Numerical results for different examples display that when polynomials of degree at most $s \geq 1$ is used for all unknowns, the convergence order of HDG approximation (q_h, u_h) is equal to 1. A theoretical work for obtaining proofs of this convergence order is the subject of ongoing work.

References

[1] Akyüz-Daşçıoğlu A, Sezer M. A Taylor polynomial approach for solving the most general linear Fredholm integro-differential-difference equations. Mathematical Methods in the Applied Sciences 2012; 35 (7): 839-844.

[2] Arqub OA, Al-Smadi M, Shawagfeh N. Solving Fredholm integro-differential equations using reproducing kernel Hilbert space method. Applied Mathematics and Computation 2013; 219 (17): 8938-8948.

[3] Bildik N, Konuralp A, Yalçınbaş S. Comparison of Legendre polynomial approximation and variational iteration method for the solutions of general linear Fredholm integro-differential equations. Computers and Mathematics with Applications 2010; 59 (6): 1909-1917.

[4] Celiker F, Cockburn B, Shi K. Hybridizable discontinuous Galerkin methods for Timoshenko beams. Journal of Scientific Computing 2010; 44 (1): 1-37.

[5] Chen J, He M, Huang Y. A fast multiscale Galerkin method for solving second order linear Fredholm integro-differential equation with Dirichlet boundary conditions. Journal of Computational and Applied Mathematics 2020; 364: 112352.

[6] Chen J, Huang Y, Rong H, Wu T, Zeng T. A multiscale Galerkin method for second-order boundary value problems of Fredholm integro-differential equation. Journal of Computational and Applied Mathematics 2015; 290: 633-640.

[7] Cockburn B, Gopalakrishnan J, Lazarov R. Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. SIAM Journal on Numerical Analysis 2009; 47 (2): 1319-1365.

[8] Darania P, Ebadian A. A method for the numerical solution of the integro-differential equations. Applied Mathematics and Computation 2007; 188 (1): 657-668.

[9] Ebrahimi N, Rashidinia J. Spline collocation for Fredholm and Volterra integro-differential equations. International Journal of Mathematical Modelling and Computations 2014; 4 (3) (SUMMER): 289-298.

[10] Farnoosh R, Ebrahimi M. Monte Carlo method for solving Fredholm integral equations of the second kind. Applied Mathematics and Computation 2008; 195 (1): 309-315.

[11] Jalius C, Abdul-Majid Z. Numerical Solution of Second-Order Fredholm Integrodifferential Equations with Boundary Conditions by Quadrature-Difference Method. Journal of Applied Mathematics 2017; 2017.

[12] Kajani MT, Ghasemi M, Babolian E. Comparison between the homotopy perturbation method and the sine-cosine wavelet method for solving linear integro-differential equations. Computers and Mathematics with Applications 2007; 54 (7-8): 1162-1168.

[13] Mahmoodi Z, Rashidinia J, Babolian E. B-Spline collocation method for linear and nonlinear Fredholm and Volterra integro-differential equations. Applicable Analysis 2013; 92 (9): 1787-1802.

[14] Rahman M. Integral equations and their applications. WIT press, 2007.

[15] Wazwaz AM. Linear and nonlinear integral equations. Berlin: Springer, 2011.

[16] Xue Q, Niu J, Yu D, Ran C. An improved reproducing kernel method for Fredholm integro-differential type two-point boundary value problems. International Journal of Computer Mathematics 2018; 95 (5): 1015-1023.

[17] Yulan W, Chaolu T, Jing P. New algorithm for second-order boundary value problems of integro-differential equation. Journal of Computational and Applied Mathematics 2009; 229 (1): 1-6.

[18] Yüzbaşı Ş, Yildirim G. Pell-Lucas collocation method to solve high-order linear Fredholm-Volterra integro-differential equations and residual correction. Turkish Journal of Mathematics 2020; 44 (4): 1065-1091.

[19] Yüzbaşı Ş, Ismailov N. An operational matrix method for solving linear Fredholm Volterra integro-differential equations. Turkish Journal of Mathematics 2018; 42 (1): 243-256.

[20] Yüzbaşı Ş. An exponential method to solve linear Fredholm-Volterra integro-differential equations and residual improvement. Turkish Journal of Mathematics 2018; 42 (5): 2546-2562.

[21] Yüzbaşı Ş, Karaçayır M. A Galerkin-like approach to solve high-order integrodifferential equations with weakly singular kernel. Kuwait Journal of Science 2016; 43(2).

[22] Yüzbaşı Ş. Numerical solutions of system of linear Fredholm–Volterra integro-differential equations by the Bessel collocation method and error estimation. Applied Mathematics and Computation 2015; 250: 320-338.

[23] Yüzbaşı Ş. A collocation method based on Bernstein polynomials to solve nonlinear Fredholm–Volterra integro-differential equations. Applied Mathematics and Computation 2016; 273: 142-154.

[24] Zhao J, Corless RM. Compact finite difference method for integro-differential equations. Applied Mathematics and Computation 2006; 177 (1): 271-288.