Invited Paper

Numerical verification methods for a system of elliptic PDEs, and their software library

Kouta Sekine¹a), Mitsuhiro T. Nakao², and Shin’ichi Oishi³

¹ Faculty of Information Networking for Innovation and Design, Toyo University, 1-7-11 Akabanedai, kita-ku, Tokyo 115-0053, Japan
² Faculty of Science and Engineering, Waseda University, 3-4-1 Okubo, Tokyo 169-8555, Japan
³ Department of Applied Mathematics, Faculty of Science and Engineering, Waseda University, 3-4-1 Okubo, Tokyo 169-8555, Japan

a) k.sekine@computation.jp

Received October 14, 2019; Revised April 26, 2020; Published January 1, 2021

Abstract: Since the numerical verification method for solving boundary value problems for elliptic partial differential equations (PDEs) was first developed in 1988, many methods have been devised. In this paper, existing verification methods are reformulated using a convergence theorem for simplified Newton-like methods in the direct product space $V_h \times V_{\perp}$ of a computable finite-dimensional space $V_h$ and its orthogonal complement space $V_{\perp}$. Additionally, the Verified Computation for PDEs (VCP) library is provided, which is a software library written in the C++ programming language. The VCP library is introduced as a software library for numerical verification methods of solutions to PDEs. Finally, numerical examples are presented using the reformulated verification methods and VCP library.

Key Words: Verified numerical computation, partial differential equation, computer-assisted proof

1. Introduction

Let $\Omega \subset \mathbb{R}^d$, $(d = 1, 2, \cdots)$ be a bounded domain with a Lipschitz boundary. Let $L^p(\Omega)$, $p \in [1, \infty)$ denote the functional space of the $p$-th power Lebesgue integrable functions. For $p = 2$, let us define the inner product $(u, v)_{L^2} := \int_\Omega u(x)v(x)dx$ and its norm $\|u\|_{L^2} := \sqrt{(u, u)_{L^2}}$. The first-order $L^2$ Sobolev space is denoted by $H^1(\Omega)$, $H^1_0(\Omega) := \{u \in H^1(\Omega) \mid u = 0$ on $\partial \Omega\}$ is defined with the inner product $(u, v)_{H^1_0} := (\nabla u, \nabla v)_{L^2}$, and the topological dual of $H^1_0(\Omega)$ is denoted by $H^{-1}(\Omega)$ so that $(H^1_0(\Omega), H^{-1}(\Omega))$ is an adjoint pair with duality product $\langle \cdot, \cdot \rangle_{H^{-1},H^1_0}$. Let $\tilde{V}_h \subset H^1_0(\Omega)$ be a finite-dimensional subspace spanned by the basis $\{\phi_1, \cdots, \phi_n\}$.

Let $X = (L^2(\Omega))^N$ be a direct product space with inner product $(u, v)_X := \sum_{i=1}^N (u_i, v_i)_{L^2}$, $u = (u_1, \cdots, u_N)^T$, $v = (v_1, \cdots, v_N)^T \in X$ and its norm $\|u\|_X := \sqrt{(u, u)_X}$, $u = (u_1, \cdots, u_N)^T \in X$. 

41
Similarly, let $V = (H^1_0(\Omega))^N$ be a direct product space with inner product $(u, v)_V := \sum_{i=1}^{N} (u_i, v_i)_{H^1_0(\Omega)}$, $u = (u_1, \cdots, u_N)^T, v = (v_1, \cdots, v_N)^T \in V$ and its norm $\|u\|_V := \sqrt{(u, u)_V}, u = (u_1, \cdots, u_N)^T \in V$, and $V^*$ denote a direct product space $(H^{-1}(\Omega))^N$ with norm $\|u\|_{V^*} := \sqrt{\sum_{i=1}^{N} \|u_i\|_{H^{-1}}^2}$, $u = (u_1, \cdots, u_N)^T \in V^*$ and duality pairing $(u, v)_{V^*} := \sum_{i=1}^{N} (u_i, v_i)_{H^{-1}, H^1_0}, u = (u_1, \cdots, u_N)^T \in V^*, v = (v_1, \cdots, v_N)^T \in V$. $V_h$ denotes a direct product space $(V_h)^N$.

We define some operators on these direct product spaces using matrix symbols and operations. Operators $\mathcal{A} : V \rightarrow V^*$ and $f : V \rightarrow V^*$ are defined by

\[
\mathcal{A} := \begin{pmatrix} (-\Delta)^{-1} & 0 & \cdots & 0 \\ 0 & (-\Delta)^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (-\Delta)^{-1} \end{pmatrix} \quad \text{and} \quad f(u) := \begin{pmatrix} f_1(u_1, u_2, \cdots, u_N) \\ f_2(u_1, u_2, \cdots, u_N) \\ \vdots \\ f_N(u_1, u_2, \cdots, u_N) \end{pmatrix},
\]

where $-\Delta : H^1_0(\Omega) \rightarrow H^{-1}(\Omega)$ is the weak Laplace operator and $u = (u_1, \cdots, u_N)^T \in V$. From the Lax-Milgram theorem (e.g., [11, 49]), the weak Laplace operator $-\Delta : H^1_0(\Omega) \rightarrow H^{-1}(\Omega)$ has an inverse operator. Thus, the linear operator $\mathcal{A}$ also has an inverse operator

\[
\mathcal{A}^{-1} := \begin{pmatrix} (-\Delta)^{-1} & 0 & \cdots & 0 \\ 0 & (-\Delta)^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (-\Delta)^{-1} \end{pmatrix}.
\]

In this paper, we study an approach to numerically proving the existence of solutions to a system of elliptic problems:

\[
\begin{align*}
-\Delta u_1 &= f_1(u_1, u_2, \cdots, u_N) \quad \text{in} \quad \Omega, \\
-\Delta u_2 &= f_2(u_1, u_2, \cdots, u_N) \quad \text{in} \quad \Omega, \\
& \quad \vdots \\
-\Delta u_N &= f_N(u_1, u_2, \cdots, u_N) \quad \text{in} \quad \Omega, \\
u_1 &= u_2 = \cdots = u_N = 0 \quad \text{on} \quad \partial \Omega,
\end{align*}
\]

where $f_i, i = 1, \cdots, N$ are given nonlinear functions from $V$ to $H^{-1}(\Omega)$ which is assumed to be Fréchet differentiable. Note that Eq. (2) can be transformed into the following fixed point form.

Find $u \in V$ s.t. $\mathcal{A}u = f(u)$.

A computer-assisted proof method for a semi-linear elliptic partial differential equation (PDE) was first developed by the second author in 1988 [24]; this method is called FS-Int. In FS-Int, for $u \in V$, Ritz projection $R_h : V \rightarrow V_h$ is defined by

\[
((I - R_h)u, v_h)_V = 0, \quad v_h \in V_h.
\]

Let $V_\perp := \{ u \in V \mid (u, v_h)_V = 0, \quad v_h \in V_h \}$ be an orthogonal complement of $V_h$. For a given approximate solution $\hat{u} \in V_h$, setting $w := u - \hat{u}, w_h := R_hw$, and $w_\perp := (I - R_h)w$, FS-Int uses the following fixed point formulation:

find $w_h \in V_h, w_\perp \in V_\perp$ satisfying

\[
\begin{align*}
w_h &= R_hA^{-1}(f(w_h + w_\perp + \hat{u}) - A\hat{u}) \\
w_\perp &= (I - R_h)A^{-1}(f(w_h + w_\perp + \hat{u}) - A\hat{u}).
\end{align*}
\]

Moreover, in FS-Int, we prepare candidate sets $W_h \subset V_h$ and $W_\perp \subset V_\perp$ for $w_h$ and $w_\perp$, respectively, and show, using a computer, these sets satisfy the assumption in Schauder’s or Banach’s fixed point theorems.

Furthermore, in 1990, the second author applied Newton’s method to the finite-dimensional part of the fixed point Eq. (3) so that a sufficient condition of the fixed point theorem is easily validated [25];
this method is called FN-Int in [28]. Let \( f'(v) : V \to V^* \) be the Fréchet derivative at \( v \in V \) of nonlinear term \( f(u) \), and let \( \mathcal{F}'[v] \) be a linear operator defined by
\[
\mathcal{F}'[v] := A - f'(v).
\]
Then, for some approximate solution \( \hat{u} \in V_h \) to (2), we can rewrite the finite-dimensional part of (3) as
\[
w_h = (R_h A^{-1} \mathcal{F}'[\hat{u}]|_{V_h})^{-1} R_h A^{-1} (f(w_h + w_\perp + \hat{u}) - A\hat{u} - f'(\hat{u})w_h).
\]
FN-Norm as an improvement of FN-Int was also developed in 2004 [27]. In FN-Norm, the candidate set of the finite-dimensional part was changed from an interval polynomial to a ball with a certain norm bound in the finite-dimensional space. The common features of FS-Int, FN-Int, and FN-Norm are fixed point equations divided into finite and infinite dimensions using the Ritz projection \( R_h \).

In [32, 33], Plum developed a method that uses the infinite-dimensional Newton method (IN method) without using the Ritz projection \( R_h \). This procedure is considered as an application of the Newton–Kantorovich theorem for the IN method. In the verification by using the IN method, it is necessary to evaluate the upper bound of the norm
\[
\|\mathcal{F}'[\hat{u}]^{-1} (A\hat{u} - f(\hat{u}))\|_V.
\]
However, \( \mathcal{F}'[\hat{u}]^{-1} (A\hat{u} - f(\hat{u})) \) is an infinite dimensional operator which we could not directly evaluate using a computer. Therefore, the above norm estimation is decomposed as
\[
\|\mathcal{F}'[\hat{u}]^{-1}\|_{L(V^*,V)} \|A\hat{u} - f(\hat{u})\|_{V^*}.
\]
Moreover, Plum proposed a method for estimating the norm \( \|\mathcal{F}'[\hat{u}]^{-1}\|_{L(V^*,V)} \) via an eigenvalue bound for concerned operator [33]. Because Plum’s method does not use the Ritz projection \( R_h \), the problems on unbounded domains can also be considered (see e.g., [34–36]).

In 1995, the third author proposed a method for computing the norm evaluation of \( \|\mathcal{F}'[\hat{u}]^{-1}\|_{L(V^*,V)} \) using a general projection [30]. Particularly, this method can be applied for the problems not only in a Hilbert space but also in a Banach space without inner products.

In 2005, the second author with collaborators also proposed a method for computing the norm evaluation of \( \|\mathcal{F}'[\hat{u}]^{-1}\|_{L(V^*,V)} \) using the Ritz projection \( R_h \) [26]. The feature of this method is that it is possible to use the Aubin–Nitsche trick by limiting it to a Hilbert space, and as a result, the efficient norm estimation yields an advantage in the successful verification. Since 2005, a great deal of research has been conducted on improving the norm evaluation \( \|\mathcal{F}'[\hat{u}]^{-1}\|_{L(V^*,V)} \) based on methods proposed in 2005 (e.g., [20, 29, 44–46]).

Recently, the present authors proposed a method describing \( \mathcal{F}'[\hat{u}]^{-1} \) by an operator matrix on direct product space \( V_h \times V_\perp \) (see Appendix C). According to this result, in the IN method, we can directly compute the finite-dimensional part without decomposing the norm \( \|\mathcal{F}'[\hat{u}]^{-1}\|_{L(V^*,V)} \), similar to FN methods.

On the system of elliptic partial differential equations, some numerical proof methods have also been developed (e.g., [4–7, 43]). In [43] and [6], they use FN-Norm. And [4, 5, 7] use the radii polynomial approach (see e.g., [13]). Note that the radii polynomial approach is also considered as the finite-dimensional Newton method in the sense that it does not use the infinite dimensional linearized operator \( \mathcal{F}'[\hat{u}]^{-1} \). Thus we can say that there is still no approach for the solution of system of elliptic partial differential equations by using the IN method up to now.

The first aim of this paper is to reformulate FS, FN, and IN methods using a convergence theorem for simplified Newton-like methods in the direct product space \( V_h \times V_\perp \). Namely, this paper gives a consideration in which the Ritz projection methods are classified and organized according to their properties. Additionally, linearized operator \( \tilde{L} \) used in the Newton-like method is provided for each FS, FN, and IN method using operator matrices \( H_{FS}, H_{FN} \), and \( H_{IN} \), respectively. Therefore, we propose a framework that includes the IN method for the solution of system of elliptic partial differential
equations. We also present the FN method and IN method as Corollary 1 and Corollary 2 of a convergence theorem, Theorem 1, for simplified Newton-like methods, respectively. These corollaries are evaluated so that the constants required for the FN method and IN method are the same, that is, these corollaries only require the constants in Definition 1. Therefore, if the constants in Definition 1 are obtained, then both of the FN method and IN method can be applied and evaluated simultaneously. Note that the IN method in this paper avoids the norm estimation $|F'(\hat{u})^{-1}|_{L(V^*, V)}$. In summary, our contributions are

- development of the IN method for systems of partial differential equations as Corollary 2.
- development of simultaneous check algorithm for FN method and IN method as Corollary 1 and Corollary 2.
- development of an algorithm that removes the norm evaluation of $|F'(\hat{u})^{-1}|_{L(V^*, V)}$ in the existing IN method, which is a drawback of the existing IN method.

The second aim of this paper is to introduce a software library. The realization of computer-assisted existence proof of solutions cannot be completed without a computer environment. However, various kinds of techniques are required to estimate the errors that occur in all computations. Additionally, the computer-assisted proof for existence of solutions of PDEs requires numerical accuracy with a reasonable time of computation. For this purpose, the Verified Computation for PDEs (VCP) library is introduced as a software library for the computer-assisted existence proof of solutions to PDEs. The VCP library is a software library developed by the first author in the C++ programming language. In particular, in this paper, we present how to use the matrix class and Legendre basis class of the VCP library. A feature of the matrix class of the VCP library is that it can be integrated with policy-based design, for example,

- high-speed approximate computation by Intel®MKL with double data type [14]
- high precision approximate computation using MPFR [37]
- numerical linear algebra with guaranteed accuracy using the above data type combined with the kv library [17].

Additionally, because the VCP library has extensibility, which is one of the features of policy-based design, it is designed to withstand the computer-assisted proof of PDEs. Moreover, we present the Legendre basis class that generates the matrices and vectors which are necessary to implement the verification procedures based on Corollary 1 (FN method) and Corollary 2 (IN method).

This paper is organized as follows: In Section 2, we present the notation used throughout this paper. In Section 3, we present a convergence theorem for simplified Newton-like methods. In Section 4, we reformulate the FN method and IN method as Corollary 1 and Corollary 2 using direct product space $V_h \times V_\perp$. In Sections 5 and 6, we demonstrate how to calculate the constants in Definition 1 needed for Corollary 1 and Corollary 2. In Section 7, we describe how to use the VCP library. In Section 8, we present numerical examples of computing Corollary 1 and Corollary 2 using the VCP library. In Appendix A, we present the Deuflhard–Heindl–Yamamoto formulation [9, 48] of a convergence theorem for Newton-like methods. In Appendix B, we outline methods for using the infinite-dimensional eigenvalue problems that are not covered in this paper. In Appendix C, we present the theorem that concerns the operator matrix representation of the solution of linear equations on a general Banach space. In Appendix D, we list the matrix class functions in the VCP library.

## 2. Notation

For two Banach spaces $\mathcal{X}$ and $\mathcal{Y}$, the set of bounded linear operators from $\mathcal{X}$ to $\mathcal{Y}$ is denoted by $L(\mathcal{X}, \mathcal{Y})$ with the operator norm $\|T\|_{L(\mathcal{X}, \mathcal{Y})} := \sup\{\|Tu\|_{\mathcal{Y}}/\|u\|_{\mathcal{X}} : u \in \mathcal{X} \setminus \{0\}\}$ for $T \in L(\mathcal{X}, \mathcal{Y})$. When $\mathcal{X} = \mathcal{Y}$, we simply use $L(\mathcal{X})$. Let $T^* : H^1_0(\Omega) \to H^{-1}(\Omega)$ be an adjoint operator of $T \in L(H^1_0(\Omega), H^{-1}(\Omega))$ satisfying $\for…$
\[ \langle Tu, v \rangle_{H^{-1}, H^1_0} = \langle u, T^* v \rangle_{H^1_0, H^{-1}}. \]

Similarly, for \( T \in L(V, V^*) \), let \( T^* : V \rightarrow V^* \) be a bounded linear operator satisfying
\[ \langle Tu, v \rangle_{V^*, V} = \langle u, T^* v \rangle_{V^*, V}. \]

Note that when we set
\[ T = \begin{pmatrix}
T_{11} & T_{12} & \cdots & T_{1N} \\
T_{21} & T_{22} & \cdots & T_{2N} \\
\vdots & \vdots & \ddots & \vdots \\
T_{N1} & T_{N2} & \cdots & T_{NN}
\end{pmatrix} \]
for each \( T_{ij} \in L(H^1_0(\Omega), H^{-1}(\Omega)) \), we obtain
\[ T^* = \begin{pmatrix}
T_{11}^* & T_{12}^* & \cdots & T_{1N}^* \\
T_{21}^* & T_{22}^* & \cdots & T_{2N}^* \\
\vdots & \vdots & \ddots & \vdots \\
T_{N1}^* & T_{N2}^* & \cdots & T_{NN}^*
\end{pmatrix}. \]

Let \( B(u, \rho) \) be an open ball centered at \( u \in V \) with radius \( \rho \in [0, \infty) \), i.e., \( B(u, \rho) := \{ v \in V \mid \| u - v \|_V < \rho \} \), and let \( \overline{B}(u, \rho) \) be a closure of \( B(u, \rho) \).

Let \( F : V \rightarrow V^* \) be a nonlinear operator defined by \( F(u) := Au - f(u) \). Then, Eq. (2) can be rewritten as
\[ \text{Find } u \in V \text{ s.t. } F(u) = 0. \tag{5} \]

We use a convergence theorem for Newton-like methods for the problem (5). Therefore, we present some notations for the Fréchet derivative. Let \( f'[v] : V \rightarrow V^* \) be the Fréchet derivative at \( v \in V \) of the nonlinear term \( f(u) \), and let \( F'[v] : V \rightarrow V^* \) also be the Fréchet derivative defined by
\[ F'[v] := A - f'[v]. \tag{6} \]

Additionally, a convergence theorem for the Newton-like method requires a linear operator \( \tilde{L}^{-1} \) which is an approximation of the inverse operator \( F'[u]^{-1} \). In general, the computer-assisted proof method is dependent on the selection of \( \tilde{L}^{-1} \).

Let \( V_h \times V_\perp \) be a direct product space with the norm
\[ \left\| \begin{pmatrix}
u_h \\
u_\perp
\end{pmatrix} \right\|_{V_h \times V_\perp} := \left( \left\| \begin{pmatrix}\|u_h\|_V \\
\|u_\perp\|_V
\end{pmatrix} \right\|_E \right)^\frac{1}{2}, \quad u_h \in V_h, \ u_\perp \in V_\perp, \]
where \( \| \cdot \|_E \) denotes the Euclidean norm of vectors. We note that the inner products of \( V_h \) and \( V_\perp \) are the same as \( (\cdot, \cdot)_V \) in \( V \). From the definition of the orthogonal projection \( R_h \), for \( z \in V \), we also have
\[ \|z\|_V = \sqrt{\|R_h z\|_V^2 + \|(I - R_h)z\|_{V^*}^2} = \left( \left\| \begin{pmatrix} R_h z \|_V \\
(I - R_h) z \|_{V^*} \end{pmatrix} \right\|_E \right)^\frac{1}{2}. \]

Let \( u_h \in V_h \subset V \) be an exact solution that satisfies the equation
\[ (u_h, v_h)_V = (f(u_h), v_h)_X, \quad \forall v_h \in V_h. \tag{7} \]

We note that the discretized solution \( u_h \in V_h \) satisfies \( R_h A^{-1} F(u_h) = 0 \). Furthermore, let \( \hat{u} \in V_h \) be an approximation of \( u_h \) as introduced in Section 1.

Finally, the following assumptions are required for a given \( f \) and \( \hat{u} \) (or \( u_h \)) throughout this paper.
**Assumption 1** We assume that a given approximate solution \( \hat{u} \in V_h \) satisfies
\[
f'([\hat{u}])v \in X \subset V^*, \quad \forall v \in V,
\]
and
\[
f'(\hat{u})^*v \in X \subset V^*, \quad \forall v \in V,
\]
that is, we assume that \( f'(\hat{u}) : V \to V^* \) and \( f'(\hat{u})^* : V \to V^* \) are also linear bounded operators from \( V \) to \( X \).

**Remark 1** Assumption 1 guarantees that \( f'(\hat{u}) : V \to V^* \) and \( f'(\hat{u})^* : V \to V^* \) become compact operators via a natural embedding \( X \hookrightarrow V^* \), respectively. Note that Assumption 1 is not so strong because it is not necessary that \( f(u) \in X \) for \( u \in V \). Moreover, it is not required that the Fréchet derivative \( f'(z) \) at any \( z \in V \) belongs to \( L(V,X) \). In Assumption 1, \( f'(\hat{u}) \in L(L,X) \) is required only for a given approximate solution \( \hat{u} \), which is a quite usual assumption.

For example, in case of setting \( N = 1, d = 3 \) and \( f(u) := u^4 \), the function \( u^4 \) is no longer in \( L^2(\Omega) \) for general \( u \in H^1_0(\Omega) \) but \( u^4 \in H^{-1}(\Omega) \). Similarly, from \( f'(z)v = 4z^3v \) for general \( v, z \in H^1_0(\Omega), f'(z)v \) is not in \( L^2(\Omega) \) but in \( H^{-1}(\Omega) \). However, for a given approximate solution \( \hat{u} \in V_h \), it is expected that \( \hat{u} \in L^\infty(\Omega) \). Thus, if \( \hat{u} \in V_h \cap L^\infty(\Omega) \), then from \( f'(\hat{u})v = 4\hat{u}^3v \) for \( v \in V \), \( f'(\hat{u})v \) is also in \( L^2(\Omega) \), i.e., \( f'(\hat{u}) \in L(V,X) \).

### 3. Convergence theorem for simplified Newton-like methods

The Kantorovich theorem is a well-known convergence theorem for the Newton method [16]. There are various theorems about convergence for Newton-like methods (see [48] for a history of convergence theorems for Newton-like methods, and also see Appendix A for the original convergence theorem for Newton-like methods).

In this paper, we use the following theorem, which is a convergence theorem for simplified Newton-like methods for the numerical existence proof. In the numerical existence proof, it is necessary to determine the concrete value for various quantities in the convergence theorem for simplified Newton-like methods. The following theorem should be very useful because the numerical value is automatically determined, except for an approximate solution \( \hat{u} \) and a linear operator \( \hat{L}^{-1} : V^* \to V \). For example, in Theorem 3 in Appendix A, the determination of open convex set \( D_0 \) is cumbersome. On the similar convergence theorem, see, e.g., [50, Chapter 5]

**Theorem 1 (Convergence theorem for simplified Newton-like methods)** Let \( \hat{u} \in V_h \) and \( \hat{L}^{-1} : V^* \to V \) be given. Let \( \eta > 0 \) be a positive constant that satisfies
\[
\| \hat{L}^{-1}F(\hat{u}) \|_V \leq \eta. \tag{8}
\]
Let \( m \geq 0 \) be a constant that satisfies
\[
\| I - \hat{L}^{-1}F'(\hat{u}) \|_{L(V)} \leq m, \tag{9}
\]
where \( I \) is an identity operator on \( V \). Let \( \bar{B}(0,2\eta/(1-m)) := \{ v \in V \mid \| v \|_V \leq 2\eta/(1-m) \} \). The nonlinear operator \( F \) is assumed to be Fréchet differentiable at \( \hat{u} + v \) for all \( v \in \bar{B}(0,2\eta/(1-m)) \).

Let \( K > 0 \) be a constant that satisfies
\[
\| \hat{L}^{-1}(F'(\hat{u}) - F'(\hat{u} + v)) \|_{L(V)} \leq K\| v \|_V, \quad v \in \bar{B} \left( 0, \frac{2\eta}{1-m} \right). \tag{10}
\]
If \( m < 1 \) and \( 2K\eta < (1-m)^2 \), then there exists a solution \( u^* \in V \) of \( F(u^*) = 0 \) that satisfies
\[
\| u^* - \hat{u} \|_V \leq \frac{1-m - \sqrt{(1-m)^2 - 2K\eta}}{K}. \]
Furthermore, the solution \( u^* \) is unique in \( \bar{B}(\hat{u},2\eta/(1-m)) \).
Proof. From the Neumann series theorem (e.g., [49, Theorem 2, p.69]) and the assumption \( \|I - \hat{L}^{-1}\mathcal{F}'(\hat{u})\|_{L(V)} \leq m < 1 \), the linear operator \( \hat{L}^{-1}\mathcal{F}'(\hat{u}) \) is bijective. Moreover, \( \hat{L}^{-1} \) and \( \mathcal{F}'(\hat{u}) \) are also bijective, because \( \mathcal{F}'(\hat{u}) \) is the Fredholm operator with index 0 (see Appendix B). We set \( w := u^* - \hat{u} \).

Since \( \hat{L}^{-1} \) is injective, we have

\[
\mathcal{F}(u^*) = 0
\]
\[
\iff w = w - \hat{L}^{-1}\mathcal{F}(\hat{u} + w)
\]
\[
\iff w = -\hat{L}^{-1}\mathcal{F}(\hat{u}) + w - \hat{L}^{-1}(\mathcal{F}(\hat{u} + w) - \mathcal{F}(\hat{u})).
\]

Let \( T : V \rightarrow V \) be a nonlinear operator:

\[
T(w) := -\hat{L}^{-1}\mathcal{F}(\hat{u}) + w - \hat{L}^{-1}(\mathcal{F}(\hat{u} + w) - \mathcal{F}(\hat{u})).
\]

We set

\[
\rho := \frac{1 - m - \sqrt{(1 - m)^2 - 2\eta K}}{K}.
\]

We now consider the fixed point equation: find \( w \in V \) such that \( w = T(w) \) on a closed ball \( \bar{B}(0, \rho) := \{v \in V \mid \|v\|_V \leq \rho \} \). We first show \( \bar{B}(0, \rho) \subset \bar{B}(0, 2\eta/(1 - m)) \). Therefore, we have

\[
\frac{1 - m - \sqrt{(1 - m)^2 - 2\eta K}}{K} = \frac{2\eta}{1 - m} = \frac{1 - m - \sqrt{(1 - m)^2 - 2\eta K}}{K} = \frac{(1 - m)(1 - \frac{2\eta K}{(1 - m)^2})}{K} = \frac{(1 - m)\sqrt{1 - \frac{2\eta K}{(1 - m)^2}}}{K}
\]
\[
= \frac{1 - m}{K} \left( 1 - \frac{2\eta K}{(1 - m)^2} - \sqrt{1 - \frac{2\eta K}{(1 - m)^2}} \right),
\]

and from

\[
0 < 1 - \frac{2\eta K}{(1 - m)^2} < 1,
\]

we obtain \( \rho < 2\eta/(1 - m) \).

We next show \( T(\bar{B}(0, \rho)) \subset \bar{B}(0, \rho) \), and then we prove that fixed point operator \( T \) is an operator from \( \bar{B}(0, \rho) \) to \( \bar{B}(0, \rho) \). For all \( w \in \bar{B}(0, \rho) \), from the mean-value theorem, we have

\[
\|T(w)\|_V \leq \left\| \hat{L}^{-1}\mathcal{F}(\hat{u}) \right\|_V + \left\| w - \hat{L}^{-1}(\mathcal{F}(\hat{u} + w) - \mathcal{F}(\hat{u})) \right\|_V
\]
\[
\leq \eta + \left\| w - \hat{L}^{-1}\mathcal{F}'([1 - t]\hat{u} + tw) \right\|_V dt
\]
\[
\leq \eta + \int_0^1 \| w - \hat{L}^{-1}\mathcal{F}'[\hat{u} + tw] \|_V dt
\]
\[
\leq \eta + \int_0^1 \| I - \hat{L}^{-1}\mathcal{F}'[\hat{u} + tw] \|_{L(V,V)} \| w \|_V dt
\]
\[
\leq \eta + \int_0^1 \left( \left\| \hat{L}^{-1}(\mathcal{F}'[\hat{u}] - \mathcal{F}'[\hat{u} + tw]) \right\|_{L(V,V)} + \left\| I - \hat{L}^{-1}\mathcal{F}'[\hat{u}] \right\|_{L(V,V)} \right) \| w \|_V dt
\]
\[
\leq \eta + \int_0^1 \left( \left\| \hat{L}^{-1}(\mathcal{F}'[\hat{u}] - \mathcal{F}'[\hat{u} + tw]) \right\|_{L(V,V)} + m \right) \| w \|_V dt.
\]

Furthermore, from \( \rho < 2\eta/(1 - m) \), \( tw \in \bar{B}(0, \rho) \subset \bar{B}(0, 2\eta/(1 - m)) \), we can estimate

\[
\|T(w)\|_V \leq \eta + \int_0^1 (K\|tw\|_V + m) \| w \|_V dt
\]
\[
\leq \frac{K}{2}\| w \|_V^2 + m\| w \|_V + \eta
\]

47
Because $x \in \mathbb{R}$ that satisfies

$$x = \frac{(1 - m) \pm \sqrt{(1 - m)^2 - 2\eta K}}{K}$$

we have $\frac{K}{2} \rho^2 + m \rho + \eta$. Thus, from $\|T(w)\|_V \leq \rho$ for all $w \in \tilde{B}(0, \rho)$, we have proved that $T : \tilde{B}(0, \rho) \to \tilde{B}(0, \rho)$.

Next, we show that the operator $T$ is a contraction operator on $\tilde{B}(0, 2\eta/(1 - m))$. For any $w_1, w_2 \in \tilde{B}(0, 2\eta/(1 - m))$, we have

$$\|T(w_2) - T(w_2)\|_V = \|w_1 - w_2 - \tilde{T}^{-1}(F(\tilde{u} + w_1) - F(\tilde{u} + w_2))\|_V$$

$$= \left\| (w_1 - w_2) - \tilde{T}^{-1} \int_0^1 \tilde{F}'(\tilde{u} + (1-t)w_2 + tw_1)w_1 - w_2) dt \right\|_V$$

$$\leq \int_0^1 \| I - \tilde{T}^{-1}\tilde{F}'(\tilde{u} + (1-t)w_2 + tw_1) \|_{L(V,V)} dt \|w_1 - w_2\|_V$$

$$\leq \int_0^1 \left( \| \tilde{T}^{-1}(\tilde{F}'(\tilde{u}) - \tilde{F}'(\tilde{u} + (1-t)w_2 + tw_1)) \|_{L(V,V)} + m \right) dt \|w_1 - w_2\|_V.$$

For any $w_1, w_2 \in \tilde{B}(0, 2\eta/(1 - m))$ and $0 \leq t \leq 1$, we have $\| (1-t)w_2 + tw_1\|_V \leq (1-t)\|w_2\|_V + t\|w_1\|_V \leq 2\eta/(1 - m)$, and

$$\|T(w_2) - T(w_2)\|_V \leq \int_0^1 (K \| (1-t)w_2 + tw_1\|_V + m) dt \|w_1 - w_2\|_V$$

$$\leq \left( \frac{2\eta K}{1 - m} + m \right) \|w_1 - w_2\|_V.$$

Furthermore, for $2\eta K/(1 - m)$, we have

$$\frac{2\eta K}{1 - m} - (2\eta K + m - m^2) = \frac{2\eta K}{1 - m} - \frac{2\eta K(1 - m)}{1 - m} - \frac{m(1 - m)^2}{1 - m}$$

$$= \frac{2\eta K m}{1 - m} - \frac{m(1 - m)^2}{1 - m}.$$

Because $0 \leq m < 1$ and $0 < (1 - m)^2 - 2\eta K$, we have

$$\frac{2\eta K}{1 - m} \leq 2\eta K + m - m^2,$$

and we can estimate

$$\|T(w_2) - T(w_2)\|_V \leq \left( \frac{2\eta K}{1 - m} + m \right) \|w_1 - w_2\|_V$$

$$\leq (2\eta K + 2m - m^2) \|w_1 - w_2\|_V.$$

Thus, from the assumption, we have

$$2\eta K < (1 - m)^2 = 1 - 2m + m^2$$

$$\Leftrightarrow 2\eta K + 2m - m^2 < 1,$$

and $T$ is the contraction operator on $\tilde{B}(0, 2\eta/(1 - m))$. From $\tilde{B}(0, \rho) \subset \tilde{B}(0, 2\eta/(1 - m))$, $T$ is also the contraction operator on $\tilde{B}(0, \rho)$. Thus, from the Banach fixed point theorem, a solution $u^* \in \tilde{B}(\tilde{u}, \rho)$ that satisfies $F(u^*) = 0$ exists, and the solution $u^*$ is unique in $\tilde{B}(\tilde{u}, 2\eta/(1 - m)).$
Remark 4 We assume that $\mathcal{F}'[\hat{u}]$ is bijective. When we choose $\tilde{L}^{-1}$ as $\mathcal{F}'[\hat{u}]^{-1}$, we almost obtain Kantorovich’s theorem.

Remark 3 If the assumption of Theorem 1 is satisfied, from $\|I - \tilde{L}^{-1}\mathcal{F}'[\hat{u}]\|_{L(V)} < 1$, the linear operator $\mathcal{F}'[\hat{u}]$ has an inverse.

Remark 4 For inequality (10), if the nonlinear operator $f$ is twice Fréchet differentiable, then we have

$$\|\tilde{L}^{-1}(\mathcal{F}'[\hat{u}] - \mathcal{F}'[\hat{u} + v])\|_{L(V)} = \|\tilde{L}^{-1}(f'[\hat{u}] + f'[\hat{u} + v])\|_{L(V)}$$

$$= \|\tilde{L}^{-1}\int_0^1 f''[\hat{u} + tv]v dt\|_{L(V)}$$

$$\leq \int_0^1 \|\tilde{L}^{-1}f''[\hat{u} + tv]v\|_{L(V)} dt.$$  

For $v \in V$, we define a non-decreasing function $K' : [0, 1] \times [0, \infty) \to [0, \infty)$ that satisfies

$$\|\tilde{L}^{-1}f''[\hat{u} + tv]v\|_{L(V)} \leq K'(t, \|v\|_V)\|v\|_V.$$  

Then, using function $K'$, we can estimate constant $K$ of (10) as

$$K = \int_0^1 K'(t, \frac{2\eta}{1 - m}) dt.$$  

4. Verification frameworks

In this section, we demonstrate how to choose $\tilde{L}^{-1}$ in Theorem 1. Let $H_{11} : V_h \to V_h$, $H_{12} : V_\perp \to V_h$, $H_{21} : V_\perp \to V_h$ and $H_{22} : V_\perp \to V_\perp$. For a given $g \in V^*$, we first define the operator matrix

$$H = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} : V_h \times V_\perp \to V_h \times V_\perp$$

that satisfies

$$\begin{pmatrix} R_h \tilde{L}^{-1} g \\ (I - R_h) \tilde{L}^{-1} g \end{pmatrix} = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} \begin{pmatrix} R_h A^{-1} g \\ (I - R_h) A^{-1} g \end{pmatrix}. \quad (12)$$

The simple choice of $\tilde{L}^{-1}$ is $A^{-1}$, which corresponds to FS-Int in 1988 [24], that is, we set

$$H_{FS} := \begin{pmatrix} I_{V_h} & 0 \\ 0 & I_{V_\perp} \end{pmatrix},$$

where $I_X$ denotes the identity operator on $X$. However, the calculation for (9) yields

$$\|I - A^{-1}\mathcal{F}'[\hat{u}]\|_{L(V)} = \sup_{z \in V} \frac{\|f'[\hat{u}]z\|_{V^*}}{\|z\|_V},$$

which the sufficient condition $m < 1$ is, in general, not expected to satisfy. Therefore, this simple selection of operator $\tilde{L}^{-1}$ should be not suitable in the application of Theorem 1. In order to overcome this difficulty, we introduce two kinds of method.

Now we provide the following definition which plays the essential role in our framework:

Definition 1 Let $u_h \in V_h$ be an exact solution that satisfies the finite-dimensional Eq. (7). Let $T : V_h \to V_h$ be a finite-dimensional operator:

$$T = R_h A^{-1} \mathcal{F}'[u_h]|_{V_h}, \quad (13)$$
where $\cdot|_{\mathcal{X}}$ denotes the restriction for the domain of the operator. We assume that operator $T$ is bijective. Let $\delta$ be a constant that satisfies
\[
\|(I - R_h)A^{-1}F(u_h)\|_V \leq \delta. \tag{14}
\]

Let $C_1, C_2, C_3$ be constants satisfying
\[
\begin{align*}
\|T^{-1}R_hA^{-1}f'[u_h]|_{V_{\perp}}v\|_V & \leq C_1\|v\|_{V_{\perp}}, \quad \forall v \in V_{\perp}, \\
\|(I - R_h)A^{-1}f'[u_h]|_{V_{\perp}}z_h\|_V & \leq C_2\|z_h\|_V, \quad \forall z_h \in V_h, \\
\|(I - R_h)A^{-1}f'[u_h]|_{V_{\perp}}z_{\perp}\|_V & \leq C_3\|z_{\perp}\|_V, \quad \forall z_{\perp} \in V_{\perp},
\end{align*}
\]
respectively. Actual evaluation for these constants will be described in Section 5. For $v \in V$ and $t \in [0,1]$, we define the function $K'_T: \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ that satisfies
\[
\|\left( \begin{array}{c}
\|T^{-1}R_hA^{-1}f''[u_h + tv]|_{V}\|_{L(V)} \\
\|(I - R_h)A^{-1}f''[u_h + tv]|_{V}\|_{L(V)} \\
\end{array} \right) \| \leq K'_T(t, \|v\|_V)\|v\|_V. \tag{15}
\]

### 4.1 Method I: Finite-dimensional Newton method (FN method)

Method I was developed in 1990 [25] and then improved in 2004 [27]. Method I is called FN-Norm in [28]. An alternative version of Method I is also extensively used for the numerical existence proof of solutions to some kind of dynamical systems (see, e.g., [10]).

In Method I, we define the operator matrix $H$ as
\[
H_{FN} := \begin{pmatrix} T^{-1} & 0 \\
0 & I_{V_{\perp}} \end{pmatrix}.
\]

Then, Method I is given as the following Corollary.

**Corollary 1 (of Theorem 1: FN method)** Let $\hat{u} \in V_h$ in Theorem 1 be taken as an exact solution $u_h$ that satisfies (7). Assume that nonlinear function $f$ is twice Fréchet differentiable. We set constants $\eta, m, K > 0$ in Theorem 1 as follows:
\[
\begin{align*}
\eta := & \delta, \\
m := & \sqrt{C_1^2 + C_2^2 + C_3^2 + \sqrt{(C_1^2 - C_2^2 + C_3^2)^2 + 4C_2^2C_3^2}}/2, \\
K := & \int_0^1 K'_T \left( t, \frac{2\eta}{1-m} \right) dt.
\end{align*}
\]
If $m < 1$ and $2K\eta < (1 - m)^2$, then there exists a solution $u^* \in V$ of $F(u^*) = 0$ that satisfies
\[
\|u^* - u_h\|_V \leq \frac{1 - m - \sqrt{(1 - m)^2 - 2K\eta}}{K}.
\]
Furthermore, the solution $u^*$ is unique in $B(u_h, 2\eta/(1 - m))$.

**Proof (of Corollary 1)** The upper bound of the norm $\|\hat{L}^{-1}F(\hat{u})\|_V$ in (8) is calculated as
\[
\|\hat{L}^{-1}F(\hat{u})\|_V = \left\| \begin{pmatrix} T^{-1} & 0 \\
0 & I_{V_{\perp}} \end{pmatrix} \begin{pmatrix} R_hA^{-1}F(\hat{u}) \\
(I - R_h)A^{-1}F(\hat{u}) \end{pmatrix} \right\|_{V_h \times V_{\perp}}.
\]
Note that the term $T^{-1}R_hA^{-1}F(\hat{u})$ is a Newton operator for the finite-dimensional Eq. (7). Therefore, by the assumption on the setting of $\hat{u}$, this term vanishes. Thus, we have
\[
\|\hat{L}^{-1}F(\hat{u}_h)\|_V = \|(I - R_h)A^{-1}F(u_h)\|_V.
\]
It implies that we can take as $\eta = \delta$ in (8).
Next, for the constant \( m \) in (9), we have the estimates
\[
\|I - \tilde{\mathcal{L}}^{-1}\mathcal{F}'[\bar{u}]\|_{L(V)} = \sup_{z \in V \setminus \{0\}} \frac{\|(I - \tilde{\mathcal{L}}^{-1}\mathcal{F}'[\bar{u}])z\|_V}{\|z\|_V}
\]
and
\[
\|(I - \tilde{\mathcal{L}}^{-1}\mathcal{F}'[\bar{u}])z\|_V = \left\| \begin{pmatrix} R_h z - T^{-1} R_h A^{-1} \mathcal{F}'[\bar{u}](z) \\ (I - R_h) z - (I - R_h) A^{-1} \mathcal{F}'[\bar{u}](z) \end{pmatrix} \right\|_{V_h \times V_{\perp}}
\]
\[
= \left\| \begin{pmatrix} I_{V_h} - T^{-1} R_h A^{-1} \mathcal{F}'[\bar{u}][V_h] \\ (I - R_h) A^{-1} f'[\bar{u}][V_h] \end{pmatrix} \right\|_{V_h \times V_{\perp}}
\]
\[
= \left\| \begin{pmatrix} 0 \\ (I - R_h) A^{-1} f'[\bar{u}][V_h] \end{pmatrix} \right\|_{V_h \times V_{\perp}}
\]
\[
\leq \left\| \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} \right\|_E \|z\|_V.
\]
Then, by replacing \( \bar{u} \) with \( u_h \), from the requirement in Theorem 1 we obtain
\[
m = \sqrt{\frac{C_1^2 + C_2^2 + C_3^2 + \sqrt{(C_1^2 - C_2^2 + C_3^2)^2 + 4C_2 C_3}}{2}} < 1. \tag{16}
\]

Finally, we demonstrate how to calculate constant \( K \) that satisfies (10). For \( v \in V \) and \( t \in [0, 1] \), from Remark 4 and the function \( K'_T \) in (15), we have
\[
\left\| \tilde{\mathcal{L}}^{-1} f''[\bar{u} + tv] \right\|_{L(V)} \leq \left\| \begin{pmatrix} T^{-1} R_h A^{-1} f''[u_h + tv] \|_V \end{pmatrix} \right\|_E
\]
\[
\leq K'_T(t, \|v\|_V) \|v\|_V.
\]
Thus, we have the following estimate
\[
K = \int_0^1 K'_T(t, \frac{2\delta}{1-m}) dt.
\]

\[\square\]

### 4.2 Method II: Infinite-dimensional Newton method

In Method II, we set \( \tilde{\mathcal{L}}^{-1} := \mathcal{F}'[\bar{u}]^{-1} \) with operator matrix \( H \). This was first studied by Plum in 1991 [32] and has been studied for a long time [26, 30, 33, 35, 36], including the second author and third author. Furthermore, because it is difficult to compute \( \mathcal{F}'[\bar{u}]^{-1} \) directly, the evaluation of \( \|\mathcal{F}'[\bar{u}]^{-1}\|_{L(V^\ast, V)} \) has been intensively studied using norm decomposition. See Appendix B for the method of using \( \|\mathcal{F}'[\bar{u}]^{-1}\|_{L(V^\ast, V)} \).

By contrast, recently, by using an operator matrix \( H \) the authors first introduced in [40], it has become possible to make a direct evaluation of \( \mathcal{F}'[\bar{u}]^{-1} \) without norm estimation \( \|\mathcal{F}'[\bar{u}]^{-1}\|_{L(V^\ast, V)} \). In this method, it is possible to decompose the norms of various patterns. In the present paper, we provide the following corollary in which we use the same symbols as Corollary 1 with Definition 1.

**Corollary 2 (of Theorem 1: IN method)** Let \( \bar{u} \in V_h \) in Theorem 1 be taken as an exact solution \( u_h \) that satisfies (7). Assume that the nonlinear function \( f \) is twice Fréchet differentiable. Let \( \kappa \) be a constant defined by
\[
\kappa := C_3 + C_1 C_2.
\]
We set constants \( \eta, m, K > 0 \) in Theorem 1 as follows:
\[ \eta := \frac{\sqrt{1 + C_1^2} \delta}{1 - \kappa} \]
\[ m := 0 \]
\[ K := \frac{1}{1 - \kappa} \left\| \begin{pmatrix} 1 - C_3 & C_1 \\ C_2 & 1 \end{pmatrix} \right\|_E \int_0^1 K_T'(t, 2\eta) \, dt, \]
where
\[ \left\| \begin{pmatrix} 1 - C_3 & C_1 \\ C_2 & 1 \end{pmatrix} \right\|_E = \sqrt{1 + C_1^2 + C_2^2 + (1 - C_3)^2 + \sqrt{(C_2^2 + (1 - C_3)^2 - (1 + C_1^2))^2 + 4(C_1(1 - C_3) + C_2)^2}}. \]

If \( \kappa < 1 \) and \( 2K\eta < 1 \), then there exists a solution \( u^* \in V \) of \( F(u^*) = 0 \) that satisfies
\[ \|u^* - u_h\|_V \leq \frac{1 - \sqrt{1 - 2K\eta}}{K}. \]
Furthermore, the solution \( u^* \) is unique in \( B(u_h, 2\eta) \).

Before proving Corollary 2, we introduce the theorem related to the definition of the operator matrix \( H \). We consider a solution \( \psi \in V \) that satisfies the linear equation
\[ F'[\hat{u}]\psi = g \tag{17} \]
for a given \( g \in V^* \). Let \( D : V_h \times V_\perp \rightarrow V_h \times V_\perp \) be an operator matrix
\[ D := \begin{pmatrix} T & -R_hA^{-1}f'[\hat{u}]|_{V_\perp} \\ -(I - R_h)A^{-1}f'[\hat{u}]|_{V_h} & I_{V_\perp} - (I - R_h)A^{-1}f'[\hat{u}]|_{V_\perp} \end{pmatrix}. \]
Transforming Eq. (17) using operator matrix \( D \) yields
\[ D \begin{pmatrix} R_h\psi \\ (I - R_h)\psi \end{pmatrix} = \begin{pmatrix} R_hA^{-1}g \\ (I - R_h)A^{-1}g \end{pmatrix}. \]
The following theorem provides conditions that \( D : V_h \times V_\perp \rightarrow V_h \times V_\perp \) and \( F'[\hat{u}] : V \rightarrow V^* \) are bijective. Thus, this theorem also presents the explicit form \( D^{-1} \), which is the operator matrix \( H \) of operator \( F'[\hat{u}]^{-1} \) (see Appendix C for details of the proof).

**Theorem 2** The finite-dimensional operator \( T : V_h \rightarrow V_h \), defined by (13), is assumed to be nonsingular. Let \( B : V_\perp \rightarrow V_\perp \) be a linear operator defined by
\[ B := (I - R_h)A^{-1}f'[\hat{u}]|_{V_\perp} + (I - R_h)A^{-1}f'[\hat{u}]|_{V_h}T^{-1}R_hA^{-1}f'[\hat{u}]|_{V_\perp}. \tag{18} \]
Let \( S : V_\perp \rightarrow V_\perp \) be a linear operator defined by
\[ S := I_{V_\perp} - B. \tag{19} \]
If, for the constant \( \kappa \) in Corollary 2, \( \kappa < 1 \) holds, then the operators \( S, D, \) and \( F'[\hat{u}] \) are bijective and the operator matrix \( H_{IN} \) that satisfies (12) is obtained as
\[ H_{IN} = \begin{pmatrix} T^{-1} + T^{-1}R_hA^{-1}f'[\hat{u}]|_{V_\perp}S^{-1}(I - R_h)A^{-1}f'[\hat{u}]|_{V_h}T^{-1} & T^{-1}R_hA^{-1}f'[\hat{u}]|_{V_\perp}S^{-1} \\ S^{-1}(I - R_h)A^{-1}f'[\hat{u}]|_{V_h}T^{-1} & S^{-1} \end{pmatrix}, \]
and we have
\[ \|S^{-1}\|_{L(V_\perp)} \leq \frac{1}{1 - \kappa}. \]

52
Proof (of Corollary 2) First, we show \(\|B\|_{L(V_\perp)} < 1\). Because the norm of \(V_\perp\) is \(\|\cdot\|_V\), we have
\[
\|B\|_{L(V_\perp)} = \sup_{z_\perp \in V_\perp \setminus \{0\}} \frac{\|Bz_\perp\|_V}{\|z_\perp\|_V} \\
\leq \sup_{z_\perp \in V_\perp \setminus \{0\}} \frac{\|(I-R_h)A^{-1}f'(\hat{u})\|_{V_\perp} + \|(I-R_h)A^{-1}f'(\hat{u})\|_{V_\perp} T^{-1}R_hA^{-1}f'(\hat{u})\|_{V_\perp} \|z_\perp\|_V}{\|z_\perp\|_V} \\
\leq C_3 + C_1 C_2 = \kappa.
\]
Therefore, \(\|B\|_{L(V_\perp)} < 1\) is satisfied from the assumption \(\kappa < 1\) of Corollary 2.
Next, using Theorem 2, we can estimate (8) as
\[
\|F'[\hat{u}]^{-1}F(\hat{u})\|_V = \left\| H_{IN} \left( \frac{R_hA^{-1}F(\hat{u})}{(I-R_h)A^{-1}F(\hat{u})} \right) \right\|_{V_\perp \times V_\perp}.
\]
Furthermore, we choose the approximate solution \(\hat{u} = u_h\), which is the exact solution that satisfies the finite-dimensional Eq. (7), and because \(R_hA^{-1}F(\hat{u})\) is zero, we have
\[
\|F'[u_h]^{-1}F(u_h)\|_V = \left\| \left( \frac{T^{-1}R_hA^{-1}f'[u_h]_{V_\perp} S^{-1}(I-R_h)A^{-1}F(u_h)_V}{S^{-1}(I-R_h)A^{-1}F(u_h)_V} \right) \right\|_E.
\]
Thus, because
\[
\|S^{-1}\|_{L(V_\perp)} = \sup_{z_\perp \in V_\perp \setminus \{0\}} \frac{\|S^{-1}z_\perp\|_V}{\|z_\perp\|_V} \leq \frac{1}{1-\kappa} \Leftrightarrow \|S^{-1}z_\perp\|_V \leq \frac{1}{1-\kappa} \|z_\perp\|_V, \forall z_\perp \in V_\perp,
\]
we can estimate
\[
\tilde{\eta} = \sqrt{1 + \frac{C_1^2}{1-\kappa}} \delta.
\]
Next, we show how to calculate constant \(K\) that satisfies (10). Using operator \(H_{FN}\), we have
\[
H_{FN} = \begin{pmatrix}
I_{V_\perp} + T^{-1}R_hA^{-1}f'[\hat{u}]_{V_\perp} S^{-1}(I-R_h)A^{-1}f'[\hat{u}]_{V_\perp} & T^{-1}R_hA^{-1}f'[\hat{u}]_{V_\perp} S^{-1}S^{-1} \& T^{-1}R_hA^{-1}f'[\hat{u}]_{V_\perp} S^{-1}S^{-1}
\end{pmatrix}
\]
Furthermore, using \(K_f(t, \|v\|_V)\) in (15), for \(v \in \bar{B}(0, 2\eta)\), we can estimate
\[
\|F'[\hat{u}]^{-1}f''(\hat{u} + tv)v\|_{L(V)} \leq \left\| \left( \begin{array}{c}
1 + \frac{C_1 C_3}{1-\kappa} \\
\frac{C_2}{1-\kappa} \frac{C_1}{1-\kappa} \\
\frac{C_2}{1-\kappa} \frac{C_1}{1-\kappa}
\end{array} \right) \right\|_E \frac{K_f(t, 2\eta)\|v\|_V}{1-\kappa}.
\]

5. Evaluation of constants \(C_1, C_2, \text{ and } C_3\)
In this section, we show how to calculate the constants \(C_1, C_2, \text{ and } C_3\) in Definition 1 which are required in Corollary 1 and Corollary 2. We first define some constants as follows:

Definition 2 Let \(D(A) := \{z \in V : Az \in X\}\). Let \(A\) be a linear operator that satisfies
\[
A = A|_{D(A)}.
\]
Note that we have \(A^{-1}g = A^{-1}g\) for any \(g \in X\). We define a constant \(C_h\) that satisfies
\[
\|u - R_hu\|_V \leq C_h \|Au\|_X, \forall u \in D(A).
\]

(20)
Then, the constant $C_h$ also satisfies

$$\|u - R_h u\|_X \leq C_h \|u - R_h u\|_V, \quad \forall u \in V$$

from the Aubin–Nitsche trick [2]. The constant $C_h$ depends on the basis of $V_h$. See, e.g., [18] for the case of only convex domain and [23] for that both of convex and nonconvex domain in the finite element method. See also [19] for the spectral Legendre method.

We define the Sobolev type embedding constant $C_{s,p}$ from $V$ to $(L^p(\Omega))^N$ as

$$\|u\|_{(L^p)^N} \leq C_{s,p} \|u\|_V, \quad \forall u \in V,$$

where $p \in [2, \infty)$ for $d = 2$, and $p \in [2, 2d/(d - 2)]$ for $d > 2$. In particular, $C_{s,2}$ is called the Poincaré constant. See, for example, [42, Theorem1.1,Corollary A.1,Corollary A.2] for the verified computation method.

Next, we define the constants $\tau$ and $\hat{\tau}$ as

$$\|T^{-1} R_h A^{-1}\|_{L(X,V)} \leq \tau \quad \text{and} \quad \|T^{-1} R_h A^{-1}\|_{L(V^*,V)} \leq \hat{\tau}. \quad (22)$$

Let $f'_X[u_h]$ be a part of the bounded operator of $f'|u_h|$ on $X$, that is, there exists constant $C$ satisfying $\|f'_X[u_h]|v\|_X \leq C\|v\|_X$ for any $v \in X$. Additionally, let

$$f'_V[u_h] := f'[u_h] - f'_X[u_h].$$

Finally, from Assumption 1, we define the constants $C_{f'_V}, C_{f'_V}$ and $C_{f'_V}$ as

$$\|f'_V[u_h]|v\|_X \leq C_{f'_V}\|v\|_X, \quad \forall v \in V, \quad (23)$$

$$\|f'_V[u_h]|v\|_X \leq C_{f'_V}\|v\|_V, \quad \forall v \in V, \quad (24)$$

$$\|f'_V[u_h]^*|v\|_X \leq C_{f'_V}\|v\|_V, \quad \forall v \in V. \quad (25)$$

Then constants $C_1, C_2,$ and $C_3$ can be rewritten as the following lemma:

**Lemma 1** We assume Assumption 1, and use the same notations as Definition 2. Then, constants $C_1, C_2,$ and $C_3$ in Definition 1 are given by

$$C_1 := C_h \left(\tau C_{f'_V} + \tau C_{f'_V}\right), \quad C_2 := C_h \left(C_{f'_V} + C_{s,2} C_{f'_V}\right), \quad C_3 := C_h \left(C_{f'_V} + C_h C_{f'_V}\right),$$

respectively.

**Proof (of Lemma 1)** Regarding $C_1$:

For any $z_\perp \in V_\perp$, we have

$$\|f'_V[u_h]|z_\perp\|_{V^*} = \|A^{-1} f'_V[u_h]|z_\perp\|_V = \langle A^{-1} f'_V[u_h]|z_\perp, A^{-1} f'_V[u_h]|z_\perp \rangle_V$$

$$= \langle f'_V[u_h]|z_\perp, A^{-1} f'_V[u_h]|z_\perp \rangle_V = \langle z_\perp, f'_V[u_h]^* A^{-1} f'_V[u_h]|z_\perp \rangle_X$$

$$\leq \|z_\perp\|_X \|f'_V[u_h]|z_\perp\|_{V^*} \|A^{-1} f'_V[u_h]|z_\perp\|_X$$

$$\leq C_h \|f'_V[u_h]|z_\perp\|_V \|A^{-1} f'_V[u_h]|z_\perp\|_X$$

$$= C_h \|f'_V[u_h]|z_\perp\|_V \|f'_V[u_h]|z_\perp\|_V.$$  

Thus, we obtain

$$\|T^{-1} R_h A^{-1} f'|u_h|_V z_\perp \|_V$$

$$= \|T^{-1} R_h A^{-1} (f'_V[u_h] + f_X[u_h])|_{V_\perp z_\perp}|_V$$

$$\leq \|T^{-1} R_h A^{-1}\|_{L(V^*,V)} \|f'_V[u_h]|z_\perp\|_V + \|T^{-1} R_h A^{-1}\|_{L(X,V)} \|f'_V[u_h]|z_\perp\|_X$$

$$\leq \hat{\tau} \|f'_V[u_h]|z_\perp\|_V + \tau C_{f'_V}\|z_\perp\|_X$$

$$\leq C_h \left(\tau C_{f'_V} + \tau C_{f'_V}\right) \|z_\perp\|_V.$$  

54
Regarding $C_2$:
For any $z_h \in V_h$, we have
\[
\|(I - R_h)A^{-1}f'[u_h]\|_{V_h} \leq C_h \|f'[u_h]\|_{X}.
\]
\[
\leq C_h \|f'[u_h]\|_{V_h} z_h, x
\]
\[
\leq C_h \left( f_{X}^T[u_h] z_h, x + f_{X}^T[u_h] z_h, x \right)
\]
\[
\leq C_h \left( f_{X}^T[v_h] z_h, v + f_{X}^T[v_h] z_h, x \right)
\]
\[
\leq C_h \left( f_{X}^T + C_{s}d f_{X}^T \right) \|z_h\|_{V}.
\]

Regarding $C_3$:
For any $z_{\perp} \in V_{\perp}$, we have
\[
\|(I - R_h)A^{-1}f'[u_h]\|_{V_{\perp}} \leq C_h \|f'[u_h]\|_{X}.
\]
\[
\leq C_h \|f'[u_h]\|_{V_{\perp} z_{\perp}}
\]
\[
\leq C_h \left( f_{X}^T[u_h] z_{\perp}, x + f_{X}^T[u_h] z_{\perp}, x \right)
\]
\[
\leq C_h \left( f_{X}^T[v_h] z_{\perp}, v + f_{X}^T[v_h] z_{\perp}, x \right)
\]
\[
\leq C_h \left( f_{X}^T + C_{s}d f_{X}^T \right) \|z_{\perp}\|_{V}.
\]

\[\square\]

5.1 Evaluation of constants $\tau$ and $\hat{\tau}$ in (22)
In this subsection, we demonstrate how to compute constants $\tau$ and $\hat{\tau}$ defined in Definition 2. We first define the matrix for the computation of $\tau$ and $\hat{\tau}$.

Definition 3 (Matrix $D, L, Q$) Let $D, L, Q$ be $n \times n$ matrices whose elements are
\[
D_{i,j} := (\phi_{j}, \phi_{i})_{V}, \quad L_{i,j} := (\phi_{j}, \phi_{i})_{X}, \quad Q_{i,j} := (f'[u_h]\phi_{j}, \phi_{i})_{X},
\]
respectively. Thus, because $D$ and $L$ are symmetric positive definite matrices, we define $D^{\frac{1}{2}}$ and $L^{\frac{1}{2}}$ as the Cholesky decomposition
\[
D = D^{\frac{1}{2}} D^{\frac{1}{2}}, \quad L = L^{\frac{1}{2}} L^{\frac{1}{2}},
\]
where $^T$ denote the transpose of matrix.

Then, constants $\tau$ and $\hat{\tau}$ can be reduced to the following eigenvalue problem:

Lemma 2 Let $\mu_{\min}$ and $\hat{\mu}_{\min}$ be the smallest eigenvalues that satisfy the eigenvalue problems
\[
\text{Find } (\mu, x) \in \mathbb{R} \times \mathbb{R}^n \text{ s.t. } (D - (Q + Q^T) + Q D^{-1} Q^T)x = \mu Lx
\]
and
\[
\text{Find } (\hat{\mu}, x) \in \mathbb{R} \times \mathbb{R}^n \text{ s.t. } (D - (Q + Q^T) + Q D^{-1} Q^T)x = \hat{\mu} D x,
\]
respectively. Then, constants $\tau$ and $\hat{\tau}$ satisfy
\[
\tau = \frac{1}{\sqrt{\mu_{\min}}}, \quad \hat{\tau} = \frac{1}{\sqrt{\hat{\mu}_{\min}}}.
\]

Proof We first consider $\tau$. Let $P_{h} : X \rightarrow V_{h}$ be the $X$-projection defined by
\[
((I - P_{h})u, v_{h})_{X} = 0, \quad v_{h} \in V_{h}
\]
for any $u \in X$. 

55
Let $P_h u$ and $w_h \in V_h$ be

$$P_h u := \sum_{i=0}^{N} u_i \phi_i, \quad w_h := \sum_{i=0}^{N} w_i \phi_i,$$

where $u := (u_1, \ldots, u_n)^T$, $w := (w_1, \ldots, w_n)^T \in \mathbb{R}^n$. We set $w_h = T^{-1} R_h A^{-1} u$, and we have

$$(T w_h, v_h)_V = (R_h A^{-1} u, v_h)_V$$
$$(R_h A^{-1}(A - f'(u_h)) w_h, v_h)_V = (R_h A^{-1} u, v_h)_V$$
$$(w_h, v_h)_V - (f'(u_h) w_h, v_h)_X = (P_h u, v_h)_X$$
$$(D - Q) w = L u$$

Thus, we have

$$\|T^{-1} R_h A^{-1}\|_{L(X,V)} = \sup_{u \in X \setminus \{0\}} \frac{\|T^{-1} R_h A^{-1} u\|_V}{\|u\|_X} = \sup_{u \in X \setminus \{0\}} \frac{\|w_h\|_V}{\|u\|_X}$$

$$= \sup_{u \in X \setminus \{0\}} \frac{\|D^{\frac{1}{2}} u\|_E}{\|u\|_X} = \sup_{u \in X \setminus \{0\}} \frac{\|D^{\frac{1}{2}} (D - Q)^{-1} L u\|_E}{\|u\|_X}$$

$$\leq \|D^{\frac{1}{2}} (D - Q)^{-1} L^{\frac{1}{2}}\|_E \sup_{u \in X \setminus \{0\}} \frac{\|L^{\frac{1}{2}} u\|_E}{\|u\|_X}$$

$$\leq \|D^{\frac{1}{2}} (D - Q)^{-1} L^{\frac{1}{2}}\|_E \sup_{u \in X \setminus \{0\}} \frac{\|P_h u\|_X}{\|u\|_X}$$

$$\leq \|D^{\frac{1}{2}} (D - Q)^{-1} L^{\frac{1}{2}}\|_E.$$

Let $\lambda_{\text{max}}$ be the largest eigenvalue that satisfies

$$\text{Find } (\lambda, x) \in \mathbb{R} \times \mathbb{R}^n \text{ s.t. } \left(D^{\frac{1}{2}} (D - Q)^{-1} L^{\frac{1}{2}}\right)^T \left(D^{\frac{1}{2}} (D - Q)^{-1} L^{\frac{1}{2}}\right) x = \lambda x.$$ 

Then, we have $\|D^{\frac{1}{2}} (D - Q)^{-1} L^{\frac{1}{2}}\|_E = \sqrt{\lambda_{\text{max}}}$. Furthermore, we transform the eigenvalue problem

$$\left(D^{\frac{1}{2}} (D - Q)^{-1} L^{\frac{1}{2}}\right)^T \left(D^{\frac{1}{2}} (D - Q)^{-1} L^{\frac{1}{2}}\right) x = \lambda x$$
$$\Longleftrightarrow L^{\frac{1}{2}} (D - Q)^{-T} D (D - Q)^{-1} L^{\frac{1}{2}} x = \lambda x$$
$$\Longleftrightarrow (D - Q)^{-T} D (D - Q)^{-1} x = \lambda L^{\frac{1}{2}} x$$
$$\Longleftrightarrow D (D - Q)^{-1} x = \lambda (D - Q)^T L^{\frac{1}{2}} x$$
$$\Longleftrightarrow (D - Q)^{-1} x = \lambda (D - Q)^T L^{\frac{1}{2}} x$$
$$\Longleftrightarrow x = \lambda (D - Q)^T L^{\frac{1}{2}} x$$
$$\Longleftrightarrow x = \lambda (D - Q)(I - D^{-1} Q^T) L^{\frac{1}{2}} x$$
$$\Longleftrightarrow L x = \lambda \left(D - (Q + Q^T) + Q D^{-1} Q^T\right) x$$
$$\Longleftrightarrow \left(D - (Q + Q^T) + Q D^{-1} Q^T\right) x = \frac{1}{\lambda} L x.$$ 

Therefore, we have $\|D^{\frac{1}{2}} (D - Q)^{-1} L^{\frac{1}{2}}\|_E = \sqrt{\lambda_{\text{max}}} = 1/\sqrt{\mu_{\text{min}}}$. 

Next, we show about $\hat{\tau}$. Let $R_h A^{-1} u$ be

$$R_h A^{-1} u := \sum_{i=0}^{N} u_i \phi_i, \quad w_h := \sum_{i=0}^{N} w_i \phi_i,$$

where $u := (u_1, \ldots, u_n)^T$, $w := (w_1, \ldots, w_n)^T \in \mathbb{R}^n$. We set $w_h = T^{-1} R_h A^{-1} u$, and we have
Thus, we have
\[
\|T^{-1}R_h A^{-1}\|_{L(V^*, V)} = \sup_{u \in V^* \setminus \{0\}} \frac{\|T^{-1}R_h A^{-1}u\|_V}{\|u\|_{V^*}} = \sup_{u \in V^* \setminus \{0\}} \frac{\|u_h\|_V}{\|u\|_{V^*}}
\]
\[
= \sup_{u \in V^* \setminus \{0\}} \frac{D^\frac{1}{2} u_E}{\|u\|_{V^*}} \sup_{u \in V^* \setminus \{0\}} \frac{D^\frac{1}{2} (D - Q)^{-1} D u_E}{\|u\|_{V^*}}
\]
\[
\leq \|D^\frac{1}{2} (D - Q)^{-1} D^\frac{1}{2}\|_E \sup_{u \in V^* \setminus \{0\}} \frac{D^\frac{1}{2} u_E}{\|u\|_{V^*}} \sup_{u \in V^* \setminus \{0\}} \frac{D^\frac{1}{2} (D - Q)^{-1} D u_E}{\|u\|_{V^*}}
\]
\[
\leq \|D^\frac{1}{2} (D - Q)^{-1} D^\frac{1}{2}\|_E.
\]

Let \(\lambda_{\text{max}}\) be the largest eigenvalue that satisfies

\[
\text{Find} \ (\lambda, x) \in \mathbb{R} \times \mathbb{R}^n \text{ s.t.} \ (D^\frac{1}{2} (D - Q)^{-1} D^\frac{1}{2})^T (D^\frac{1}{2} (D - Q)^{-1} D^\frac{1}{2}) x = \lambda x.
\]

Then, we have \(\|D^\frac{1}{2} (D - Q)^{-1} D^\frac{1}{2}\|_E = \sqrt{\lambda_{\text{max}}}\.\) Furthermore, we transform the eigenvalue problem

\[
(D^\frac{1}{2} (D - Q)^{-1} D^\frac{1}{2})^T (D^\frac{1}{2} (D - Q)^{-1} D^\frac{1}{2}) x = \lambda x
\]
\[
\Leftrightarrow (D - (Q + Q^T) + Q D^{-1} Q^T) x = \frac{1}{\lambda} Dx.
\]

Therefore, we obtain \(\|D^\frac{1}{2} (D - Q)^{-1} D^\frac{1}{2}\|_E = \sqrt{\lambda_{\text{max}}} = 1/\sqrt{\mu_{\text{min}}}\).

**6. Evaluation of residual constant \(\delta\) in (14)**

In this section, we introduce how to evaluate residual constant \(\delta\) that satisfies (14). The method for computing the residual constant varies whether \(Au_h \in X\) or \(Au_h \not\in X\). In the case of \(Au_h \not\in X\), see [47] and [41]. In this paper, we only demonstrate how to compute constant \(\delta\) in the case of \(Au_h \in X\):

\[
\|(I - R_h) A^{-1} f(u_h)\|_V = \|(I - R_h) A^{-1} (Au_h - f(u_h))\|_V \leq C_h \|Au_h - f(u_h)\|_X
\]
\[
= C_h \sqrt{(Au_h, Au_h)_X - 2(Au_h, f(u_h))_X + (f(u_h), f(u_h))_X}. \quad (26)
\]

Note that the inner product in (26) can be computable.

**7. VCP library**

In Sections 2–6, we mathematically described methods of the computer-assisted existence proof of solutions to PDEs. However, when using a computer for the calculation, it is necessary to perform the computation with a guaranteed accuracy that takes into account all errors. For example, in the eigenvalue problem of Lemma 2, verification is necessary not only for computing eigenvalues but also for rigorously generating coefficient matrices \(D, Q,\) and \(L\). Additionally, the computation of residuals in Section 6 requires high-precision computations. For this reason, computer-assisted proofs for solutions to PDEs require various techniques, such as parts that require high-speed computation and parts that require high-precision computation. However, there are few numerical computation
libraries (e.g. [1, 3, 15, 38]) with guaranteed accuracy that satisfy all requirements, such as high accuracy, high speed, and numerical linear algebra.

In this paper, we introduce the VCP library, which is a software library that provides several classes written in the C++ programming language. Since the VCP library is provided as header files of C++, the VCP library does not require special installation. For example, when we use the matrix class of the VCP library, it is necessary to write a program such as `#include<vcp/matrix.hpp>`. The latest version can be downloaded using the following URL:

https://verified.computation.jp/VCP_Lib/vcp_latest.zip

In this paper, we introduce two classes:

- **Matrix class**: Classes that compute linear algebra calculations.
- **Legendre basis class**: Class that computes the Legendre basis.

We require using the following libraries in the VCP library:

- BLAS and Lapack:
- MPFR [37]: multiple-precision floating-point number library
- kv library [17]: numerical verification library with guaranteed accuracy written in the C++ programming language.

### 7.1 VCP’s matrix class

The matrix class of the VCP library is based on a policy using a template. Therefore, when declaring the matrix, it is necessary to determine “data type for elements” and “algorithm policy” as template arguments. The VCP library provides the following four Algorithm policies:

- **vcp::mats< T >**: performs approximate computations on general data type T.
- **vcp::pdblas**: performs fast approximate computations on `double` data type using BLAS and Lapack.
- **vcp::imats< T >**: performs verified computations on general data type T.
- **vcp::pidblas**: performs fast verified computations on `double` data type using BLAS and Lapack.

For example, we can use the VCP library as shown in Source code 1. Line 6 in Source code 1 declares matrix $A$, $b$, $x$ with “double-type” and “vcp::mats<double>-algorithm.” Random matrix $A$ and random vector $b$ are created on lines 7 and 8, respectively. On line 9, the result of $A$ times $b$ is substituted for $b$. Line 10 uses the function lss to determine the solution $x$ of simultaneous linear equation $Ax = b$. Finally, line 11 displays the solution $x$.

**Source code 1.** How to use VCP’s matrix class

```cpp
#include <iostream>
#include <vcp/matrix.hpp>
#include <vcp/matrix_assist.hpp>

int main(void){
    vcp::matrix< double, vcp::mats<double>> A, b, x;
    A.rand(10); // Create a 10*10 random matrix
    b.rand(10,1); // Create a 10*1 random vector
    b = A*b; // Compute A times b
    x = lss(A, b); // Solve Ax = b
    std::cout << x << std::endl; // Display x
}
```
The feature of the VCP's matrix class is that it can be changed to verify a numerical computation by changing the “kv::interval< T >” data type and “algorithm policy.” For example, when verifying a computation with high precision, the VCP’s matrix class can be written as Source code 2. Line 5 in Source code 2 declares matrix A, b, x with "kv::interval<kv::mpfr<300>>-type" and "vcp::imats<kv::mpfr<300>>-algorithm," where kv::mpfr<300> is an MPFR type with a mantissa part of 300 bits, and kv::interval<T> is kv’s interval arithmetic class. Because "vcp::imats<T>" is selected as the algorithm policy in line 12, the matrix computation is an algorithm with guaranteed accuracy. Therefore, lines 13 to 17 in Source code 2 are the same as lines 7 to 11 in Source code 1, but the results of the matrix-vector product on line 15 and the solution x on line 16 in Source code 2 contain exact solutions.

Source code 2. VCP’s matrix class using the kv library

```cpp
#include <iostream>
#include <kv/interval.hpp> // kv library
#include <kv/mpfr.hpp> // kv library
#include <kv/rmpfr.hpp> // kv library
#include <vcp/imats.hpp>
#include <vcp/matrix.hpp>
#include <vcp/matrix_assist.hpp>

int main(void){
    vcp::matrix< kv::interval<kv::mpfr<300>>, vcp::imats<kv::mpfr<300>> > A, b, x;
    A.rand(10); // Create a 10*10 random matrix
    b.rand(10,1); // Create a 10*1 random vector
    b = A*b; // Compute A times b
    x = lss(A, b); // Solve Ax = b
    std::cout << x << std::endl; // Display x
}
```

Additionally, if we declare a matrix with vcp::matrix<double, vcp::pdblas>, it can be changed to an algorithm using BLAS and Lapack. When using high-speed numerical computation with guaranteed accuracy using BLAS and Lapack, we declare a matrix similar to vcp::matrix<kv::interval<double>, vcp::pidblas>. See Appendix D for other functions of VCP’s matrix class.

7.2 VCP’s Legendre basis generator class

Let Ω = (0, 1)^d. We define the set \{\tilde{\phi}_1, \tilde{\phi}_2, \cdots, \tilde{\phi}_n\} of Legendre basis as

\[
\tilde{\phi}_i(x) := \frac{1}{i(i+1)} x(1-x) \frac{d}{dx} P_i(x), \quad i = 1, 2, \cdots, \quad \text{with} \quad P_i = \frac{(-1)^i}{i!} \left( \frac{d}{dx} \right)^i x^i (1-x)^i.
\]

We also define the finite-dimensional subspace as a tensor product:

\[
\tilde{V}_h^M := \left( \text{span}(\tilde{\phi}_1, \cdots, \tilde{\phi}_M) \right)^d.
\]

Then, for \(x = (x_1, x_2, \cdots, x_d)^T\), \(\tilde{u} \in \tilde{V}_h^M\) can be written by

\[
\tilde{u}(x) = \sum_{i_1=1}^{M} \sum_{i_2=1}^{M} \cdots \sum_{i_d=1}^{M} \tilde{u}_{i_1,i_2,\cdots,i_d}(x_1) \tilde{\phi}_{i_1}(x_1) \tilde{\phi}_{i_2}(x_2) \cdots \tilde{\phi}_{i_d}(x_d),
\]

where \(\tilde{u}_{i_1,i_2,\cdots,i_d}\) is a real number. We redefine index \(i = 1, \cdots, M^d\) for index pair \((i_1, i_2, \cdots, i_d)\), and we define \(\phi_i(x) = \tilde{\phi}_{i_1}(x_1) \tilde{\phi}_{i_2}(x_2) \cdots \tilde{\phi}_{i_d}(x_d)\) for index pair \((i_1, i_2, \cdots, i_d)\). Moreover, \(\tilde{u} \in \tilde{V}_h^M\) can also be written by

\[
\tilde{u}(x) = \sum_{i=1}^{M^d} \tilde{u}_i \phi_i(x).
\]

The following four preparations are required to use the VCP’s Legendre basis generator class.
7.2.1 Step 1: Declaration of objects for the Legendre basis class
The declaration of the object Generator of the Legendre basis class of the VCP library is

\[
\text{vcp::Legendre_Bases_Generator< Type1, Type2, MatrixPOLICY > Generator;}
\]

where Type1 must be \text{kv::interval<kv::mpfr<N>>}. Type2 is a data type of the matrix/vector element and MatrixPOLICY is its matrix algorithm policy.

7.2.2 Step 2: The setting method function
The object Generator has the following three modes:

- Verification mode: \text{mode}=1: generates matrices, vectors, and constants that are required to compute \( u_h \) and Lemma 1.
- Residual mode: \text{mode}=2: computes an inner product in (26).
- Approximation mode: \text{mode}\geq 3: creates matrices and vectors necessary to compute \( \hat{u} \). The larger the value of the mode, the better the accuracy of the approximate numerical integration.

We initialize object Generator, such as the mode, using a method function setting. The \text{setting} method is used in Residual mode and Approximation mode as follows:

\[
\text{Generator.settling(uh_order, p, Dimension, Number_of_variables, mode);}
\]

where \( uh_order \) is number of bases \( n \), \( p \) is the maximum degree of the original equation, Dimension is the dimension of given domain \( \Omega \), and \( \text{Number_of_variables} \) is the number of variables spanned by Legendre polynomials.

Additionally, the \text{setting} method with the verification mode is written as follows:

\[
\text{Generator.settling(VOrder, p, Dimension, Number_of_variables, mode, uh_order);}
\]

where \( VOrder \) is also number of bases \( n \). Moreover, \( uh_order \) is the order of bases that already exist.

7.2.3 Step 3: The setting_list method function
Next, in all modes, we need to execute the following method function that creates a list that maps from index \( i \) to \( i_1, i_2, \cdots, i_d \):

\[
\text{Generator.settling_list();}
\]

Method function \text{setting_list} creates a list using the full index. By contrast, to assume symmetry for the Legendre basis, we create a list with only the index as follows:

\[
\text{Generator.settling_evenlist();}
\]

The created list can be obtained as follows:

\[
\text{vcp::matrix< int > list_uh = Generator.output_list();}
\]

7.2.4 Step 4: The setting_uh method function
Next, in all modes, we need to insert \( uh \), which is an \( N_{list} \times M_{nv} \) matrix with data type \text{vcp::matrix< Type2, MatrixPOLICY >}, where \( N_{list} = \text{list_uh.rowsize}() \) and \( M_{nv} = \text{Number_of_variables} \). For example, \( uh \) inserts the initial value or iteration of Newton’s method when determining approximate solutions. The \text{setting_uh} method is used in Residual mode and Approximation mode as follows:

\[
\text{Generator.settling_uh(uh);}
\]

Additionally, the \text{setting_uh} method with the verification mode is written as follows:

\[
\text{Generator.settling_uh(uh, list_uh, list_type);}
\]

where \( list_type \) is an integer number. If \( list_uh \) is a full list, then \( list_type=1 \). If \( list_uh \) is an even list, then \( list_type=2 \).
7.2.5 Method functions for VCP’s Legendre basis generator class

VCP’s Legendre basis generator class is given by the following method functions:

- **Generator.dphidphi()**: creates matrix $D \in \mathbb{R}^{M' \times M'}$ that satisfies $D_{i,j} = (\phi_j, \phi_i)_{H^1}$.
- **Generator.phiphi()**: creates matrix $L \in \mathbb{R}^{M' \times M'}$ that satisfies $L_{i,j} = (\phi_j, \phi_i)_{L^2}$.
- **Generator.uhiphi(p1, p2, \ldots, p_N)**: creates vector $f \in \mathbb{R}^{M'}$ that satisfies $f_i = (\hat{u}_1^{p_1} \hat{u}_2^{p_2} \cdots \hat{u}_N^{p_N}, \phi_i)_{L^2}$ for given $\hat{u}_1, \cdots, \hat{u}_N \in \hat{V}_h^M$.
- **Generator.uhiphiphi(p1, p2, \ldots, p_N)**: creates matrix $Q' \in \mathbb{R}^{M' \times M'}$ that satisfies $Q'_{i,j} = (\hat{u}_1^{p_1} \hat{u}_2^{p_2} \cdots \hat{u}_N^{p_N}, \phi_j)_{L^2}$ for given $\hat{u}_1, \cdots, \hat{u}_N \in \hat{V}_h^M$.
- **Generator.global_min(x, minimum size)**, **Generator.global_max(x, minimum size)**: solves each minimum/maximum values that satisfies $\hat{u}_1(x), \cdots, \hat{u}_N(x)$ for given $\hat{u}_1, \cdots, \hat{u}_N \in \hat{V}_h^M$.
- **Generator.Poincare_constant<T>()**: computes constant $C_{s,2}$ that satisfies (21) in Definition 2.
- **Generator.Sobolev_constant<T>()**: computes constant $C_{s,p}$, $p > 2$ that satisfies (21) in Definition 2.
- **Generator.Ritz_projection_error<T>()**: computes constant $C_{h}$ that satisfies (20) in Definition 2.
- **Generator.integral_LuhLuh(i)**: computes $(-\Delta \hat{u}_i, -\Delta \hat{u}_i)_{L^2}$ for given $\hat{u}_i \in \hat{V}_h^M$.
- **Generator.integral_Luhuh(i, p1, p2, \ldots, p_N)**: computes $(-\Delta \hat{u}_i, \hat{u}_1^{p_1} \hat{u}_2^{p_2} \cdots \hat{u}_N^{p_N})_{L^2}$ for given $\hat{u}_1, \cdots, \hat{u}_N \in \hat{V}_h^M$.
- **Generator.integral_uh(p1, p2, \ldots, p_N)**: computes $\int \hat{u}_1^{p_1} \hat{u}_2^{p_2} \cdots \hat{u}_N^{p_N} d\Omega$ for given $\hat{u}_1, \cdots, \hat{u}_N \in \hat{V}_h^M$.

8. Numerical examples

In this section, we present numerical examples. All computations were implemented on a computer with 2.20 GHz Intel Xeon E7-4830 v2 CPU $\times 4$, 2 TB RAM, and CentOS 7.2 using C++11 with GCC version 4.8.5.

8.1 Example 1

We first present an example in which our method is used to verify a solution of the elliptic boundary value problem:

$$
\begin{aligned}
-\Delta u &= u^2 \quad \text{in } \Omega, \\
 u &= 0 \quad \text{on } \partial \Omega,
\end{aligned}
$$

where $\Omega = (0,1)^2$. The source code can be downloaded from

https://verified.computation.jp/testcode/NOLTA2019/Emden.cpp

Additionally, the first and second Fréchet derivative of $f$ at $u_h$ are

$$
\begin{aligned}
f'(u_h) &= 2u_h \text{ and } f''(u_h) = 2.
\end{aligned}
$$

We set $f'_V[u_h]$ and $f'_X[u_h]$ defined by Definition 2 as

$$
\begin{aligned}
f'_V[u_h] &= 0 \quad \text{and} \quad f'_X[u_h] = 2u_h.
\end{aligned}
$$
Therefore, from $\|2u_h v\|_{L^2} \leq 2\|u_h\|_{L^\infty} \|v\|_{L^2}, \forall v \in L^2(\Omega)$, constants $C_{f'}$, $C_{f''}$, $C_{f'''}$ defined by (23)–(25) are

$$C_{f'} = C_{f''} = 0 \quad \text{and} \quad C_{f'''} = 2\|u_h\|_{L^\infty},$$

respectively.

Next, we consider the function $K'_T(t, \|v\|_V)$ defined by (15). Using $\tau$, $C_h$ and $C_{s,A}$ defined by (22), (20), and (21), respectively, we obtain

$$\begin{align*}
&\left\| \left( \left| T^{-1} R_h A^{-1} f''(u_h + tv) v \right|_{L(H_0^1)} \right) \right\|_E = \left\| \left( \left| (I - R_h) A^{-1} f''(u_h + tv) v \right|_{L(H_0^1)} \right) \right\|_E \\
&\leq 2 \left( \frac{\tau}{C_h} \right) \|v\|_{L(H_0^1,L^2)} = 2\sqrt{\tau^2 + C_h^2} \sup_{\phi \in V \setminus \{0\}} \frac{\|v\phi\|_{L^2}}{\|\phi\|_{H_0^1}} \\
&\leq 2C_{s,A}^2 \sqrt{\tau^2 + C_h^2} \|v\|_{H_0^1}.
\end{align*}$$

Therefore, from $K'_T(t, \|v\|_V) = 2C_{s,A}^2 \sqrt{\tau^2 + C_h^2} \int_0^1 K'_T(t, \cdot) dt$ determined by nonlinear term $f(u)$ is

$$\int_0^1 K'_T(t, \cdot) dt = 2C_{s,A}^2 \sqrt{\tau^2 + C_h^2}.$$

For this setting, an approximate solution $u_h$ was computed numerically, the graphs of which are displayed in Fig. 1.

![Fig. 1. Approximate solution of (27).](image)

The constants in Definition 1, Definition 2, and Lemma 1 were estimated and are presented in Table I, where “Order” denotes the value of $u h_{\text{Order}}$ and $v_{\text{Order}}$ in the VCP library, that is, the order of the Legendre basis.

The FN method (Corollary 1) yielded the results presented in Table II. In the case of “Order” = 20 and 40, the existence of a solution can be proven using the FN method to satisfy $m < 1$ and $\frac{2K_q}{(1-m)^2} < 1$. Additionally, the upper bound of the norm $\|u^* - u_h\|_{H_0^1}$ was obtained. Constant $m$, which is important for verification using the FN method, is close to the maximum values of $C_1$, $C_2$, and $C_3$. Therefore, verification may fail if either $C_1$, $C_2$, or $C_3$ is large.

The IN method (Corollary 2) yielded the results presented in Table III. The IN method also proved the existence of an exact solution and yielded the upper bound of norm $\|u^* - u_h\|_{H_0^1}$. Constant $\kappa = C_3 + C_1 C_2$, which is important for verification using the IN method, was less than 1.

Next, we compare Tables II with III. As described in Remark 3, Fréchet derivative $F'(u_h)$ is invertible if $m$ in Table II is less than 1. Similarly, if $\kappa$ in Table III is less than 1, then Fréchet derivative $F'(u_h)$ is invertible. Moreover, comparing $m$ with $\kappa$, $\kappa$ is smaller than $m$, thus it is easier for the IN method to prove invertibility of $F'(u_h)$ in this example. Additionally, for the error bound $\|u^* - u_h\|_{H_0^1}$, the IN method is smaller than the FN method.
Table I. Results for some constants.

| Order | $C_{s,2}$ | $C_{s,4}$ | $C_h$ | $\tau$ | $C_f$ | $C_1$ | $C_2$ | $C_3$ |
|-------|-----------|-----------|-------|--------|-------|-------|-------|-------|
| 20    | 0.2251    | 0.3184    | 0.0213 | 0.1579 | 58.52 | 0.1969 | 0.2808 | 0.02660 |
| 40    | 0.2251    | 0.3184    | 0.0115 | 0.1579 | 58.52 | 0.1063 | 0.1515 | 0.007741 |

Table II. Verification results using the FN method (Corollary 1).

| Order | $\eta$ (= $\delta$) | $m$ | $K$ | $\frac{2K\eta}{(1 - 2m^2)}$ | $\|u^* - u_h\|_{H^1}$ |
|-------|--------------------|-----|-----|--------------------------|---------------------|
| 20    | $4.023 \times 10^{-6}$ | 0.2833 | 0.03228 | $5.054 \times 10^{-7}$ | $5.613 \times 10^{-6}$ |
| 40    | $2.213 \times 10^{-11}$ | 0.1519 | 0.03207 | $1.974 \times 10^{-12}$ | $4.949 \times 10^{-9}$ |

Table III. Verification results using the IN method (Corollary 2).

| Order | $\eta$ | $\kappa$ | $K$ | $2K\eta$ | $\|u^* - u_h\|_{H^1}$ |
|-------|-------|---------|-----|---------|---------------------|
| 20    | $4.466 \times 10^{-6}$ | 0.08189 | 0.04365 | $3.899 \times 10^{-7}$ | $4.466 \times 10^{-6}$ |
| 40    | $2.281 \times 10^{-11}$ | 0.02383 | 0.03727 | $1.700 \times 10^{-12}$ | $2.281 \times 10^{-11}$ |

Table IV. Computational time [sec].

| Order | $u_h$ (dd) | $\|u_h\|_{L^\infty}$ (double) | $C_1, C_2, C_3, K$ (double) | $\delta$ (dd) |
|-------|-----------|-------------------------------|----------------------------|----------------|
| 20    | 0.828     | 1.648                         | 0.853                      | 0.206          |
| 40    | 46.932    | 10.766                        | 2.990                      | 1.379          |

Finally, we show the computational times in Table IV. The label in the column of Table IV indicates computed functions/constants and the data type. It can be seen that the main computational times are $u_h$ and $\|u_h\|_{L^\infty}$. For $u_h$, the data type is set to double to small the value of $\delta$, it takes time to calculate. Note that changing from double to double is very easy in the VCP library that use templates. However, the computation of $\|u_h\|_{L^\infty}$ requires the computation of a global optimization, i.e., we use `Generator.global_min` and `Generator.global_max` in the VCP library to compute $\|u_h\|_{L^\infty}$. Therefore, BLAS and Lapack cannot be used to compute $\|u_h\|_{L^\infty}$ because it is not a numerical linear algebra. Note that `Generator.global_min` and `Generator.global_max` in the VCP library are computed in parallel by OpenMP. In Table IV, BLAS and Lapack are used in the calculation of Lemma 2, which is included in the constant $C_1$. If $C_1, C_2, C_3, K$ and $\delta$ are obtained, Corollary 1 and Corollary 2 can be computed. Thus, the time for the verification procedure by Corollary 1 and Corollary 2 was almost negligible compared with other computational costs.

8.2 Example 2
Next, we present an example for a system of elliptic boundary value problems for the Lotka–Volterra equation with diffusion (e.g., [8, 22, 31]):

$$
\begin{cases}
-\Delta u = au - u^2 - cuv & \text{in } \Omega, \\
-\Delta v = bv + duv - v^2 & \text{in } \Omega, \\
u = v = 0 & \partial\Omega, 
\end{cases}
$$

(28)

where $\Omega = (0, 1)^2$, and $a, b, c, d \in \mathbb{R}$ are parameters. The source code can be downloaded from https://verified.computation.jp/testcode/NOLTA2019/Lotka_Volterra2.cpp

We set exact solutions $u^*$ and $v^*$ to $z^* = (u^*, v^*)^T$. We also set approximate solutions $u_h$ and $v_h$ to $z_h = (u_h, v_h)^T$. Additionally, the first and second Fréchet derivatives of $f$ at $z_h$ are

63
\[ f'[z_h] = \begin{pmatrix} a - 2u_h - cv_h \\ dv_h \end{pmatrix} \frac{-cu_h}{b + du_h - 2v_h} \] and \[ f''[z_h] = \begin{pmatrix} -2 \\ d \end{pmatrix}. \]

We set \( f'_V[u_h] \) and \( f'_X[u_h] \) defined by Definition 2 as

\[ f'_V[u_h] = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{and} \quad f'_X[u_h] = \begin{pmatrix} a - 2u_h - cv_h \\ dv_h \end{pmatrix} \frac{-cu_h}{b + du_h - 2v_h}. \]

Therefore, from

\[
\begin{align*}
\| \begin{pmatrix} a - 2u_h - cv_h \\ dv_h \end{pmatrix} \frac{-cu_h}{b + du_h - 2v_h} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \|_X &= \| \begin{pmatrix} a - 2u_h - cv_h \end{pmatrix}w_1 - cu_h w_2 \|_{L^2} \|_E \\
&\leq \| \begin{pmatrix} a - 2u_h - cv_h \end{pmatrix}w_1 + (b + du_h - 2v_h)w_2 \|_{L^2} \|_E \\
&\leq \| \begin{pmatrix} a - 2u_h - cv_h \|_{L^\infty} \\ \| b + du_h - 2v_h \|_{L^\infty} \end{pmatrix} \|_E \| \begin{pmatrix} w_1 \|_{L^2} \\ \| w_2 \|_{L^2} \end{pmatrix} \|_E , \forall (w_1, w_2)^T \in X,
\end{align*}
\]

constants \( C_{f'_V}, C_{f'_X}, C_{f'_X} \) defined by (23)-(25) are

\[ C_{f'_V} = C_{f'_X} = 0 \quad \text{and} \quad C_{f'_X} = \| \begin{pmatrix} \| a - 2u_h - cv_h \|_{L^\infty} \\ \| b + du_h - 2v_h \|_{L^\infty} \end{pmatrix} \|_E , \]

respectively.

Next, we consider the function \( K'_T(t, \| w \|_V) \) defined by (15). Using \( \tau, C_h \) and \( C_{s, A} \) defined by (22), (20), and (21), respectively, we obtain

\[
\begin{align*}
\left\| \begin{pmatrix} \| T^{-1}R_hA^{-1}f''[u_h + tv]w \|_{L(V)} \\ \| (I - R_h)A^{-1}f''[u_h + tv]w \|_{L(V)} \end{pmatrix} \right\|_E &= \left\| \begin{pmatrix} \| T^{-1}R_hA^{-1}f''[u_h + tv]w \|_{L(V)} \\ \| (I - R_h)A^{-1}f''[u_h + tv]w \|_{L(V)} \end{pmatrix} \right\|_E \\
&\leq \left\| \begin{pmatrix} \tau \\ C_h \end{pmatrix} \right\|_E \| f''[u_h + tv]w \|_{L(V,X)} \leq \left\| \begin{pmatrix} \tau \\ C_h \end{pmatrix} \right\|_E \left\| \begin{pmatrix} 2 \\ [d] \end{pmatrix} \right\|_{[2]} \| w \|_{L(V,X)} \\
&= \left\| \begin{pmatrix} \tau \\ C_h \end{pmatrix} \right\|_E \| f''[u_h + tv]w \|_{L(V,X)} \leq \left\| \begin{pmatrix} \tau \\ C_h \end{pmatrix} \right\|_E \left\| \begin{pmatrix} 2 \\ [d] \end{pmatrix} \right\|_{[2]} \sup_{\phi \in V \setminus \{0\}} \| \phi \|_{V} \\
&\leq C_{s, A}\sqrt{\tau^2 + C_h^2} \left( \begin{pmatrix} 2 \\ [d] \end{pmatrix} \right)_{[2]} \| w \|_{V}, \forall w \in V.
\end{align*}
\]

Therefore, from

\[ K'_T(t, \| w \|_V) = C_{s, A}\sqrt{\tau^2 + C_h^2} \left( \begin{pmatrix} 2 \\ [d] \end{pmatrix} \right)_{[2]} \| w \|_{V} , \]

\[ \int_0^1 K'_T(t, \cdot) dt \] determined by nonlinear term \( f(u) \) is

\[ \int_0^1 K'_T(t, \cdot) dt = 2C_{s, A}\sqrt{\tau^2 + C_h^2}.
\]

We set \( a = 110, b = 7, c = 100, d = -8 \). For this setting, the approximate solutions \( u_h \) and \( v_h \) were computed numerically, and the graphs are shown in Fig. 2.

The constants in Definition 1, Definition 2, and Lemma 1 were estimated as presented in Table V, where “Order” means the value of \( \text{uh.Order} \) and \( \text{vOrder} \) in the VCP library, that is, it is the order of the Legendre basis.

The FN method (Corollary 1) yielded the results presented in Table VI. For all “Order” in Table VI, verification failed because \( m \) was greater than 1. The value of \( m \) was close to the maximum values of \( C_1, C_2, \) and \( C_3 \). Therefore, it seems that validation failed because \( C_1 \) was greater than 1.
Fig. 2. Approximate solutions of (28). Left: $u_h$, Right: $v_h$.

Table V. Results for some constants.

| Order | $C_{s,2}$ | $C_{s,4}$ | $C_h$ | $\tau$ | $C_{f_{e}}^\prime$ | $C_1$ | $C_2$ | $C_3$ |
|-------|----------|----------|-------|-------|-----------------|-------|-------|-------|
| 20    | 0.2251   | 0.3184   | 0.02133 | 0.8934 | 235.9           | 4.493 | 1.132 | 0.1073 |
| 40    | 0.2251   | 0.3184   | 0.01151 | 0.8934 | 235.9           | 2.424 | 0.6106| 0.03120|
| 60    | 0.2251   | 0.3184   | 0.007877| 0.8934 | 235.9           | 1.660 | 0.4182| 0.01464|
| 80    | 0.2251   | 0.3184   | 0.005989| 0.8934 | 235.9           | 1.262 | 0.3180| 0.008461|

Table VI. Verification results using the FN method (Corollary 1).

| Order | $\eta$ ($= \delta$) | $m$ | $K$ | $2K\eta\left[1-m\eta\right]$ | $\|z^\ast - z_h\|_V$ |
|-------|---------------------|-----|-----|-----------------------------|-----------------|
| 20    | $3.329 \times 10^{-6}$ | 4.494 | 9.058 |                             |                |
| 40    | $8.403 \times 10^{-11}$ | 2.424 | 9.056 |                             |                |
| 60    | $2.545 \times 10^{-12}$ | 1.660 | 9.056 |                             |                |
| 80    | $4.146 \times 10^{-12}$ | 1.261 | 9.056 |                             |                |

Table VII. Verification results using the IN method (Corollary 2).

| Order | $\eta$ | $\kappa$ | $K$ | $2K\eta$ | $\|z^\ast - z_h\|_V$ |
|-------|-------|---------|-----|-----------|-----------------|
| 20    |       | 5.192   |     |           |                 |
| 40    |       | 1.511   |     |           |                 |
| 60    | $1.691 \times 10^{-11}$ | 0.7085 | 66.17 | $2.238 \times 10^{-9}$ | $1.692 \times 10^{-11}$ |
| 80    | $1.131 \times 10^{-11}$ | 0.4097 | 28.65 | $6.480 \times 10^{-10}$ | $1.131 \times 10^{-11}$ |

The IN method (Corollary 2) yielded the results presented in Table VII. For “Order” = 20, 40 in Table VII, verification failed because $\kappa$ was greater than 1. When “Order” was 60 and 80, $\kappa$ was less than 1. Moreover, because $2K\eta$ was also less than 1, we were able to show the existence of a solution. Additionally, the upper bound of norm $\|z^\ast - z_h\|_V$ was obtained.

Next, we consider the relationship between various constants that appear in Corollary 1 (FN method) and Corollary 2 (IN method). Recall that

\[
C_1 = C_h \tau C_{f_{e}}^\prime, \quad C_2 = C_h C_{\kappa^2} C_{f_{e}}^\prime, \quad C_3 = C_h^2 C_{f_{e}}^\prime, \\
m \approx \max(C_1, C_2, C_3) \quad \text{and} \quad \kappa = C_3 + C_1 C_2.
\]

Table VIII shows the convergence rate of each constant $C_1$, $C_2$, $C_3$, $m$, and $\kappa$. For example, $C_1$ in 40/20 means $2.424/4.493$ in Table V. The table shows that constants $C_1$ and $C_2$ have first-order convergence rates with respect to “Order.” By contrast, constant $C_3$ has a quadratic convergence rate.
Table VIII. Convergence rate of each constant $C_1$, $C_2$, $C_3$, $m$, and $\kappa$.

| Order/Order | $C_1$ | $C_2$ | $C_3$ | $m$ | $\kappa$ |
|-------------|-------|-------|-------|-----|---------|
| 20/20       | 1     | 1     | 1     | 1   | 1       |
| 40/20       | 0.5395| 0.5393| 0.2907| 0.5393| 0.2910  |
| 60/20       | 0.3694| 0.3694| 0.1364| 0.3693| 0.1364  |
| 80/20       | 0.2808| 0.2809| 0.07885| 0.2806| 0.07890 |

Table IX. Computational time [sec].

| Order | $u_h, v_h$ (dd) | $\|u_h\|_{L^\infty}, \|v_h\|_{L^\infty}$ (double) | $C_1, C_2, C_3, K$ (double) | $\delta$ (dd) |
|-------|-----------------|---------------------------------|----------------------------|----------------|
| 60    | 877472.755      | 9248.604                        | 141.288                    | 18.689         |
| 80    | 933797.991      | 20689.618                       | 889.380                    | 51.928         |

with respect to “Order.” This is related to the order of constant $C_h$ for each constant. Furthermore, constant $m$ has a first-order convergence rate because $m \approx \max(C_1, C_2, C_3)$. Moreover, the constant $\kappa$ has a quadratic convergence rate because $C_3$ and $C_1C_2$ have a quadratic convergence rate. Therefore, provided the convergence rates of the constants $C_1$, $C_2$, and $C_3$ do not change, the IN method is better than the FN method in terms of convergence rates of each constant.

Finally, we show the computational times in Table IX. The label in the column of Table IX indicates computed functions/constants and the data type. The main computational time is $u_h, v_h$, which creates an approximate solution. For $u_h$ and $v_h$, the data type is set to dd to small the value of $\delta$, it takes time to calculate. Note that changing from dd to double is very easy in the VCP library that use templates. In Table IX, BLAS and Lapack are used in the calculation of Lemma 2, which is included in the constant $C_1$. If $C_1$, $C_2$, $C_3$, $K$ and $\delta$ are obtained, Corollary 1 and Corollary 2 can be computed. Thus, the time for the verification procedure by Corollary 1 and Corollary 2 was almost negligible compared with other computational costs. Anyway we need fairly long computational time for Example 2. Further improvement of the computational technique and development of a new method will be necessary to improve it, which is our subject in the future.

9. Conclusions
In this paper, we reformulated the FN method and IN method as Corollary 1 and Corollary 2, respectively, using a convergence theorem for simplified Newton-like methods with direct product space $V_h \times V_1$. Theses corollaries only require the constants in Definition 1. Therefore, the FN method and IN method can be checked simultaneously.

Additionally, we provided the VCP library, which is a software library written in the C++ programming language. The VCP library was introduced as a software library for numerical verification methods of solutions to PDEs.

Finally, we presented numerical examples using the reformulated verification methods and VCP library. In the numerical experiments, Corollary 2(IN method) performed better than Corollary 1(FN method). However, in all cases, Corollary 2(IN method) was not shown to be better. Therefore, we believe that it is better to consider using both of Corollary 1(FN method) and Corollary 2(IN method) simultaneously.

Acknowledgments
This work was supported by JST CREST Grant Number JPMJCR14D4, and MEXT under the “Exploratory Issue on Post-K computer” project (Development of verified numerical computations and super high-performance computing environment for extreme research). The second author was supported by JSPS KAKENHI Grant Number 18K03434. We thank the editors and reviewers for providing useful comments that helped to improve the content of this manuscript.
Appendix

A. Deuflhard–Heindl–Yamamoto’s formulation of a convergence theorem for Newton-like methods

In this section, we show Deuflhard–Heindl–Yamamoto’s formulation [9, 48] of a convergence theorem for Newton-like methods.

**Theorem 3 (Original convergence theorem for Newton-like methods [9, 48])** Let \( \hat{u} \in V \) be given. Let \( B(u, \rho) \) be an open ball defined by \( B(u, \rho) := \{v \in V \mid \|u - v\|_V < \rho\} \), and let \( \tilde{B}(u, \rho) \) be a closure of \( B(u, \rho) \). Let \( \tilde{L}[v] \) be a given invertible linear operator, which is an approximation to \( F'[v] \), \( v \in V \). Let \( \eta > 0 \) be a positive constant that satisfies

\[
\|\tilde{L}[\hat{u}]^{-1}F(\hat{u})\|_V \leq \eta.
\]

Let \( D_0 \subset V \) be an open convex set, and assume that the nonlinear operator \( F \) is Fréchet differentiable in \( D_0 \). Let \( K > 0 \), \( L \geq 0 \), \( l \geq 0 \), \( M \geq 0 \), \( m \geq 0 \) be constants that satisfy

\[
\begin{align*}
&\|\tilde{L}[\hat{u}]^{-1} (F'[v] - F'[w])\|_{L(V)} \leq K\|v - w\|_V, \quad v, w \in D_0, \\
&\|\tilde{L}[\hat{u}]^{-1} (\tilde{L}[\hat{u}] - \tilde{L}[v])\|_{L(V)} \leq L\|\hat{u} - v\|_V + l, \quad v \in D_0, \\
&\|\tilde{L}[\hat{u}]^{-1} (\tilde{L}[v] - F'[v])\|_{L(V)} \leq M\|\hat{u} - v\|_V + m, \quad v \in D_0.
\end{align*}
\]

We set

\[
\sigma = \max \left( 1, \frac{L + M}{K} \right).
\]

If

\[
\begin{align*}
l + m &< 1, \\
2\sigma K\eta &\leq (1 - l - m)^2,
\end{align*}
\]

\[
\tilde{S} := \tilde{B} \left( \hat{u} - \tilde{L}[\hat{u}]F(\hat{u}), \frac{1 - l - m - \sqrt{(1 - l - m)^2 - 2\sigma K\eta}}{\sigma K} \right) \subset D_0
\]

holds, then there exists a solution \( u^* \in \tilde{S} \) of \( F(u^*) = 0 \). Furthermore, the solution \( u^* \) is unique in

\[
\tilde{S} = \begin{cases} 
B \left( \hat{u}, \frac{1 - m + \sqrt{(1 - m)^2 - 2K\eta}}{K} \right) \cap D_0 & (2\sigma K\eta < (1 - l - m)^2) \\
\tilde{B} \left( \hat{u}, \frac{1 - m + \sqrt{(1 - m)^2 - 2K\eta}}{K} \right) \cap D_0 & (2\sigma K\eta = (1 - l - m)^2)
\end{cases}
\]

B. Other Method: Infinite-dimensional Newton method without operator matrix \( H \). Compute \( \|F'[\hat{u}]^{-1}\| \leq K \).

Let \( \hat{L}^{-1} := F'[\hat{u}]^{-1} \). Then, for (9), we have

\[
\|I - F'[\hat{u}]^{-1}F'[\hat{u}]\|_{L(V)} = 0.
\]

Therefore, we have \( m = 0 \). However, we note that the evaluation is difficult because \( F'[\hat{u}] \) is an infinite-dimensional operator. Therefore, the calculation of \( \eta \) in (8) uses the following evaluation

\[
\|F'[\hat{u}]^{-1}F(\hat{u})\|_V \leq \|F'[\hat{u}]^{-1}\|_{L(V^*, V)}\|F(\hat{u})\|_{V^*}.
\]

Similarly, the calculation of \( K \) in (10) uses the following evaluation

\[
\|F'[\hat{u}]^{-1} (F'[\hat{u}] - F'[\hat{u} + v])\|_{L(V)} \leq \|F'[\hat{u}]^{-1}\|_{L(V^*, V)}\|F'[\hat{u}] - F'[\hat{u} + v]\|_{L(V, V^*)},
\]

(B-1)
In this method, we first need to prove that \( \mathcal{F}'[\hat{u}] \) has an inverse. In particular, we evaluate constant \( K \) that satisfies

\[
\|u\|_V \leq K\|\mathcal{F}'[\hat{u}]u\|_{V^*}, \quad \forall u \in V \tag{B-2}
\]
to show that \( \mathcal{F}'[\hat{u}] \) is injective. Because \( \Omega \) is a bounded domain and we assume Assumption 1, \( f'[\hat{u}] : V \to V^* \) is a compact operator. We note that

\[
\mathcal{A}^{-1}\mathcal{F}'[\hat{u}] = I - \mathcal{A}^{-1}f'[\hat{u}]
\]
and

\[
\mathcal{F}'[\hat{u}]\mathcal{A}^{-1} = I - f'[\hat{u}]\mathcal{A}^{-1}.
\]

Since \( \mathcal{A}^{-1} : V^* \to V \) is a bounded operator, \( \mathcal{A}^{-1}f'[\hat{u}] \) and \( f'[\hat{u}]\mathcal{A}^{-1} \) are also compact operators. Moreover, \( \mathcal{F}'[\hat{u}] \) is the Fredholm operator from [39, Lemma 2.4]. Thus, from the Fredholm alternative theorem, \( \mathcal{F}'[\hat{u}] \) is bijective. We note that constant \( K \) also satisfies \( \|\mathcal{F}'[\hat{u}]^{-1}\|_{L(V^*,V)} \leq K \). Therefore, the evaluation of constant \( K \) that satisfies (B-2) has been studied a long time. There are two methods to obtain constant \( K \):

- Reformulate the constant \( K \) to the eigenvalue problems.

  This method requires some assumptions. First, let the inner product of \( V \) be \( \langle u, v \rangle_V + \sigma(u, v)_X \) instead of \( \langle u, v \rangle_V \). If \( \mathcal{F}'[\hat{u}] \) is a self-adjoint operator, then the eigenvalue problem is \( \mathcal{F}'[\hat{u}]\psi = \lambda(\mathcal{A} + \sigma I)\psi \). Thus, if the absolute minimum eigenvalue \( \lambda_{\text{min}} \) is not equal to zero, then constant \( K \) is obtained as \( 1/|\lambda_{\text{min}}| \). The upper bounds of infinite-dimensional eigenvalues are easily obtained from the Rayleigh–Ritz bound. By contrast, it is difficult to obtain the lower bound of infinite-dimensional eigenvalues. For this purpose, we obtain rough lower bounds of the eigenvalues using Plum’s homotopy method and the Liu-Oishi theorem [23]. Furthermore, the lower bounds of sharp eigenvalues are obtained using the Lehmann–Goerisch method [12, 21]. Therefore, constant \( K \) can be obtained with high accuracy.

  However, if \( \mathcal{F}'[\hat{u}] \) is a non-self-adjoint operator, then the eigenvalue problem is \( \langle \mathcal{F}'[\hat{u}]\psi, \mathcal{F}'[\hat{u}]z \rangle = \lambda\langle (\mathcal{A} + \sigma I)\psi, (\mathcal{A} + \sigma I)z \rangle \). Therefore, the \( \langle A\psi, Az \rangle \) calculation is required.

- Reformulate the constant \( K \) to a linear elliptic problem \( \mathcal{F}'[\hat{u}]\psi = g \) for all \( g \in V^* \).

  This method was considered by the third author in 1995 [30]. Furthermore, it was improved by the second author from 2005 to 2019 [20, 26, 29, 44–46]. This method has recently been updated and generalized by the authors to Method II.

C. Matrix operator formula for the inverse operator for a general linear operator \( \mathcal{L} \)

In this section, we present a theorem that provides the inverse operator of a linear operator \( \mathcal{L} \) in general scenarios. See [40, Theorem 1] for the original idea. Therefore, the notation in this appendix is independent of other part in this paper.

Let \( \mathcal{X} \) and \( \mathcal{Y} \) be Banach spaces. Let \( \mathcal{L} : \mathcal{D}(\mathcal{L}) \subset \mathcal{X} \to \mathcal{Y} \) be a closed densely defined operator, and \( \mathcal{D}(\mathcal{L}) := \{ u \in \mathcal{X} \mid \mathcal{L}u \in \mathcal{Y} \} \). We determine a solution \( \psi \in \mathcal{D}(\mathcal{L}) \) that satisfies the linear equation

\[
\mathcal{L}\psi = g \tag{C-1}
\]
for a given \( g \in \mathcal{Y} \).

Let \( \mathcal{N} : \mathcal{X} \to \mathcal{Y} \) be a bounded linear operator. Let \( \mathcal{A} : \mathcal{D}(\mathcal{L}) \subset \mathcal{X} \to \mathcal{Y} \) be a closed densely defined operator that satisfies \( \mathcal{A} := \mathcal{L} + \mathcal{N} \), and we assume that operator \( \mathcal{A} \) has a bounded inverse operator from \( \mathcal{Y} \) to \( \mathcal{X} \). Note that the range of bounded inverse operator \( \mathcal{A}^{-1} \) is \( \mathcal{R}(\mathcal{A}^{-1}) = \mathcal{D}(\mathcal{L}) \). Let \( \mathcal{P}_h \) be a continuous projection on \( \mathcal{D}(\mathcal{L}) \) that satisfies \( \mathcal{P}_h = \mathcal{P}_h^2 \), and we denote the range by \( \mathcal{R}(\mathcal{P}_h) := \{ \mathcal{P}_h u \mid u \in \mathcal{D}(\mathcal{L}) \} \). Note that we do not require orthogonality \( \mathcal{P}_h = \mathcal{P}_h^* \). Because \( \mathcal{P}_h \) is a
continuous projection, $\mathcal{R}(P_h)$ and $\mathcal{R}(I-P_h)$ are closed subspaces of $\mathcal{D}(\mathcal{L})$, where $I$ denotes the identity operator on $\mathcal{D}(\mathcal{L})$. The range $\mathcal{R}(I-P_h)$ is a “supplémentaire topologique” of $\mathcal{R}(P_h)$, that is, $\mathcal{R}(I-P_h)$ is a closed subspace, $\mathcal{D}(\mathcal{L}) = \{u + v\mid u \in \mathcal{R}(P_h), v \in \mathcal{R}(I-P_h)\}$, and $\mathcal{R}(P_h) \cap \mathcal{R}(I-P_h) = \{0\}$. Thus, for $z \in \mathcal{D}(\mathcal{L})$, $u \in \mathcal{R}(P_h)$ and $v \in \mathcal{R}(I-P_h)$ that satisfy $z = u + v$ are unique.

We multiply $A^{-1}$ from the left on both sides of (C-1), and decompose the result into $\mathcal{R}(P_h)$ and $\mathcal{R}(I-P_h)$ parts as follows:

$$
\begin{align*}
\mathcal{L} & = \begin{pmatrix}
\mathcal{P}_h A^{-1} \mathcal{L} |_{\mathcal{R}(P_h)} & \mathcal{P}_h A^{-1} \mathcal{L} |_{\mathcal{R}(I-P_h)} \\
(I - \mathcal{P}_h) A^{-1} \mathcal{L} |_{\mathcal{R}(P_h)} & (I - \mathcal{P}_h) A^{-1} \mathcal{L} |_{\mathcal{R}(I-P_h)}
\end{pmatrix}
\begin{pmatrix}
\mathcal{P}_h \psi \\
(I - \mathcal{P}_h) \psi
\end{pmatrix} =
\begin{pmatrix}
\mathcal{P}_h A^{-1} g \\
(I - \mathcal{P}_h) A^{-1} g
\end{pmatrix}.
\end{align*}
$$

We set

$$
\begin{pmatrix}
T & B \\
C & D
\end{pmatrix} :=
\begin{pmatrix}
\mathcal{P}_h A^{-1} \mathcal{L} |_{\mathcal{R}(P_h)} & \mathcal{P}_h A^{-1} \mathcal{L} |_{\mathcal{R}(I-P_h)} \\
(I - \mathcal{P}_h) A^{-1} \mathcal{L} |_{\mathcal{R}(P_h)} & (I - \mathcal{P}_h) A^{-1} \mathcal{L} |_{\mathcal{R}(I-P_h)}
\end{pmatrix}
\begin{pmatrix}
\mathcal{P}_h \psi \\
(I - \mathcal{P}_h) \psi
\end{pmatrix} =
\begin{pmatrix}
\mathcal{P}_h A^{-1} g \\
(I - \mathcal{P}_h) A^{-1} g
\end{pmatrix}.
$$

and we assume that $T$ is bijective. Moreover, we set $S := D - CT^{-1}B$.

**Theorem 4** Assume that operator $T$ is bijective. If $S$ is bijective, then $\mathcal{L}$ is also bijective, and the solution $\psi \in \mathcal{D}(\mathcal{L})$ for Eq. (C-1) satisfies

$$
\begin{pmatrix}
\mathcal{P}_h \psi \\
(I - \mathcal{P}_h) \psi
\end{pmatrix} =
\begin{pmatrix}
(1 + T^{-1}B S^{-1}CT^{-1} - T^{-1}BS^{-1} \\
-S^{-1}CT^{-1}
\end{pmatrix}
\begin{pmatrix}
T & B \\
C & D
\end{pmatrix}
\begin{pmatrix}
I & 0 \\
0 & I_{\mathcal{R}(I-P_h)}
\end{pmatrix}
\begin{pmatrix}
\mathcal{P}_h A^{-1} g \\
(I - \mathcal{P}_h) A^{-1} g
\end{pmatrix}.
$$

**Proof** We first prove Eq. (C-2). From the definition $T : \mathcal{R}(P_h) \to \mathcal{R}(P_h)$, we have $TT^{-1} = T^{-1}T = I_{\mathcal{R}(P_h)}$, where $I_{\mathcal{R}(P_h)}$ means an identity operator on $\mathcal{R}(P_h)$. Similarly, from the definition $S : \mathcal{R}(I-P_h) \to \mathcal{R}(I-P_h)$, we have $SS^{-1} = S^{-1}S = I_{\mathcal{R}(I-P_h)}$, where $I_{\mathcal{R}(I-P_h)}$ denotes an identity operator on $\mathcal{R}(I-P_h)$. Thus, from

$$
\begin{pmatrix}
T & B \\
C & D
\end{pmatrix}
\begin{pmatrix}
T^{-1} + T^{-1}BS^{-1}CT^{-1} - T^{-1}BS^{-1} \\
-S^{-1}CT^{-1}
\end{pmatrix}
= \begin{pmatrix}
I_{\mathcal{R}(P_h)} & 0 \\
0 & I_{\mathcal{R}(I-P_h)}
\end{pmatrix}
\begin{pmatrix}
I_{\mathcal{R}(P_h)} & 0 \\
0 & I_{\mathcal{R}(I-P_h)}
\end{pmatrix}
\begin{pmatrix}
T & B \\
C & D
\end{pmatrix}
\begin{pmatrix}
T^{-1}B + T^{-1}BS^{-1}CT^{-1} - T^{-1}BS^{-1}D \\
-S^{-1}CT^{-1}B + S^{-1}D
\end{pmatrix}
\begin{pmatrix}
I & 0 \\
0 & I_{\mathcal{R}(I-P_h)}
\end{pmatrix}
\begin{pmatrix}
\mathcal{P}_h A^{-1} g \\
(I - \mathcal{P}_h) A^{-1} g
\end{pmatrix}
$$

and

$$
\begin{pmatrix}
T & B \\
C & D
\end{pmatrix}
\begin{pmatrix}
T^{-1} + T^{-1}BS^{-1}CT^{-1} - T^{-1}BS^{-1} \\
-S^{-1}CT^{-1}
\end{pmatrix}
= \begin{pmatrix}
I_{\mathcal{R}(P_h)} & 0 \\
0 & I_{\mathcal{R}(I-P_h)}
\end{pmatrix}
\begin{pmatrix}
I & 0 \\
0 & I_{\mathcal{R}(I-P_h)}
\end{pmatrix}
\begin{pmatrix}
CT^{-1} + CT^{-1}BS^{-1}CT^{-1} - CT^{-1}BS^{-1} + DS^{-1} \\
-D(CT^{-1}B)S^{-1}CT^{-1}
\end{pmatrix}
\begin{pmatrix}
I & 0 \\
0 & I_{\mathcal{R}(I-P_h)}
\end{pmatrix}
\begin{pmatrix}
\mathcal{P}_h A^{-1} g \\
(I - \mathcal{P}_h) A^{-1} g
\end{pmatrix}
\begin{pmatrix}
T & B \\
C & D
\end{pmatrix}
\begin{pmatrix}
T^{-1}B - T^{-1}BS^{-1}(D - CT^{-1}B) \\
S^{-1}(D - CT^{-1}B)
\end{pmatrix}
\begin{pmatrix}
I & 0 \\
0 & I_{\mathcal{R}(I-P_h)}
\end{pmatrix}
\begin{pmatrix}
\mathcal{P}_h A^{-1} g \\
(I - \mathcal{P}_h) A^{-1} g
\end{pmatrix}
$$

we have

$$
\begin{pmatrix}
T & B \\
C & D
\end{pmatrix}^{-1} = \begin{pmatrix}
T^{-1} + T^{-1}BS^{-1}CT^{-1} - T^{-1}BS^{-1} \\
-S^{-1}CT^{-1}
\end{pmatrix}
\begin{pmatrix}
I & 0 \\
0 & I_{\mathcal{R}(I-P_h)}
\end{pmatrix}
\begin{pmatrix}
\mathcal{P}_h A^{-1} g \\
(I - \mathcal{P}_h) A^{-1} g
\end{pmatrix}.
$$

Thus, we obtain Eq. (C-2).
We next show that linearized operator $L$ is bijective. To prove that $L$ is injective, we show that $\psi = 0$ is the only solution to the equation $L\psi = 0$. The solution $\psi$ can be decomposed uniquely as $\psi = P_h \psi + (I - P_h)\psi$ using continuous projection $P_h$. The solutions $P_h \psi$ and $(I - P_h)\psi$ are represented by (C-2), and $(P_h \psi, (I - P_h)\psi)^T = (0, 0)^T$ is obtained by substituting $g = 0$ into (C-2). Because operator matrix 

$$
\begin{pmatrix}
T & B \\
C & D
\end{pmatrix}
$$

has a bounded inverse, the solution $(P_h \psi, (I - P_h)\psi)^T = (0, 0)^T$ is unique. Thus, the solution $\psi = 0$ is also unique.

Finally, we prove that the linearized operator $L$ is surjective. For this, it is sufficient to show that there exists a solution $\phi \in \mathcal{D}(L)$ that satisfies $L\psi = g$ for any $g \in \mathcal{Y}$. Now, for $g \in \mathcal{Y}$, we define $(\psi_h, \psi_\perp)^T$ as the left-hand side of (C-2) and set $\psi := P_h \psi_h + (I - P_h)\phi$. Then, because the solution $(P_h \psi, (I - P_h)\psi)^T$ satisfies (C-2), there exists a solution $\psi \in \mathcal{D}(L)$ for any $g \in \mathcal{Y}$. Thus, operator $L$ is bijective.

\[\square\]

D. Functions for VCP’s matrix class

In this section, we present functions for VCP’s matrix class.

---

**Source code 3.** functions for VCP’s matrix class

```cpp
1  // (1) Matrix initialization
2  int n = 10;
3  int m = 5;
4  A.zeros(n); // Create an n*n zero matrix
5  A.zeros(n, m); // Create an n*m zero matrix
6  A.ones(n); // Create an n*n matrix with all elements 1
7  A.ones(n, m); // Create an n*m matrix with all elements 1
8  A.rand(n); // Create an n*n random matrix
9  A.rand(n, m); // Create an n*m random matrix
10  A.eye(n); // Create an n*n identity matrix
11  B.eye(n);
12
13  // (2) Obtain the matrix size
14  int row = A.rowsize();
15  int column = A.columnsize();
16
17  // (3) Access the elements
18  A(0,0) = 10;
19  A(5,3) = A(0,0);
20
21  // (4) Arithmetic operations
22  C = 1 + A; // For all elements
23  C = A + 2; // For all elements
24  C = A + B; // Matrix addition
25  C += A; // C = C + A
26
27  C = 1 - A; // For all elements
28  C = A - 2; // For all elements
29  C = A - B; // Matrix subtraction
30  C -= A; // C = C - A
31
32  C = 1 * A; // For all elements
33  C = A * 2; // For all elements
34  C = A * B; // Matrix multiplication
35  C *= A; // C = C*A
36  C = ltransmul(A); // C = transpose(A)*A
```

---
A = A + 1;
C = 1 / A; // For all elements
C = A / 2; // For all elements

// (5) Elementary functions, etc. (Element wise)
C = abs(A);
C = sqrt(A);
C = sin(A);
C = cos(A);
C = exp(A);
C = log(A);

// (6) MATLAB-like functions
C = sum(A);
C = diag(A);
C = transpose(A);
C = max(A);
C = min(A);
C = normone(A);
C = norminf(A);

// (7) Solve linear system Ax = b
A.rand(n);
b.ones(n,m);
b = A*b;
x = lss(A,b); // Find x s.t. Ax=b

// (8) Solve symmetric eigenvalue problem Ax = lambda x
A = transpose(A) + A;
eigsym(A, C); // Eigenvalue for the diagonal part of matrix C

// (9) Solve generalized symmetric eigenvalue problem Ax = lambda B x
B.rand(n);
A = transpose(A) + A;
B = ltransmul(B);
eigsymge(A, B, C); // Eigenvalue for the diagonal of matrix C

// (10) Display matrix
std::cout << "Matrix C = 
" << C << std::endl; //

// (11) Concatenate the matrix
D = vercat(A,B,C); // Vertically (similar to MATLAB’s [A;B;C])
D = horzcat(A,B,C); // Horizontally (similar to MATLAB’s [A,B,C])
C.rand(n);
A = vercat(D, horzcat(A,B,C));

// (12) Release the matrix
A.clear();

References
[1] CAPD DynSys library, 2014. http://capd.ii.uj.edu.pl
[2] J.P. Aubin, “Behavior of the error of the approximate solutions of boundary value problems
for linear elliptic operators by galerkin’s and finite difference methods,” Annali della Scuola
Normale Superiore di Pisa Classe di Scienze, vol. 21, no. 4, pp. 599–637, 1967.
[3] M. Berz, COSY INFINITY, 2000. https://bt.pa.msu.edu/
[4] M. Breden and R. Castelli, “Existence and instability of steady states for a triangular cross-
diffusion system: a computer-assisted proof,” Journal of Differential Equations, vol. 264, no. 10,
pp. 6418–6458, 2018.
[5] M. Breden, J.-P. Lessard, and M. Vanicat, “Global bifurcation diagrams of steady states of systems of pdes via rigorous numerics: a 3-component reaction-diffusion system,” *Acta applicandae mathematicae*, vol. 128, no. 1, pp. 113–152, 2013.

[6] S. Cai, K. Nagatou, and Y. Watanabe, “A numerical verification method for a system of fitzhugh-nagumo type,” *Numerical functional analysis and optimization*, vol. 33, no. 10, pp. 1195–1220, 2012.

[7] R. Castelli, “Rigorous computation of non-uniform patterns for the 2-dimensional gray-scott reaction-diffusion equation,” *Acta Applicandae Mathematicae*, vol. 151, no. 1, pp. 27–52, 2017.

[8] C. Cosner and A.C. Lazer, “Stable coexistence states in the volterra–lotka competition model with diffusion,” *SIAM Journal on Applied Mathematics*, vol. 44, no. 6, pp. 1112–1132, 1984.

[9] P. Deuflhard and G. Heinl, “Affine invariant convergence theorems for newton’s method and extensions to related methods,” *SIAM Journal on Numerical Analysis*, vol. 16, no. 1, pp. 1–10, 1979.

[10] J.-L. Figueras, M. Gameiro, J.-P. Lessard, and R. de la Llave, “A framework for the numerical computation and a posteriori verification of invariant objects of evolution equations,” *SIAM Journal on Applied Dynamical Systems*, vol. 16, no. 2, pp. 1070–1088, 2017.

[11] D. Gilbarg and N.S. Trudinger, *Elliptic partial differential equations of second order*, vol. 224, Springer-Verlag Berlin Heidelberg, reprint of the 1998 edition, 2001.

[12] F. Goerisch and Z. He, “Computer arithmetic and self-validating numerical methods,” chapter The Determination of Guaranteed Bounds to Eigenvalues with the Use of Variational Methods I, pp. 137–153, Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[13] A. Hungria, J.-P. Lessard, and J.D.M. James, “Rigorous numerics for analytic solutions of differential equations: the radii polynomial approach,” *Mathematics of Computation*, vol. 85, no. 299, pp. 1427–1459, 2016.

[14] Intel, Intel® Math Kernel Library, 2019. https://software.intel.com/en-us/mkl

[15] F. Johansson, “Arb: efficient arbitrary-precision midpoint-radius interval arithmetic,” *IEEE Transactions on Computers*, vol. 66, no. 8, pp. 1281–1292, 2017.

[16] L.V. Kantorovich and G.P. Akilov, *Functional Analysis*, Pergamon Press, Oxford, 1982.

[17] M. Kashiwagi, KV library, 2019. http://verifiedby.me/kv/

[18] F. Kikuchi and X. Liu, “Estimation of interpolation error constants for the p0 and p1 triangular finite elements,” *Computer methods in applied mechanics and engineering*, vol. 196, no. 37-40, pp. 3750–3758, 2007.

[19] S. Kimura and N. Yamamoto, “On explicit bounds in the error for the $H_1^0$-projection into piecewise polynomial spaces,” *Bulletin of informatics and cybernetics*, vol. 31, no. 2, pp. 109–115, 1999.

[20] T. Kinoshita, Y. Watanabe, and M.T. Nakao, “An alternative approach to norm bound computation for inverses of linear operators in hilbert spaces,” *Journal of Differential Equations*, vol. 266, no. 9, pp. 5431–5447, 2019.

[21] N.J. Lehmann, “Optimale eigenwerteinschließungen,” *Numerische Mathematik*, vol. 5, no. 1, pp. 246–272, 1963.

[22] A. Leung, “Equilibria and stabilities for competing-species reaction-diffusion equations with dirichlet boundary data,” *Journal of Mathematical Analysis and Applications*, vol. 73, no. 1, pp. 204–218, 1980.

[23] X. Liu and S. Oishi, “Verified eigenvalue evaluation for the laplacian over polygonal domains of arbitrary shape,” *SIAM Journal on Numerical Analysis*, vol. 51, no. 3, pp. 1634–1654, 2013.

[24] M.T. Nakao, “A numerical approach to the proof of existence of solutions for elliptic problems,” *Japan Journal of Applied Mathematics*, vol. 5, no. 2, pp. 313–332, 1988.

[25] M.T. Nakao, “A numerical approach to the proof of existence of solutions for elliptic problems ii,” *Japan Journal of Applied Mathematics*, vol. 7, no. 3, p. 477, 1990.

[26] M.T. Nakao, K. Hashimoto, and Y. Watanabe, “A numerical method to verify the invertibility of linear elliptic operators with applications to nonlinear problems,” *Computing*, vol. 75, no. 1, pp. 1–14, 2005.
[27] M.T. Nakao and Y. Watanabe, “An efficient approach to the numerical verification for solutions of elliptic differential equations,” *Numerical Algorithms*, vol. 37, no. 1-4, pp. 311–323, 2004.

[28] M.T. Nakao and Y. Watanabe, “Numerical verification methods for solutions of semilinear elliptic boundary value problems,” *Nonlinear Theory and Its Applications, IEICE*, vol. 2, no. 1, pp. 2–31, 2011.

[29] M.T. Nakao, Y. Watanabe, T. Kinoshita, T. Kimura, and N. Yamamoto, “Some considerations of the invertibility verifications for linear elliptic operators,” *Japan Journal of Industrial and Applied Mathematics*, vol. 32, no. 1, pp. 19–31, 2015.

[30] S. Oishi, “Numerical verification of existence and inclusion of solutions for nonlinear operator equations,” *Journal of Computational and Applied Mathematics*, vol. 60, no. 1, pp. 171–185, 1995.

[31] C.-V. Pao, “Coexistence and stability of a competition—diffusion system in population dynamics,” *Journal of Mathematical Analysis and Applications*, vol. 83, no. 1, pp. 54–76, 1981.

[32] M. Plum, “Bounds for eigenvalues of second-order elliptic differential operators,” *Zeitschrift für angewandte Mathematik und Physik ZAMP*, vol. 42, no. 6, pp. 848–863, 1991.

[33] M. Plum, “Enclosures for weak solutions of nonlinear elliptic boundary value problems,” *Inequalities and Applications*, pp. 505–521, World Scientific, 1994.

[34] M. Plum, “Computer-assisted enclosure methods for elliptic differential equations,” *Linear Algebra and its Applications*, vol. 324, no. 1, pp. 147–187, 2001.

[35] M. Plum, “Existence and multiplicity proofs for semilinear elliptic boundary value problems by computer assistance,” *Jahresbericht der Deutschen Mathematiker Vereinigung*, vol. 110, no. 1, pp. 19–54, 2008.

[36] M. Plum, “Computer-assisted proofs for semilinear elliptic boundary value problems,” *Japan Journal of Industrial and Applied Mathematics*, vol. 26, no. 2-3, pp. 419–442, 2009.

[37] MPFR Project, The GNU MPFR Library, 2019. https://www.mpfr.org/

[38] S.M. Rump, “INTLAB – INTerval LABoratory,” In Tibor Csendes, editor, *Developments in Reliable Computing*, pp. 77–104, Kluwer Academic Publishers, Dordrecht, 1999. http://www.ti3.tuhh.de/rump/

[39] M. Schechter, “Basic theory of fredholm operators,” *Annali della Scuola Normale Superiore di Pisa-Classe di Scienze*, vol. 21, no. 2, pp. 261–280, 1967.

[40] K. Sekine, M.T. Nakao, and S. Oishi, “A new formulation using the Schur complement for the numerical existence proof of solutions to elliptic problems: without direct estimation for an inverse of the linearized operator,” *Numer. Math.*, 146, pp. 907–926, 2020.

[41] A. Takayasu, X. Liu, and S. Oishi, “Verified computations to semilinear elliptic boundary value problems on arbitrary polygonal domains,” *NOLTA*, vol. 4, no. 1, pp. 34–61, 2013.

[42] K. Tanaka, K. Sekine, M. Mizuguchi, and S. Oishi, “Sharp numerical inclusion of the best constant for embedding $H_0^1(\Omega) \hookrightarrow L^p(\Omega)$ on bounded convex domain,” *Journal of Computational and Applied Mathematics*, vol. 311, pp. 306–313, 2017.

[43] Y. Watanabe, “A numerical verification method for two-coupled elliptic partial differential equations,” *Japan Journal of Industrial and Applied Mathematics*, vol. 26, no. 2-3, pp. 233–247, 2009.

[44] Y. Watanabe, T. Kinoshita, and M. Nakao, “A posteriori estimates of inverse operators for boundary value problems in linear elliptic partial differential equations,” *Mathematics of Computation*, vol. 82, no. 283, pp. 1543–1557, 2013.

[45] Y. Watanabe, T. Kinoshita, and M.T. Nakao, “An improved method for verifying the existence and bounds of the inverse of second-order linear elliptic operators mapping to dual space,” *Journal of Industrial and Applied Mathematics*, pp. 1–14, 2019.

[46] Y. Watanabe, K. Nagatou, M. Plum, and M.T. Nakao, “Norm bound computation for inverses of linear operators in hilbert spaces,” *Journal of Differential Equations*, vol. 260, no. 7, pp. 6363–6374, 2016.

[47] N. Yamamoto and M.T. Nakao, “Numerical verifications for solutions to elliptic equations using residual iterations with a higher order finite element,” *Journal of Computational and Applied Mathematics*, vol. 60, no. 1-2, pp. 271–279, 1995.

[48] T. Yamamoto, “Historical developments in convergence analysis for newton’s and newton-like
Methods,” *Journal of Computational and Applied Mathematics*, vol. 124, no. 1-2, pp. 1–23, 2000.

[49] K. Yosida, *Functional analysis*, Springer–Verlag Berlin Heidelberg, reprint of the sixth (1980) edition, 1995.

[50] E. Zeidler and P.R. Wadsack, *Nonlinear functional analysis and its applications: Part I: Fixed-point theorems /transl. by Peter R. Wadsack*, Springer-Verlag, 1986.