East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.2023-065.050423

Vol. **13**, No. 3, pp. 435-463
August 2023

The Random Feature Method for Time-Dependent Problems

Jing-Run Chen[1,2], Weinan E[3] and Yi-Xin Luo[1,2,*]

[1] *School of Mathematical Sciences, University of Science and Technology of China, Hefei 230026, China.*
[2] *Suzhou Institute for Advanced Research, University of Science and Technology of China, Suzhou 215123, China.*
[3] *AI for Science Institute, Beijing and Center for Machine Learning Research and School of Mathematical Sciences, Peking University.*

Received 21 February 2023; Accepted (in revised version) 5 April 2023.

Dedicated to Professor Tao Tang on the occasion of his 60th birthday.

Abstract. We present a framework for solving time-dependent partial differential equations (PDEs) in the spirit of the random feature method. The numerical solution is constructed using a space-time partition of unity and random feature functions. Two different ways of constructing the random feature functions are investigated: feature functions that treat the spatial and temporal variables (STC) on the same footing, or functions that are the product of two random feature functions depending on spatial and temporal variables separately (SoV). Boundary and initial conditions are enforced by penalty terms. We also study two ways of solving the resulting least-squares problem: the problem is solved as a whole or solved using the block time-marching strategy. The former is termed the space-time random feature method (ST-RFM). Numerical results for a series of problems show that the proposed method, i.e. ST-RFM with STC and ST-RFM with SoV, have spectral accuracy in both space and time. In addition, ST-RFM only requires collocation points, not a mesh. This is important for solving problems with complex geometry. We demonstrate this by using ST-RFM to solve a two-dimensional wave equation over a complex domain. The two strategies differ significantly in terms of the behavior in time. In the case when block time-marching is used, we prove a lower error bound that shows an exponentially growing factor with respect to the number of blocks in time. For ST-RFM, we prove an upper bound with a sublinearly growing factor with respect to the number of subdomains in time. These estimates are also confirmed by numerical results.

AMS subject classifications: 65M20, 65M55, 65M70

Key words: Time-dependent PDEs, partition of unity method, random feature method, collocation method, separation-of-variables random features.

---

[*] Corresponding author. *Email addresses:* `jingrunchen@ustc.edu.cn` (J.-R. Chen), `weinan@math.pku.edu.cn` (W.-N. E), `seeing@mail.ustc.edu.cn` (Y.-X. Luo)

1. Introduction

Time-dependent partial differential equations (PDEs), such as diffusion equation, wave equation, Maxwell equation, and Schrödinger equation, are widely used for modeling the dynamic evolution of physical systems. Numerical methods, including finite difference method [11], finite element methods [17], and spectral methods [15], have been proposed to solve these PDEs. Despite the great success in theory and application, these methods still face some challenges, to name a few, complex geometry, mesh generation, and possibly high dimensionality.

Along another line, the success of deep learning in computer vision and natural language processing [8] attracts great attention in the community of scientific computing. As a special class of functions, neural networks are proved to be universal approximators to continuous functions [3]. Many researchers seek for solving ordinary and partial differential equations with neural networks [5–7, 9, 14, 16, 19]. Since the PDE solution can be defined in the variational (if exists), strong, and weak forms, deep Ritz method [5], deep Galerkin method [16] and physics-informed neural networks [14], and weak adversarial network [19] are proposed using loss (objective) functions in the variational, strong, and weak forms, respectively. Deep learning-based algorithms have now made it fairly routine to solve a large class of PDEs in high dimensions without the need for mesh generation of any kind.

For low-dimensional problems, traditional methods are accurate, with reliable error control, stability analysis and affordable cost. However, in practice, coming up with a suitable mesh is often a highly non-trivial task, especially for complex geometry. On the contrary, machine-learning methods are mesh-free and only collocation points are needed. Even for low-dimensional problems, this point is still very attractive. What bothers a user is the absence of reliable error control in machine-learning methods. For example, without an exact solution, the numerical approximation given by a machine-learning method does not show a clear trend of convergence as the number of parameters increases.

There are some efforts to combine the merits of traditional methods and deep-learning based methods. The key ingredient is to replace deep neural networks by a special class of two-layer neural networks with the inner parameters fixed, known as random features [12, 13] or extreme learning machine [10]. Random feature functions are proved to be universal approximators as well, meanwhile only the parameters of the output layer need to be optimized, leading to a convex optimization problem. Extreme learning machines are employed to solve ordinary and partial differential equations in [18] and [1], respectively. Spectral accuracy is obtained for problems with analytic solutions, and the simplicity of network architectures reduces the training difficulty in terms of execution time and solution accuracy, compared to deep neural networks. In [4], a special kind of partition of unity (PoU), termed as domain decomposition, is combined with extreme learning machines to approximate the PDE solution and the block time-marching strategy is proposed for long time simulations. Spectral accuracy is obtained in both space and time for analytic solutions, but the error grows exponentially fast in most cases as the simulation time increases. In [2], combining PoU and random feature functions, the random feature method (RFM) is proposed to solve static PDEs with complex geometries. An automatic rescaling strategy is also proposed to balance the weights of equations and boundary conditions, which is found to work well for linear elasticity and Stokes flow over complex geometries.

The objective in this article is to propose a methodology for solving time-dependent PDEs that shares the merits of both traditional and machine learning-based algorithms. This new class of algorithms can be made spectrally accurate in both space and time. Meanwhile, they are also mesh-free, making them easy to use even in settings with complex geometry. Our starting point is based on a combination of rather simple and well-known ideas: We use space-time PoU and random feature functions to represent the approximate solution, the collocation method to take care of the PDE as well as the boundary conditions in the least-squares sense, and a rescaling procedure to balance the contributions from the PDE and initial/boundary conditions in the loss function. This method is called ST-RFM. For time-dependent problems, a random feature function can depend on both spatial and temporal variables — i.e. space-time concatenation input (STC), or is the product of two random feature functions depending on spatial and temporal variables separately (SoV). STC can be viewed as natural extensions of random feature method for static problems [2], while SoV may be a better choice for some time-dependent PDEs. Both STC and SoV are proved to be universal approximators. For long time intervals, PoU for the temporal variable — i.e. domain decomposition along the time direction, is proposed to solve time-dependent problems. Error estimates of the ST-RFM and the block-time marching strategy are provided. ST-RFM yields spectrally accurate results with slowly growing error in terms of the number of subdomains in time, while the error generated by the block time-marching strategy in [4] grows exponentially fast in terms of the number of blocks. These findings are confirmed by numerical results in one and two dimensions.

This article is organized as follows. In Section 2, we present the ST-RFM and prove the approximation property of STC and SoV. Upper bound error estimate for ST-RFM and lower bound error estimate for the block time-marching strategy are provided. In Section 3, numerical experiments in one and two dimensions for heat, wave, and Schrödinger equations are conducted to show the spectral accuracy of the proposed method, and confirm the error estimates for long time simulations. Application of ST-RFM to a two-dimensional wave equation over a complex geometry is also provided. Conclusions are drawn in Section 4.

2. Random Feature Method for Solving Time-Dependent PDEs

For completeness, we first recall the RFM for solving static problems. We then introduce the ST-RFM for solving time-dependent PDEs, and prove the universal approximation property and the error estimate.

2.1. Random feature method for static problems

Let $\boldsymbol{x} \in \Omega \subset \mathbb{R}^{d_x}$, where $d_x \in \mathbb{N}^+$ is the dimension of $\boldsymbol{x}$, and let $d_u \in \mathbb{N}^+$ be the dimension of output. Consider the following boundary-value problem:

$$\mathscr{L}\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega,$$

$$\mathscr{B}\boldsymbol{u}(\boldsymbol{x})=\boldsymbol{g}(\boldsymbol{x}),\quad \boldsymbol{x}\in\partial\Omega,$$

where $\boldsymbol{f}$ and $\boldsymbol{g}$ are known functions, $\mathscr{L}$ and $\mathscr{B}$ are differential and boundary operators, respectively. $\partial\Omega$ is the boundary of Ω.

The RFM has the following components. First, N points $\{\hat{\boldsymbol{x}}_i\}$ are chosen from Ω, typically uniformly distributed. Then Ω is decomposed to N subdomains $\{\Omega_i\}$ with $\hat{\boldsymbol{x}}_i\in\Omega_i$. Thus, we have $\Omega\subset\cup_i\Omega_i$. For each Ω_i, a PoU function ψ_i with support Ω_i, i.e. $\operatorname{supp}(\psi_i)=\Omega_i$, is constructed. In one dimension, two PoU functions are commonly used

$$\begin{aligned}&\psi_a(x)=\mathbb{I}_{[-1,1]}(x),\\&\psi_b(x)=\mathbb{I}_{[-5/4,-3/4]}(x)\frac{1+\sin(2\pi x)}{2}+\mathbb{I}_{[-3/4,3/4]}(x)+\mathbb{I}_{[3/4,5/4]}(x)\frac{1-\sin(2\pi x)}{2}.\end{aligned}$$

The first PoU is discontinuous, while the second one is continuously differentiable. In high dimensions, the PoU function ψ can be constructed as a tensor product of d_x one-dimensional PoU functions ψ, i.e. $\psi(\boldsymbol{x})=\Pi_{i=1}^{d_x}\psi(x_i)$.

Then for each Ω_i, RFM constructs $J_n\in\mathbb{N}^+$ random features ϕ_{ij} by a two-layer neural network with random but fixed parameters $\boldsymbol{k}_{ij}$ and b_{ij}, i.e.

$$\phi_{ij}(\boldsymbol{x})=\sigma\left(\boldsymbol{k}_{ij}^\top l_i(\boldsymbol{x})+b_{ij}\right),\quad j=1,2,\ldots,J_n,\tag{2.1}$$

where the nonlinear activation function σ is chosen as the hyperbolic tangent or trigonometric functions in [2]. In (2.1), each component of $\boldsymbol{k}_{ij}$ and b_{ij} is independently sampled from $\mathbb{U}(-R_m,R_m)$, where $R_m\in\mathbb{R}^+$ controls the magnitude of parameters. In particular, both the weights and the biases are fixed in the optimization procedure. Moreover, $l_i(\boldsymbol{x})$ is a linear transformation to transform inputs in Ω_i to $[-1,1]^{d_x}$. The approximate solution in RFM is the linear combination of these random features together with the PoU

$$u_M(\boldsymbol{x})=\sum_{i=1}^N\psi_i(\boldsymbol{x})\sum_{j=1}^{J_n}u_{ij}\phi_{ij}(\boldsymbol{x}),\tag{2.2}$$

where $u_{ij}\in\mathbb{R}$ are unknown coefficients to be sought and $M=N\cdot J_n$ denotes the degree of freedom. For vectorial solutions, we approximate each component of the solution by (2.2), i.e.

$$\boldsymbol{u}_M(\boldsymbol{x})=\left(u_M^1(\boldsymbol{x}),\ldots,u_M^{d_u}(\boldsymbol{x})\right)^\top.\tag{2.3}$$

To find the optimal set of parameters $\{u_{ij}\}$, RFM evaluates the original problem on collocation points and formulates a least-squares problem. To be specific, RFM samples $Q\in\mathbb{N}^+$ collocation points $\boldsymbol{x}_{i,1},\ldots,\boldsymbol{x}_{i,Q}$ in each Ω_i, and computes the rescaling parameters $\lambda_{i,k,q}^p>0$ and $\lambda_{i,k,e}^b>0$ for $i\in[N]$, $k\in[d_u]$, $q\in[Q]$ and $e\in[Q]$ satisfying $\boldsymbol{x}_{i,e}\in\partial\Omega$. Let $\lambda_{i,q}^p=\operatorname{diag}(\lambda_{i,1,q}^p,\ldots,\lambda_{i,d_u,q}^p)^\top$ and $\lambda_{i,e}^b=\operatorname{diag}(\lambda_{i,1,e}^b,\ldots,\lambda_{i,d_u,e}^b)^\top$. Then the random feature method minimizes

$$\operatorname{Loss}\big(\{u_{i,j,k}\}\big)=\sum_{i=1}^N\left(\sum_{q=1}^Q\left\|\lambda_{i,q}^p\left(\mathscr{L}\boldsymbol{u}_M(\boldsymbol{x}_{i,q})-\boldsymbol{f}(\boldsymbol{x}_{i,q})\right)\right\|_2^2\right.$$

$$+\sum_{\boldsymbol{x}_{i,e}\in\partial\Omega}\left\|\lambda_{i,e}^{b}\left(\mathscr{B}\boldsymbol{u}_M(\boldsymbol{x}_{i,e})-\boldsymbol{g}(\boldsymbol{x}_{i,e})\right)\right\|_2^2\Bigg), \tag{2.4}$$

where $\boldsymbol{u}_M$ is of the form (2.3) and $u_{i,j,k}$ is the k-th coefficient of the random feature ϕ_{ij}.

The above problem (2.4) is a linear least-squares problem when $\mathscr{L}$ and $\mathscr{B}$ are linear operators. Moreover, when the discontinuous PoU ψ_a is used, continuity conditions between adjacent subdomains must be imposed by adding regularization terms in (2.4), while no regularization is required when ψ_b is used for second-order equations. By minimizing (2.4), the optimal coefficients $\boldsymbol{u}^* = (u_{ijk}^*,)^\top$ are obtained and the numerical solution is constructed by (2.3).

2.2. Space-time random feature method

Now, we consider time-dependent PDEs of the following form with the final time $T > 0$:

$$\begin{aligned} \mathscr{L}\boldsymbol{u}(\boldsymbol{x},t) &= \boldsymbol{f}(\boldsymbol{x},t), & \boldsymbol{x},t &\in \Omega\times[0,T], \\ \mathscr{B}\boldsymbol{u}(\boldsymbol{x},t) &= \boldsymbol{g}(\boldsymbol{x},t), & \boldsymbol{x},t &\in \partial\Omega\times[0,T], \\ \mathscr{I}\boldsymbol{u}(\boldsymbol{x},0) &= \boldsymbol{h}(\boldsymbol{x}), & \boldsymbol{x} &\in \Omega, \end{aligned} \tag{2.5}$$

where $\boldsymbol{f}$, $\boldsymbol{g}$ and $\boldsymbol{h}$ are known functions and $\mathscr{I}$ is the initial operator.

Following the same routine as in RFM, we construct a partition of $\Omega\times[0,T]$. First, we decompose the spatial domain Ω to N_x subdomains where each subdomain Ω_i contains a central point $\hat{\boldsymbol{x}}_i$. We then decompose the temporal interval $[0,T]$ to N_t subdomains — i.e.

$$[0,T) = [t_0,t_1)\cup[t_1,t_2),\ldots,[t_{N_t-1},t_{N_t}),$$

where each subdomain contains a central point $\hat{t}_i = (t_{i-1}+t_i)/2$. The product of the PoUs in space and in time results in the space-time PoU — i.e.

$$\psi_{i_x,i_t}(\boldsymbol{x},t) = \psi_{i_x}(\boldsymbol{x})\psi_{i_t}(t),$$

where i_x and i_t are indices for spatial and temporal subdomains, respectively.

Next, we generalize spatial random features (2.1) to space-time random features. There are two options. The first can be viewed as a natural extension of (2.1), where the concatenation of spatial and temporal variables is fed into ϕ_{ij} as the input and the output is a random feature function — i.e.

$$\phi_{i_x,i_t,j}(\boldsymbol{x},t) = \sigma\left(\left(\boldsymbol{k}_{i_x,i_t,j}^{x}\right)^\top l_{i_x}(\boldsymbol{x}) + k_{i_x,i_t,j}^{t} l_{i_t}(t) + b_{i_x,i_t,j}\right). \tag{2.6}$$

Here $\boldsymbol{k}_{i_x,i_t,j}^{x}$ and $k_{i_x,i_t,j}^{t}$ are the weights associated with spatial and temporal inputs, respectively. $b_{i_x,i_t,j}$ is the bias, $l_{i_x}(\boldsymbol{x})$ and $l_{i_t}(t)$ are linear transformations from $\boldsymbol{x}\in\Omega_{i_x}$ to $[-1,1]^{d_x}$ and from $t\in[t_{i_t-1},t_{i_t}]$ to $[-1,1]$, respectively. (2.6) is called STC.

The second option is to use separation of variables, which mimics the technique of separation of variables for solving PDEs

$$\phi_{i_x,i_t,j}(\boldsymbol{x},t) = \sigma\left(\left(\boldsymbol{k}_{i_x,i_t,j}^{x}\right)^\top l_{i_x}(\boldsymbol{x}) + b_{i_x,i_t,j}^{x}\right)\sigma\left(k_{i_x,i_t,j}^{t} l_{i_t}(t) + b_{i_x,i_t,j}^{t}\right). \tag{2.7}$$

In this formulation, the space-time random feature is a product of the spatial random feature and the temporal random feature, thus we term it as SoV. For both random features, the degrees of freedom $M = N_x \cdot N_t \cdot J_n$.

The combination of these space-time random features and PoU leads to the following approximate solution in the scalar case:

$$u_M(\boldsymbol{x},t) = \sum_{i_x=1}^{N_x} \sum_{i_t=1}^{N_t} \psi_{i_x}(\boldsymbol{x}) \psi_{i_t}(t) \sum_{j=1}^{J_n} u_{i_x,i_t,j} \phi_{i_x,i_t,j}(\boldsymbol{x},t), \tag{2.8}$$

where $u_{i_x,i_t,j}$ is the unknown coefficient to be sought. A vectorial solution $\boldsymbol{u}_M$ can be formulated in the same way as that in (2.3), i.e.

$$\boldsymbol{u}_M(\boldsymbol{x},t) = \left(u_M^1(\boldsymbol{x},t), \ldots, u_M^{d_u}(\boldsymbol{x},t) \right)^\top. \tag{2.9}$$

For the k-th component of $\boldsymbol{u}_M$, we denote the coefficient associated to $\phi_{i_x,i_t,j}$ by $u_{i_x,i_t,j,k}$.

By substituting (2.8) or (2.9) into (2.5), we define the loss function on collocation points. Denote $\{\boldsymbol{x}_{i_x,i_t,1}, \ldots, \boldsymbol{x}_{i_x,i_t,Q_x}\}$ spatial collocation points in the i_x-th spatial subdomain and $(i_t - 1)T/N_t = t_{i_x,i_t} = t_{i_x,i_t,0} < t_{i_x,i_t,1} < \cdots < t_{i_x,i_t,Q_t} = i_t T/N_t$ temporal collocation points in the i_t-th temporal subdomain, respectively. In RFM, we define the rescaling parameters $\lambda^e_{i_x,i_t,q_x,q_t}$, $\lambda^b_{i_x,i_t,e,q_t}$ and $\lambda^i_{i_x,i_t,q_x}$ for $i_x \in [N_x]$, $i_t \in [N_t]$, $q_x \in [Q_x]$, $q_t \in [Q_t]$ and $e \in [Q_x]$ satisfying $\boldsymbol{x}_{i_x,i_t,e} \in \partial\Omega$. Then the ST-RFM minimizes

$$\begin{aligned}
&\mathrm{Loss}\big(\{u_{i_x,i_t,j,k}\}\big) \\
&= \sum_{i_x=1}^{N_x} \sum_{i_t=1}^{N_t} \Bigg(\sum_{q_x=1}^{Q_x} \sum_{q_t=1}^{Q_t} \left\| \lambda^e_{i_x,i_t,q_x,q_t} \left(\mathscr{L}\boldsymbol{u}(\boldsymbol{x}_{i_x,i_t,q_x}, t_{i_x,i_t,q_t}) - \boldsymbol{f}(\boldsymbol{x}_{i_x,i_t,q_x}, t_{i_x,i_t,q_t}) \right) \right\|^2 \\
&\qquad + \sum_{\boldsymbol{x}_{i_x,i_t,e} \in \partial\Omega} \sum_{q_t=1}^{Q_t} \left\| \lambda^b_{i_x,i_t,e,q_t} \left(\mathscr{B}\boldsymbol{u}(\boldsymbol{x}_{i_x,i_t,e}, t_{i_x,i_t,q_t}) - \boldsymbol{g}(\boldsymbol{x}_{i_x,i_t,e}, t_{i_x,i_t,q_t}) \right) \right\|^2 \\
&\qquad + \mathbb{I}_{i_t=1} \sum_{q_x=1}^{Q_x} \left\| \lambda^i_{i_x,i_t,q_x} \left(\mathscr{I}\boldsymbol{u}(\boldsymbol{x}_{i_x,i_t,q_x}, 0) - \boldsymbol{h}(\boldsymbol{x}_{i_x,i_t,q_x}) \right) \right\|^2 \Bigg)
\end{aligned} \tag{2.10}$$

to find the numerical solution of the form (2.9). Moreover, when $N_x > 1$ or $N_t > 1$ and ψ_a is used, continuity conditions between adjacent spatial (temporal) subdomains must be imposed by adding regularization terms in (2.10), while no regularization is required when ψ_b is used for second-order equations.

For better illustration of this method, we consider an example with $d_u = 1$, the initial operator $\mathscr{I}$ being identity and rescaling parameters being one. For convenience, we set $N_x = 1$ and relabel i_t by i without confusion. Under these settings, we introduce two matrix-valued functions $\boldsymbol{\Phi}_i(t)$, $\mathbf{L}_i(t)$ and three induced matrices as follows:

$$\begin{aligned}
&\boldsymbol{\Phi}_i(t) = \big(\phi_{i,j}(\boldsymbol{x}_{i,q}, t),\big) \in \mathbb{R}^{Q_x \times J_n}, \qquad \boldsymbol{\Phi}_{i,0} = \boldsymbol{\Phi}_i(t_{i,0}), \quad \boldsymbol{\Phi}_{i,1} = \boldsymbol{\Phi}_i(t_{i,Q_t}), \\
&\mathbf{L}_i(t) = \big(\mathscr{L}\phi_{i,j}(\boldsymbol{x}_{i,q}, t),\big) \in \mathbb{R}^{Q_x \times J_n}, \quad \mathbf{L}_i = \big[\mathbf{L}_i(t_{i,0})^\top, \ldots, \mathbf{L}_i(t_{i,Q_t-1})^\top\big]^\top,
\end{aligned}$$

where q is the row index and j is the column index. Then we construct the matrix $\mathbf{A}$ as follows:

$$\mathbf{A}_i=\left[\mathbf{0}_{(J_n\times(i-1)(Q_t+1)Q_x)},\boldsymbol{\Phi}_{i,0}^\top,\mathbf{L}_i^\top,-\boldsymbol{\Phi}_{i,1}^\top,\mathbf{0}_{(J_n\times((N_t-i)(Q_t+1)-1)Q_x)}\right]^\top,\quad \mathbf{A}=\left[\mathbf{A}_1,\ldots,\mathbf{A}_{N_t}\right].$$

Define the following vectors:

$$\begin{aligned}
\boldsymbol{h}&=\left[h(\boldsymbol{x}_{1,1}),h(\boldsymbol{x}_{1,2}),\ldots,h(\boldsymbol{x}_{1,Q_x})\right]^\top,\\
\boldsymbol{f}_{i,j}&=\left[f(\boldsymbol{x}_{i,1},t_{i,j}),f(\boldsymbol{x}_{i,2},t_{i,j}),\ldots,f(\boldsymbol{x}_{i,Q_x},t_{i,j})\right]^\top,\\
\boldsymbol{f}_i&=\left[\boldsymbol{f}_{i,0}^\top,\boldsymbol{f}_{i,1}^\top,\ldots,\boldsymbol{f}_{i,Q_t-1}^\top\right]^\top,
\end{aligned}$$

then we construct the vector $\boldsymbol{b}=\sum_{i=1}^{N_t}\boldsymbol{b}_i$, where

$$\begin{aligned}
\boldsymbol{b}_1&=\left[\boldsymbol{h}^\top,\boldsymbol{f}_1^\top,\mathbf{0}^\top_{((N_t-1)(Q_t+1)Q_x)}\right]^\top,\\
\boldsymbol{b}_i&=\left[\mathbf{0}^\top_{(((i-1)(Q_t+1)+1)Q_x)},\boldsymbol{f}_i^\top,\mathbf{0}^\top_{((N_t-i)(Q_t+1)Q_x)}\right]\quad\text{for}\quad i=2,\ldots,N_t.
\end{aligned}$$

Let $\boldsymbol{u}\in\mathbb{R}^{N_tJ_n}$, then the optimal coefficient $\boldsymbol{u}^S$ by ST-RFM is obtained by

$$\boldsymbol{u}^S=\min_{\boldsymbol{u}}\|(\mathbf{A}\boldsymbol{u}-\boldsymbol{b})\|^2. \tag{2.11}$$

For time-dependent partial differential equations, Dong *et al.* [4] proposed the block time-marching strategy. Block time-marching strategy solves Eq. (2.5) in each time block individually and applies the numerical results in i-th block at the terminal time as the initial conditions of $(i+1)$-th block. Let $N_b\in\mathbb{N}^+$ be the number of time blocks, and let $N_t=1$ for simplicity. For first-order equations in time, we present the detailed process of block time-marching strategy in Algorithm 2.1, and the optimal coefficient $\boldsymbol{u}^B$ is the concatenation of optimal coefficients in all time blocks.

Algorithm 2.1 Block time-marching strategy to solve time-dependent PDEs.

```
$\tilde{\mathbf{A}}_1=\left[\boldsymbol{\Phi}_{1,0}^\top,\boldsymbol{L}_1^\top\right]^\top$, $\tilde{\boldsymbol{b}}_1=\left[\boldsymbol{g}^\top,\boldsymbol{f}_1^\top\right]^\top$;
$\boldsymbol{u}_1^B=\min_{\boldsymbol{u}}\|\tilde{\mathbf{A}}_1\boldsymbol{u}-\tilde{\boldsymbol{b}}_1\|^2$;
for $i=2,3,\ldots,N_b$ do
    $\tilde{\mathbf{A}}_i=\left[\boldsymbol{\Phi}_{i,0}^\top,\boldsymbol{L}_i^\top\right]^\top$, $\tilde{\boldsymbol{b}}_i=\left[\left(\boldsymbol{u}_{i-1}^B\right)^\top\boldsymbol{\Phi}_{i-1,1}^\top,\boldsymbol{f}_i^\top\right]^\top$;
    $\boldsymbol{u}_i^B=\min_{\boldsymbol{u}}\|\tilde{\mathbf{A}}_i\boldsymbol{u}-\tilde{\boldsymbol{b}}_i\|^2$;
end for
$\boldsymbol{u}^B=\left[\left(\boldsymbol{u}_1^B\right)^\top,\left(\boldsymbol{u}_2^B\right)^\top,\ldots,\left(\boldsymbol{u}_{N_b}^B\right)^\top\right]^\top$.
```

DOI https://doi.org/10.4208/eajam.2023-065.050423 | Generated on 2024-12-18 23:56:09

2.3. Approximation properties

Both concatenation random features and separation-of-variables random features are universal approximators. To prove these, we first recall the definition of sigmoidal functions in [3].

Definition 2.1. *A function σ is called sigmoidal if* $\lim_{z\to+\infty}\sigma(z)=1$ *and* $\lim_{z\to-\infty}\sigma(z)=0$.

The approximation property of STC is given in the following theorem.

Theorem 2.1. *Let σ be any continuous sigmoidal function. Then finite sums of*

$$G_C(\boldsymbol{x},t)=\sum_{j=1}^{N}u_j\sigma\left(\left(\boldsymbol{k}_j^x\right)^\top\boldsymbol{x}+k_j^t t+b_j\right) \tag{2.12}$$

are dense in $C(\Omega\times[0,T])$. In other words, given any $f\in C(\Omega\times[0,T])$ and $\epsilon>0$, there exists a sum $G_C(x,t)$ such that

$$|G_C(\boldsymbol{x},t)-f(\boldsymbol{x},t)|<\epsilon \quad \textit{for all} \quad \boldsymbol{x},t\in\Omega\times[0,T].$$

Proof. It is a direct consequence of [3, Theorem 2]. ☐

Corollary 2.1. *When using the hyperbolic tangent as the activation function in* (2.12)*, the finite sums* (2.12) *are dense in $C(\Omega\times[0,T])$.*

Proof. Considering the continuous sigmoidal function $\tilde{\sigma}(z)=1/(1+e^{-z})$, we note that $\tanh(z)=2\tilde{\sigma}(2z)-1$. From Theorem 2.1, there is a sum

$$\tilde{G}_J(\boldsymbol{x},t)=\sum_{j=1}^{\tilde{N}}\tilde{u}_j\tilde{\sigma}\left(\left(\tilde{\boldsymbol{k}}_j^x\right)^\top\boldsymbol{x}+\tilde{k}_j^t t+\tilde{b}_j\right)$$

such that

$$\left|\tilde{G}_J(\boldsymbol{x},t)-f(\boldsymbol{x},t)\right|<\frac{\epsilon}{2}.$$

Define $\hat{u}=(\sum_{j=1}^{\tilde{N}}\tilde{u}_j)/2$. If $\hat{u}=0$, then the finite sum

$$G_C(\boldsymbol{x},t)=\sum_{j=1}^{\tilde{N}}\frac{1}{2}\tilde{u}_j\tanh\left(\frac{1}{2}\left(\left(\tilde{\boldsymbol{k}}_j^x\right)^\top\boldsymbol{x}+\tilde{k}_j^t t+\tilde{b}_j\right)\right)$$

satisfies

$$\begin{aligned}\left|G_C(\boldsymbol{x},t)-f(\boldsymbol{x},t)\right|&=\left|\sum_{j=1}^{\tilde{N}}\frac{1}{2}\tilde{u}_j\left(2\tilde{\sigma}\left(\left(\tilde{\boldsymbol{k}}_j^x\right)^\top\boldsymbol{x}+\tilde{k}_j^t t+\tilde{b}_j\right)-1\right)-f(\boldsymbol{x},t)\right|\\&=\left|\tilde{G}_J(\boldsymbol{x},t)-f(x,t)\right|<\frac{\epsilon}{2}<\epsilon.\end{aligned}$$

If $\hat{u} \neq 0$, since $\Omega \subset \mathbb{R}^{d_x}$ is compact and thus is bounded and closed, there exist $\hat{\boldsymbol{k}}^x$, $\hat{k}^t$ and $\hat{b}$ such that

$$\left|\tanh\left(\left(\hat{\boldsymbol{k}}^x\right)^\top \boldsymbol{x} + \hat{k}_t t + \hat{b}\right) - 1\right| < \frac{\epsilon}{|\hat{u}|}$$

holds for all $(\boldsymbol{x}, t) \in \Omega \times [0, T]$. Then the finite sum

$$G_C(\boldsymbol{x}, t) = \sum_{j=1}^{\tilde{N}} \frac{1}{2} \tilde{u}_j \tanh\left(\frac{1}{2}\left(\left(\tilde{\boldsymbol{k}}_j^x\right)^\top \boldsymbol{x} + \tilde{k}_j^t t + \tilde{b}_j\right)\right) + \frac{1}{2} \hat{u} \tanh\left(\left(\hat{\boldsymbol{k}}^x\right)^\top \boldsymbol{x} + \hat{k}^t t + \hat{b}\right)$$

satisfies

$$\begin{aligned} |G_C(\boldsymbol{x}, t) - f(\boldsymbol{x}, t)| = & \left| \sum_{j=1}^{\tilde{N}} \frac{1}{2} \tilde{u}_j \left(2\tilde{\sigma}\left(\left(\tilde{\boldsymbol{k}}_j^x\right)^\top \boldsymbol{x} + \tilde{k}_j^t t + \tilde{b}_j\right) - 1\right) \right. \\ & \left. \quad + \frac{1}{2} \hat{u} \tanh\left(\left(\hat{\boldsymbol{k}}^x\right)^\top \boldsymbol{x} + \hat{k}^t t + \hat{b}\right) - f(\boldsymbol{x}, t) \right| \\ \leq & \left|\tilde{G}_J(\boldsymbol{x}, t) - f(\boldsymbol{x}, t)\right| + \frac{1}{2} |\hat{u}| \left|\tanh\left(\left(\hat{\boldsymbol{k}}^x\right)^\top \boldsymbol{x} + \hat{k}^t t + \hat{b}\right) - 1\right| \\ < & \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon \end{aligned}$$

for all $(\boldsymbol{x}, t) \in \Omega \times [0, T]$. ☐

For SoV, we first extend the definition of discriminatory functions in [3] as follows.

Definition 2.2. *σ is said to be discriminatory if for a measure $\mu \in \mathscr{M}(\Omega \times [0, T])$*

$$\int_{\Omega \times [0, T]} \sigma\left((\boldsymbol{k}^x)^\top \boldsymbol{x} + b^x\right) \sigma(k^t t + b^t) \mathrm{d}\mu(\boldsymbol{x}, t)$$

for all $\boldsymbol{k}^x \in \mathbb{R}^{d_x}$, $k^t \in \mathbb{R}$, $b^x \in \mathbb{R}$ and $b^t \in \mathbb{R}$ implies $\mu = 0$.

Lemma 2.1. *Any bounded, measurable sigmoidal function σ is discriminatory. In particular, any continuous sigmoidal function is discriminatory.*

Proof. For any $\boldsymbol{x}, t, b$, $\boldsymbol{k} = ((\boldsymbol{k}^x)^\top, k_t)^\top$, $\theta = (\theta^x, \theta^t)$, we have

$$\lim_{\lambda \to +\infty} \sigma\left(\lambda\left((\boldsymbol{k}^x)^\top \boldsymbol{x} + b\right) + \theta_x\right) = \begin{cases} 1, & \text{if} \quad (\boldsymbol{k}^x)^\top \boldsymbol{x} + b > 0, \\ 0, & \text{if} \quad (\boldsymbol{k}^x)^\top \boldsymbol{x} + b < 0 \end{cases}$$

and

$$\sigma\left(\lambda\left((\boldsymbol{k}^x)^\top \boldsymbol{x} + b\right) + \theta_x\right) = \sigma(\theta_x), \quad \text{if} \quad (\boldsymbol{k}^x)^\top \boldsymbol{x} + b = 0, \quad \lambda \in \mathbb{R}.$$

Thus, $\sigma_{x,\lambda}(\boldsymbol{x}) = \sigma(\lambda((\boldsymbol{k}^x)^\top \boldsymbol{x} + b) + \theta_x)$ converges in the pointwise and bounded sense to

$$\gamma_x(\boldsymbol{x}) = \begin{cases} 1, & \text{if} \quad (\boldsymbol{k}^x)^\top \boldsymbol{x} + b > 0, \\ 0, & \text{if} \quad (\boldsymbol{k}^x)^\top \boldsymbol{x} + b < 0, \\ \sigma(\theta_x), & \text{if} \quad (\boldsymbol{k}^x)^\top \boldsymbol{x} + b = 0 \end{cases}$$

as $\lambda \to +\infty$. Similarly, we have $\sigma_{t,\lambda}(t) = \sigma(\lambda(y_t t + \theta) + \phi_t)$ converges in the pointwise and bounded sense to

$$\gamma_t(t) = \begin{cases} 1, & \text{if} \quad k^t t + b > 0, \\ 0, & \text{if} \quad k^t t + b < 0, \\ \sigma(\theta_t), & \text{if} \quad k^t t + b = 0 \end{cases}$$

as $\lambda \to +\infty$.

Denote

$$\begin{aligned} \Pi^x_{k,b} &= \left\{(\boldsymbol{x}, t) \mid (\boldsymbol{k}^x)^\top \boldsymbol{x} + b = 0,\ k^t t + b > 0\right\}, \\ \Pi^t_{k,b} &= \left\{(\boldsymbol{x}, t) \mid (\boldsymbol{k}^x)^\top \boldsymbol{x} + b > 0,\ k^t t + b = 0\right\}. \end{aligned}$$

Let $I_{k,b}$ be the hyperline defined by $\{(\boldsymbol{x}, t) \mid (\boldsymbol{k}^x)^\top \boldsymbol{x} + b = 0,\ k^t t + b = 0\}$ and $H_{k,b}$ be the space defined by $\{(\boldsymbol{x}, t) \mid (\boldsymbol{k}^x)^\top \boldsymbol{x} + b > 0,\ k^t t + b > 0\}$. Then by the Lebesgue Bounded Convergence Theorem, we have

$$\begin{aligned} 0 &= \int_{\Omega\times[0,T]} \sigma_{x,\lambda}(\boldsymbol{x})\sigma_{t,\lambda}(t) \mathrm{d}\mu(\boldsymbol{x}, t) = \int_{\Omega\times[0,T]} \gamma_x(\boldsymbol{x})\gamma_t(t) \mathrm{d}\mu(\boldsymbol{x}, t) \\ &= \mu(H_{k,b}) + \sigma(\theta_x)\mu\left(\Pi^x_{k,b}\right) + \sigma(\theta_t)\mu\left(\Pi^t_{k,b}\right) + \sigma(\theta_x)\sigma(\theta_t)\mu(I_{k,b}) \end{aligned}$$

for all $\theta, b, \boldsymbol{k}$.

We now show that the measure of all quarter spaces being zero implies that the measure μ itself must be zero. This would be trivial if μ was a positive measure but here it is not.

For a fixed $\boldsymbol{k}$ and a bounded measurable function h, we define a linear functional F as

$$F(h) = \int_{\Omega\times[0,T]} h\left((\boldsymbol{k}^x)^\top \boldsymbol{x}\right) h(k^t t) \mathrm{d}\mu(\boldsymbol{x}, t).$$

Note that F is a bounded functional on $L^\infty(\mathbb{R})$ since μ is a finite signed measure. Set h the indicator function of the interval $[\theta, \infty)$, i.e.

$$h(u) = \begin{cases} 1, & \text{if} \quad u \geq 0, \\ 0, & \text{if} \quad u < \theta, \end{cases}$$

then

$$\begin{aligned} F(h) &= \int_{\Omega\times[0,T]} h\left((\boldsymbol{k}^x)^\top \boldsymbol{x}\right) h(k^t t) \mathrm{d}\mu(\boldsymbol{x}, t) \\ &= \mu(I_{k,-b}) + \mu\left(\Pi^x_{k,-b}\right) + \mu\left(\Pi^t_{k,-b}\right) + \mu(H_{k,-b}) = 0. \end{aligned}$$

Similarly, $f(h) = 0$ if h is the indicator function of the open interval (θ, ∞). By linearity, $F(h) = 0$ holds for the indicator function of any interval and for any simple function (sum of indicator functions of intervals). Since simple functions are dense in $L^\infty(\mathbb{R})$, $F = 0$.

In particular, a substitution of bounded measurable functions $s(\boldsymbol{x},t)=\sin(\boldsymbol{m}_x^\top\boldsymbol{x}+m_t t)$ and $c(\boldsymbol{x},t)=\cos(\boldsymbol{m}_x^\top\boldsymbol{x}+m_t t)$ gives

$$\begin{aligned}F(c+is)&=\int_{\Omega\times[0,T]}\cos\left(\boldsymbol{m}_x^\top\boldsymbol{x}+m_t t\right)+i\sin\left(\boldsymbol{m}_x^\top\boldsymbol{x}+m_t t\right)\mathrm{d}\mu(x,t)\\&=\int_{\Omega\times[0,T]}\exp\left(i\left(\boldsymbol{m}_x^\top\boldsymbol{x}+m_t t\right)\right)\mathrm{d}\mu(\boldsymbol{m},t)=0\end{aligned}$$

for all $\boldsymbol{m}=(\boldsymbol{m}_x,m_t)$. The Fourier transform of μ is zero and we have $\mu=0$. Hence, σ is discriminatory. ☐

Remark 2.1. Following the same routine, we can prove the approximation property of SoV. For any continuous sigmoidal function σ, the finite sums

$$G_S(\boldsymbol{x},t)=\sum_{j=1}^N u_j\sigma\left(\left(\boldsymbol{k}_j^x\right)^\top\boldsymbol{x}+b_j^x\right)\sigma\left(k^t t+b_j^t\right) \tag{2.13}$$

are dense in $C(\Omega\times[0,T])$.

Corollary 2.2. *For the hyperbolic tangent activation function in* (2.13)*, the finite sums of the form*

$$G_S(\boldsymbol{x},t)=\sum_{j=1}^N u_j\tanh\left(\left(\boldsymbol{k}_j^x\right)^\top\boldsymbol{x}+b_j^x\right)\tanh\left(k_j^t t+b_j^t\right)$$

are dense in $C(\Omega\times[0,T])$.

Proof. The proof is similar to that of Corollary 2.1. ☐

2.4. Error estimates of space-time random feature method

For convenience, we set $N_b=N_t$ to analyze the error of block time-marching strategy and the ST-RFM. Let $\boldsymbol{u}^S$ and $\boldsymbol{u}^B$ be exact solutions in Eq. (2.11) and the block time-marching strategy in Algorithm 2.1, and $\hat{\boldsymbol{u}}^S$ and $\hat{\boldsymbol{u}}^B$ be numerical solutions by a numerical solver, such as direct or iterative methods, respectively. Then we have the following results.

Theorem 2.2. *Assume that for any linear least-squares problem* $\min_{\boldsymbol{u}}\|\mathbf{A}\boldsymbol{u}-\boldsymbol{b}\|^2$ *in ST-RFM and block time-marching strategy, there exists* $\boldsymbol{u}^*$ *such that* $\mathbf{A}\boldsymbol{u}^*=\boldsymbol{b}$. *Then* $\boldsymbol{u}^B=\boldsymbol{u}^S$.

Proof. Under the assumption, for each subproblem in Algorithm 2.1, the optimal solution $\boldsymbol{u}_i^B$ satisfies

$$\tilde{\mathbf{A}}_i\boldsymbol{u}_i^B=\tilde{\boldsymbol{b}}_i\quad\text{for}\quad i=1,2,\ldots,N_t.$$

Recalling the definition of $\mathbf{A}_i$ in ST-RFM, we have

$$\mathbf{A}_i\boldsymbol{u}_i^B=\left[\mathbf{0}^\top_{((i-1)(Q_t+1)Q_x)},\tilde{\boldsymbol{b}}_i^\top,-\left(\boldsymbol{u}_i^B\right)^\top\boldsymbol{\Phi}_{i,1}^\top,\mathbf{0}^\top_{(((N_t-i)(Q_t+1)-1)Q_x)}\right]^\top$$

$$
\begin{aligned}
&= \left[\mathbf{0}^\top_{((i-1)(Q_t+1)Q_x)}, \left(\boldsymbol{u}^B_{i-1}\right)^\top \boldsymbol{\Phi}^\top_{i-1,1}, \boldsymbol{f}^\top_i, -\left(\boldsymbol{u}^B_i\right)^\top \boldsymbol{\Phi}^\top_{i,1}, \mathbf{0}^\top_{(((N_t-i)(Q_t+1)-1)Q_x)}\right]^\top \\
&= \left[\mathbf{0}^\top_{((i-1)(Q_t+1)Q_x)}, \left(\boldsymbol{u}^B_{i-1}\right)^\top \boldsymbol{\Phi}^\top_{i-1,1}, \mathbf{0}^\top_{(((N_t-i+1)(Q_t+1)-1)Q_x)}\right]^\top \\
&\quad + \left[\mathbf{0}^\top_{(((i-1)(Q_t+1)+1)Q_x)}, \boldsymbol{f}^\top_i, \mathbf{0}^\top_{((N_t-i)(Q_t+1)Q_x)}\right]^\top \\
&\quad - \left[\mathbf{0}^\top_{(i(Q_t+1)Q_x)}, \left(\boldsymbol{u}^B_i\right)^\top \boldsymbol{\Phi}^\top_{i,1}, \mathbf{0}^\top_{(((N_t-i)(Q_t+1)-1)Q_x)}\right]^\top.
\end{aligned}
$$

After some algebraic calculations, we have

$$
\mathbf{A}\boldsymbol{u}^B = \sum_{i=1}^{N_t} \mathbf{A}_i \boldsymbol{u}^B_i = \boldsymbol{b}.
$$

Since the optimal solution of linear least-squares problem is unique under the assumption, we have $\boldsymbol{u}^B = \boldsymbol{u}^S$. $\square$

From Theorem 2.2, solving the time-dependent PDEs by the block time-marching strategy is equivalent to solving the same problem by the ST-RFM. In practice, however, the numerical solution $\hat{\boldsymbol{u}}$ is different from the optimal solution $\boldsymbol{u}$, and we denote the difference by δu. For long time intervals, when N_b and N_t increases by one, the solution error at N_b+1 block or N_t subdomain is expected to be greater than previous blocks or subdomains due to error accumulation. To quantitatively analyze the error propagation in terms of N_t and N_b, we first introduce a matrix $\mathbf{B}_i$ as

$$
\mathbf{B}_i = \left(\tilde{\mathbf{A}}_i^\top \tilde{\mathbf{A}}_i\right)^{-1} \boldsymbol{\Phi}^\top_{i,0} \boldsymbol{\Phi}_{i-1,1} \in \mathbb{R}^{J_n \times J_n} \quad \text{for} \quad i = 1, \ldots, N_t.
$$

For simplicity, random feature functions are set to be the same over different time subdomains — i.e.

$$
\phi_{i_1,j} = \phi_{i_2,j} \quad \text{for all} \quad 1 \le i_1 \ne i_2 \le N_t, \quad j = 1, 2, \ldots, J_n.
$$

Then all $\mathbf{B}_i$ are the same and can be denoted by $\mathbf{B}$. To proceed, we need the following assumption for $\mathbf{B}$.

Assumption 2.1. $\mathbf{B}$ is diagonalizable.

Remark 2.2. Although this assumption cannot be proved, we find that numerically $\mathbf{B}$ is diagonalizable for all the numerical examples we have tested. This is not a bad assumption since it provides a lower bound error estimate of the block time-marching strategy and is verified numerically as well.

Under Assumption 2.1, there exists $\{(\lambda_1, \boldsymbol{b}_1), \ldots, (\lambda_{J_n}, \boldsymbol{b}_{J_n})\}$ such that $\boldsymbol{b}_k$ is the eigenvector of $\mathbf{B}$ with eigenvalue λ_k and $\{\boldsymbol{b}_k\}$ is a unitary orthogonal basis in $\mathbb{R}^{J_n}$. Denote the eigenvalue with the largest modulus by λ_m, i.e. $\lambda_m = \max(|\lambda_k|)$. Since $\{\boldsymbol{b}_k\}$ forms a basis, there exists $\{\delta u_k\}_{k=1}^{J_n}$ such that $\delta u = \sum_{k=1}^{J_n} \delta u_k \boldsymbol{b}_k$, where δu_k, $k = 1, \ldots, J_n$ are independent and identically distributed random variables with $\mathbb{E}[\delta u_k] = 0$, $\mathbb{E}[|\delta u_k|] = \mu > 0$ and $\mathbb{E}[(\delta u_k)^2] = \delta^2 > 0$.

We need the following lemma to bound a geometric series from above and from below.

Lemma 2.2. *For any* $\lambda \in \mathbb{C}$ *with* $|\lambda| \neq 1$ *and* $\epsilon \in (0,1)$*, there exists* $N \in \mathbb{N}^+$ *such that*

$$(1+\epsilon)\frac{C_\lambda^n}{|1-\lambda|} \geq \left|\sum_{i=0}^{n-1}\lambda^i\right| \geq (1-\epsilon)\frac{C_\lambda^n}{|1-\lambda|}$$

holds for $n > N$*, where* $C_\lambda = \max(1,|\lambda|)$*.*

Proof. When $|\lambda| < 1$, we have

$$\lim_{n\to\infty}\left|\sum_{i=0}^{n-1}\lambda^i\right||1-\lambda| = \lim_{n\to\infty}\left|\frac{1-\lambda^n}{1-\lambda}\right||1-\lambda| = \lim_{n\to\infty}|1-\lambda^n| = 1.$$

Therefore, there exists $N_1 \in \mathbb{N}^+$ such that

$$1+\epsilon \geq \left|\sum_{i=0}^{n-1}\lambda^i\right||1-\lambda| \geq 1-\epsilon$$

holds for $n > N_1$. Similarly, when $|\lambda| > 1$, we have

$$\lim_{n\to\infty}\left|\sum_{i=0}^{n-1}\lambda^i\right|\frac{|1-\lambda|}{|\lambda|^n} = \lim_{n\to\infty}\left|\frac{1-\lambda^n}{1-\lambda}\right|\frac{|1-\lambda|}{|\lambda|^n} = \lim_{n\to\infty}|\lambda^{-n}-1| = 1.$$

Therefore, there exists $N_2 \in \mathbb{N}^+$ such that

$$1+\epsilon \geq \left|\sum_{i=0}^{n-1}\lambda^i\right|\frac{|1-\lambda|}{|\lambda^n|} \geq 1-\epsilon.$$

Thus, the conclusion holds when $N = \max(N_1,N_2)$. □

Now we are ready to characterize the difference between $\hat{\boldsymbol{u}}^B$ and $\boldsymbol{u}^B$ under the perturbation δu.

Lemma 2.3. *Under a perturbation* δu *to the solution, there exist* $N \in \mathbb{N}^+$ *and* $\alpha > 0$ *such that when* $N_t > N$*, the solution difference* $\|\hat{\boldsymbol{u}}^B - \boldsymbol{u}^B\|$ *by block time-marching strategy satisfies*

$$\mathbb{E}_{\delta u}\left[\|\hat{\boldsymbol{u}}^B - \boldsymbol{u}^B\|\right] \geq \alpha\mu\sqrt{N_t}\beta(N_t),$$

where

$$\beta(x) = \begin{cases} 1, & \textit{if} \quad \max(|\lambda_k|) \leq 1 \quad \text{and} \quad 1 \notin \{\lambda_k\}, \\ x, & \textit{if} \quad \max(|\lambda_k|) = 1 \quad \text{and} \quad 1 \in \{\lambda_k\}, \\ |\lambda_m|^{x/2}, & \textit{if} \quad \max(|\lambda_k|) > 1. \end{cases}$$

Proof. The difference between $\hat{\boldsymbol{u}}_i^B$ and $\boldsymbol{u}_i^B$ satisfies

$$\hat{\boldsymbol{u}}_i^B - \boldsymbol{u}_i^B = \delta u + \left(\tilde{\mathbf{A}}_i^\top\tilde{\mathbf{A}}_i\right)^{-1}\tilde{\mathbf{A}}_i^\top\hat{\boldsymbol{b}}_i - \boldsymbol{u}_i^B$$

$$
\begin{aligned}
&= \delta u + \left(\tilde{\mathbf{A}}_i^\top \tilde{\mathbf{A}}_i\right)^{-1}\tilde{\mathbf{A}}_i^\top \left(\tilde{\boldsymbol{b}}_i + \left[\boldsymbol{\Phi}_{i-1,1}^\top, \mathbf{0}_{(J_n\times Q_tQ_x)}^\top\right]^\top \left(\hat{\boldsymbol{u}}_{i-1}^B - \boldsymbol{u}_{i-1}^B\right)\right) - \boldsymbol{u}_i^B \\
&= \delta u + \mathbf{B}\big(\hat{\boldsymbol{u}}_{i-1}^B - \boldsymbol{u}_{i-1}^B\big) \\
&= \delta u + \mathbf{B}\left(\delta u + \mathbf{B}\big(\hat{\boldsymbol{u}}_{i-2}^B - \boldsymbol{u}_{i-2}^B\big)\right) = \cdots = \left(\sum_{j=0}^{i-1}\mathbf{B}^j\right)\delta u,
\end{aligned}
$$

where $\hat{\boldsymbol{b}}_i \neq \tilde{\boldsymbol{b}}_i$ due to $\hat{\boldsymbol{u}}_{i-1}^B - \boldsymbol{u}_{i-1}^B \neq \mathbf{0}$. Notice that $\{\boldsymbol{b}_k\}$ is an orthonormal basis of $\mathbb{R}^{J_n}$. Using the relations $\mathbf{B}^j\boldsymbol{b}_k = \lambda_k\mathbf{B}^{j-1}\boldsymbol{b}_k = \cdots = \lambda_k^j\boldsymbol{b}_k$ and the unitary orthogonality of $\{\boldsymbol{b}_k\}$, we obtain

$$
\begin{aligned}
\|\hat{\boldsymbol{u}}^B - \boldsymbol{u}^B\|^2 &= \sum_{i=1}^{N_t}\|\hat{\boldsymbol{u}}_i^B - \boldsymbol{u}_i^B\|^2 = \sum_{i=1}^{N_t}\left\|\sum_{k=1}^{J_n}\sum_{j=0}^{i-1}\delta u_k\lambda_k^j\boldsymbol{b}_k\right\|^2 \\
&= \sum_{i=1}^{N_t}\sum_{k=1}^{J_n}\left|\sum_{j=0}^{i-1}\lambda_k^j\right|^2\delta u_k^2 \geq \sum_{i=1}^{N_t}\left|\sum_{j=0}^{i-1}\lambda_m^j\right|^2\delta u_m^2.
\end{aligned}
$$

The lower bound can be estimated for four different cases.

1. If $|\lambda_m| < 1$, then Lemma 2.2 and the inequality $N_t - N \geq N_t/(N+1)$ yield

$$
\begin{aligned}
\mathbb{E}_{\delta u}\big[\|\hat{\boldsymbol{u}}^B - \boldsymbol{u}^B\|\big] &\geq \mathbb{E}_{\delta u}\left[\left(\left(\sum_{i=1}^{N} + \sum_{i=N+1}^{N_t}\right)\left|\sum_{j=0}^{i-1}\lambda_m^j\right|^2\delta u_m^2\right)^{\frac{1}{2}}\right] \\
&\geq \mathbb{E}_{\delta u}\left[\left(\sum_{i=N+1}^{N_t}(1-\delta)\frac{C_{\lambda_m}^{2i}}{|1-\lambda_m|^2}\delta u_m^2\right)^{\frac{1}{2}}\right] \\
&\geq \alpha\mu\sqrt{N_t},
\end{aligned}
$$

where

$$
\alpha = \left(\frac{1-\epsilon}{|1-\lambda_m|^2(N+1)}\right)^{\frac{1}{2}}.
$$

2. If $|\lambda_m| = 1$ and $\lambda_m \neq 1$, we let $\lambda = e^{j\theta}$, where $j^2 = -1$ and $\theta \in (0, 2\pi)$. Taking into account the inequality $\sqrt{N_t - z} \geq \sqrt{1-z}\sqrt{N_t}$, $z \in [0,1)$, we obtain

$$
\begin{aligned}
\mathbb{E}_{\delta u}\big[\|\hat{\boldsymbol{u}}^B - \boldsymbol{u}^B\|\big] &\geq \mathbb{E}_{\delta u}\left[\left(\sum_{i=1}^{N_t}\left|\frac{1-e^{ji\theta}}{1-e^{j\theta}}\right|^2\delta u_m^2\right)^{\frac{1}{2}}\right] = \mathbb{E}_{\delta u}\left[\left(\sum_{i=1}^{N_t}\frac{1-\cos i\theta}{1-\cos\theta}\right)^{\frac{1}{2}}|\delta u_m|\right] \\
&= \mathbb{E}_{\delta u}\left[\frac{1}{(1-\cos\theta)^{\frac{1}{2}}}\left(N_t - \frac{\sin((N_t+1)\theta/2) - \sin(\theta/2)}{2\sin(\theta/2)}\right)^{\frac{1}{2}}|\delta u_m|\right] \\
&\geq \mathbb{E}_{\delta u}\left[\left(\frac{N_t - 1/\sin(\theta/2)}{1-\cos\theta}\right)^{\frac{1}{2}}|\delta u_m|\right] \\
&= \alpha\mu\sqrt{N_t},
\end{aligned}
$$

where

$$\alpha = 2\left(\frac{\sin(\theta/2)-1}{\sin(\theta/2)-\sin(3\theta/2)}\right)^{\frac{1}{2}}.$$

3. If $|\lambda_m| = 1$ and $\lambda_m = 1$, we use the equality

$$\sum_{i=1}^{N_t} i^2 = \frac{1}{6}N_t(N_t+1)(2N_t+1)$$

and obtain

$$\begin{aligned}\mathbb{E}_{\delta u}\left[\|\hat{\boldsymbol{u}}^B - \boldsymbol{u}^B\|\right] &\geq \mathbb{E}_{\delta u}\left[\left(\frac{1}{6}N_t(N_t+1)(2N_t+1)\delta u_m^2\right)^{\frac{1}{2}}\right] \\ &\geq \alpha\mu\sqrt{N_t}N_t,\end{aligned}$$

where $\alpha = 1/\sqrt{3}$.

4. If $|\lambda_m| > 1$, then Lemma 2.2, AM-GM inequality, and the estimate $N_t - N \geq N_t/(N+1)$ give

$$\begin{aligned}\mathbb{E}_{\delta u}\left[\|\hat{\boldsymbol{u}}^B - \boldsymbol{u}^B\|\right] &\geq \mathbb{E}_{\delta u}\left[\left(\left(\sum_{i=1}^{N} + \sum_{i=N+1}^{N_t}\right)\left|\sum_{j=0}^{i-1}\lambda_m^j\right|^2 \delta u_m^2\right)^{\frac{1}{2}}\right] \\ &\geq \mathbb{E}_{\delta u}\left[\left(\sum_{i=N+1}^{N_t}(1-\epsilon)\frac{C_{\lambda_m}^{2i}}{|1-\lambda_m|^2}\delta u_m^2\right)^{\frac{1}{2}}\right] \\ &\geq \frac{\sqrt{1-\epsilon}}{|1-\lambda_m|}\mathbb{E}_{\delta u}\left[\left((N_t-N)|\lambda_m|^{N_t-N+1}\delta u_m^2\right)^{\frac{1}{2}}\right] \\ &\geq \alpha\mu\sqrt{N_t}\beta(N_t),\end{aligned}$$

where

$$\alpha = \left(\frac{(1-\epsilon)|\lambda_m|^{1-N}}{|1-\lambda_m|^2(N+1)}\right)^{\frac{1}{2}}.$$

The proof is complete. □

If different random feature functions $\{\phi_{i,j}\}$ are chosen over different time subdomains, we observe that $\|\hat{\boldsymbol{u}}^B - \boldsymbol{u}^B\|$ also satisfies the estimation in Lemma 2.3.

Lemma 2.4. *For the space-time random feature method, there exists* $N \in \mathbb{N}^+$ *and* $\alpha > 0$ *such that*

$$\mathbb{E}_{\delta u}\left[\|\hat{\boldsymbol{u}}^S - \boldsymbol{u}^S\|\right] \leq \alpha\delta\sqrt{N_t}.$$

DOI https://doi.org/10.4208/eajam.2023-065.050423 | Generated on 2024-12-18 23:56:09

Proof. Using the Schwarz inequality, we write

$$\left(\mathbb{E}_{\delta u}\left[\|\hat{\boldsymbol{u}}^S-\boldsymbol{u}^S\|\right]\right)^2 \le \mathbb{E}_{\delta u}\left[\|\hat{\boldsymbol{u}}_i^S-\boldsymbol{u}_i^S\|^2\right] = \mathbb{E}_{\delta u}\left[\left\|\sum_{i=1}^{N_t}\sum_{k=1}^{M}\delta u_{k,i}\boldsymbol{b}_k\right\|^2\right] = \alpha\delta^2 N_t,$$

where $\alpha=\sqrt{M}$. ☐

Lemma 2.5. *Let two functions $u_M(\boldsymbol{x},t)$ and $\hat{u}_M(\boldsymbol{x},t)$ be of the form* (2.8) *with $N_x=1$ and the coefficients being $\boldsymbol{u}$, $\hat{\boldsymbol{u}}$, respectively, then there exists $C_1\ge C_2>0$ such that*

$$C_1\|\hat{\boldsymbol{u}}-\boldsymbol{u}\| \ge \|\hat{u}_M(\boldsymbol{x},t)-u_M(\boldsymbol{x},t)\|_{L^2} \ge C_2\|\hat{\boldsymbol{u}}-\boldsymbol{u}\|.$$

Proof. Notice that

$$\|\hat{u}_M(\boldsymbol{x},t)-u_M(\boldsymbol{x},t)\|_{L^2} = \left(\sum_{n=1}^{N_t}\sum_{i=1}^{J_n}(\hat{u}_{n,i}-u_{n,i})^2\int_{\Omega\times[0,T]}\phi_{n,i}(\boldsymbol{x},t)^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t\right)^{\frac{1}{2}} \le C_1\|\hat{\boldsymbol{u}}-\boldsymbol{u}\|,$$

where $C_1=\max_{n,i}\|\phi_{n,i}(\boldsymbol{x},t)\|_2$, and

$$\|\hat{u}_M(\boldsymbol{x},t)-u_M(\boldsymbol{x},t)\|_{L^2} = \left(\sum_{n=1}^{N_t}\sum_{i=1}^{J_n}(\hat{u}_{n,i}-u_{n,i})^2\int_{\Omega\times[0,T]}\phi_{n,i}(\boldsymbol{x},t)^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t\right)^{\frac{1}{2}} \ge C_2\|\hat{\boldsymbol{u}}-\boldsymbol{u}\|,$$

where $C_2=\min_{n,i}\|\phi_{n,i}(\boldsymbol{x},t)\|_2$. Moreover, for all $\phi_{n,i}\in\mathscr{L}^2(\Omega\times[0,T])$, we have $C_1,C_2<+\infty$. ☐

Theorem 2.3 (Lower bound)**.** *Denote u_e the exact solution. Let δu be the error between the numerical solution and the exact solution to the least-squares problem. Given $\epsilon>0$, we assume that*

$$\left\|u_M^B(\boldsymbol{x},t)-u_e(\boldsymbol{x},t)\right\|_{L^2(\Omega\times[t_n,t_{n+1}])} \le \epsilon, \quad \forall n=0,\dots,N.$$

Then there exists $N\in\mathbb{N}^+$ and $\alpha>0$ such that when $N_t>N$,

$$\mathbb{E}_{\delta u}\left[\left\|\hat{u}_M^B(\boldsymbol{x},t)-u_e(\boldsymbol{x},t)\right\|_{L^2}\right] \ge \alpha\mu\sqrt{N_t}\beta(N_t)-\epsilon\sqrt{N_t}.$$

Proof. Lemmas 2.3 and 2.5 give

$$\begin{aligned}&\mathbb{E}_{\delta u}\left[\left\|\hat{u}_M^B(\boldsymbol{x},t)-u_e(\boldsymbol{x},t)\right\|_{L^2}\right]\\ \ge\,&\mathbb{E}_{\delta u}\left[\left\|\hat{u}_M^B(\boldsymbol{x},t)-u_M^B(\boldsymbol{x},t)\right\|_{L^2}\right]-\left\|u_M^B(\boldsymbol{x},t)-u_e(\boldsymbol{x},t)\right\|_{L_2}\\ \ge\,&\alpha\mu\sqrt{N_t}\beta(N_t)-\left(\sum_{n=1}^{N_t}\epsilon^2\right)^{\frac{1}{2}} = \alpha\mu\sqrt{N_t}\beta(N_t)-\epsilon\sqrt{N_t}.\end{aligned}$$

The proof is complete. ☐

Theorem 2.4 (Upper bound)**.** *Denote* u_e *the exact solution. Let* δu *be the error between the numerical solution and the exact solution to the least-squares problem. Given* $\epsilon > 0$*, we assume that*

$$\left\| u_M^S(\boldsymbol{x},t) - u_e(\boldsymbol{x},t) \right\|_{L^2(\Omega\times[t_n,t_{n+1}])} \le \epsilon \quad \textit{for all} \quad n = 0,\dots,N.$$

Then there exists $N \in \mathbb{N}^+$ *and* $\alpha > 0$ *such that when* $N_t > N$*,*

$$\mathbb{E}_{\delta u}\left[\left\| \hat{u}^S(\boldsymbol{x},t) - u_e(\boldsymbol{x},t) \right\|_{L^2} \right] \le \alpha\delta\sqrt{N_t} + \epsilon\sqrt{N_t}.$$

Proof. It follows from Lemmas 2.4 and 2.5 that

$$\begin{aligned}
&\mathbb{E}_{\delta u}\left[\left\| \hat{u}^S(\boldsymbol{x},t) - u_e(\boldsymbol{x},t) \right\|_{L^2} \right] \\
\le\, &\mathbb{E}_{\delta u}\left[\left\| \hat{u}^S(\boldsymbol{x},t) - u^S(\boldsymbol{x},t) \right\|_{L^2} \right] + \left\| u^S(\boldsymbol{x},t) + u_e(\boldsymbol{x},t) \right\|_{L_2} \\
\le\, &\alpha\sigma\sqrt{N_t} + \left(\sum_{n=1}^{N_t} \epsilon^2 \right)^{\frac{1}{2}} = \alpha\sigma\sqrt{N_t} + \sqrt{N_t}\,\epsilon.
\end{aligned}$$

The proof is complete. ☐

From Theorems 2.3 and 2.4, we see that the error of solving the least-squares problem by the block time-marching strategy increases exponentially in time, while the error in the ST-RFM does not have this problem. These are also confirmed by the numerical results given below.

3. Numerical Results

In this section, we present numerical results for one-dimensional and two-dimensional problems with simple geometry and a two-dimensional problem with complex geometry to demonstrate the effectiveness of the ST-RFM and confirm theoretical results.

3.1. One-dimensional problems

3.1.1. Heat equation

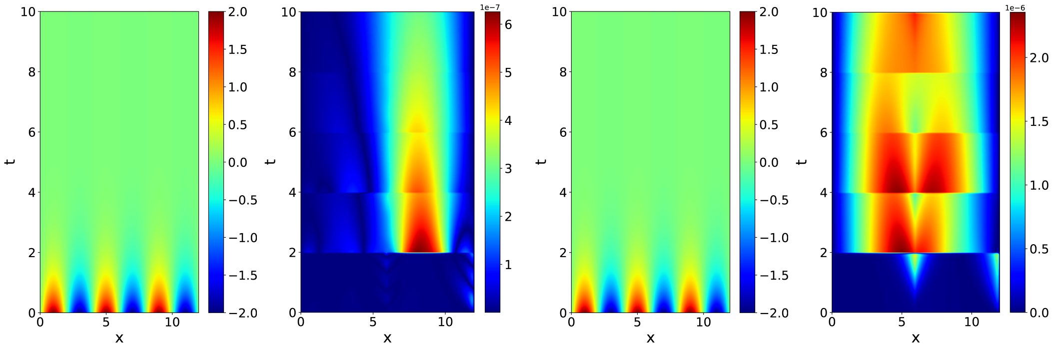
Consider the following problem:

$$\begin{aligned}
&\partial_t u(x,t) - \alpha^2 \partial_x^2 u(x,t) = 0, && x \in [x_0, x_1], && t \in [0,T], \\
&u(x_0,t) = g_1(t), && && t \in [0,T], \\
&u(x_1,t) = g_2(t), && && t \in [0,T], \\
&u(x,0) = h(x), && x \in [x_0, x_1],
\end{aligned} \tag{3.1}$$

where $\alpha = \pi/2$, $x_0 = 0$, $x_1 = 12$ and $T = 10$. The exact solution is chosen to be

$$u_e(x,t) = 2\sin(\alpha x)e^{-t}. \tag{3.2}$$

DOI https://doi.org/10.4208/eajam.2023-065.050423 | Generated on 2024-12-18 23:56:09



(a) STC (b) SoV

Figure 1: Results for the heat equation. (a) STC: Left, numerical solution; right, absolute error. (b) SoV: Left, numerical solution; right, absolute error.

We choose the initial condition $h(x)$ by restricting Eq. (3.2) to $t = 0$, and the boundary conditions $g_1(t)$ and $g_2(t)$ by restricting Eq. (3.2) to $x = x_0$ and $x = x_1$, respectively.

Set the default hyper-parameters $N_x = 2$, $N_t = 5$, $Q_x = 20$, $Q_t = 20$, $J_n = 400$ and $N_b = 1$. Numerical solutions and errors of STC (Eq. (2.6)) and SoV (Eq. (2.7)) are plotted in Fig. 1. The L^∞ error in the ST-RFM is small ($< 4.5e-6$), which indicates that both random feature functions have strong approximation properties.

Figs. 2(a)-2(d) show the convergence behavior with respect to different parameters. In Fig. 2(a), we set $N_b = 1$, $N_x = 2$, $J_n = 400$, $Q_x = Q_t = 20$ and $N_t = 1,\ldots,5$ to verify the convergence with respect to N_t. In Fig. 2(b), we set $N_x = 2$, $N_t = 1$, $J_n = 400$, $Q_x = Q_t = 20$ and $N_b = 1,\ldots,5$ to verify the convergence with respect to N_b. In Fig. 2(c), we set $N_b = 5$, $N_x = 2$, $N_t = 1$, $Q_x = Q_t = 20$ and $J_n = 50,100,200,300,400$ to verify the convergence with respect to J_n. In Fig. 2(d), we set $N_b = 5$, $N_x = 2$, $N_t = 1$, $J_n = 400$ and $Q_x = Q_t = 5,10,15,20,25$ to verify the convergence with respect to Q_x/Q_t. A clear trend of spectral accuracy is observed for the ST-RFM in both spatial and temporal directions.

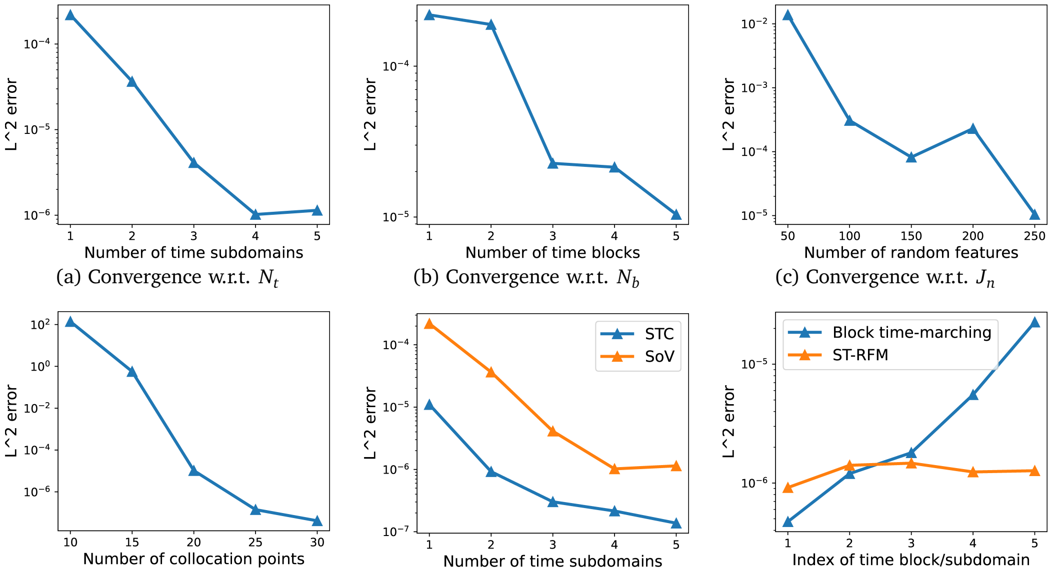
Now, we compare STC and SoV in Fig. 2(e), where the default hyper-parameter setting is used. For this example, STC performs better than SoV. The comparison between the block time-marching strategy and the ST-RFM is plotted in Fig. 2(f), where we set $N_b = 5$ and $N_t = 1$ for the block time-marching strategy and $N_b = 1$ and $N_t = 5$ for the ST-RFM. The L^2 error of solution by the block time-marching strategy increases exponentially fast with respect to the number of blocks, while the error in the ST-RFM remains almost flat over all time subdomains.

3.1.2. Heat equation with nonsmooth initial condition

Consider the heat equation (3.1) with $x_0 = 0$, $x_1 = 8$ and the nonsmooth initial condition as follows:

$$h(x) = 2\mathbb{I}_{0\le x<4}\sin(\pi x/2) + 2\mathbb{I}_{4\le x\le 8}\sin(\pi x).$$

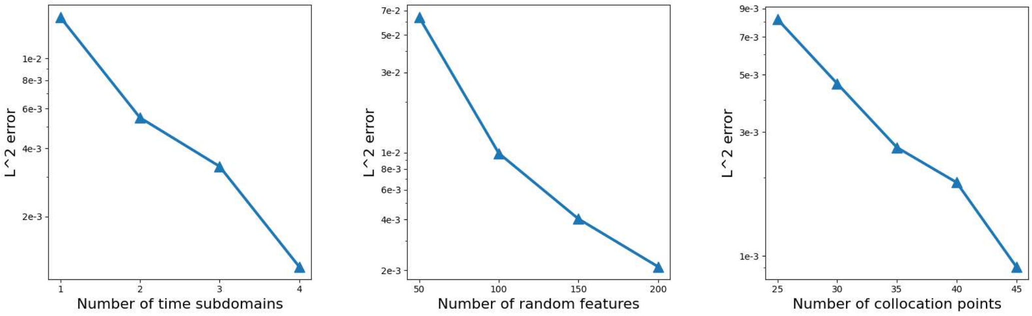
It is easy to check that $h(x)$ only belongs to $C([0,8])$.



(a) Convergence w.r.t. N_t (b) Convergence w.r.t. N_b (c) Convergence w.r.t. J_n

(d) Convergence w.r.t. Q (e) STC v.s. SoV (f) Block time-marching v.s. ST-RFM

Figure 2: Convergence behavior as different hyper-parameters are varied for the heat equation. (a) the number of time subdomains in the ST-RFM; (b) the number of time blocks in the block time-marching strategy; (c) the number of random features in each direction; (d) the number of collocation points in each direction; (e) comparison of STC and SoV; (f) comparison of the ST-RFM and the block time-marching strategy.



(a) Convergence w.r.t. time subdomains (b) Convergence w.r.t. J_n (c) Convergence w.r.t. Q

Figure 3: Convergence behavior as different hyper-parameters are varied for the heat equation with nonsmooth initial condition. (a) the number of time subdomains in the ST-RFM; (b) the number of random features in each direction; (c) the number of collocation points in the temporal direction.

Since an exact solution is not analytically available for the problem under consideration, we employ a set of hyper-parameters, namely $N_x = 2$, $N_t = 5$, $Q_x = 30$, $Q_t = 50$, $J_n = 250$, and $N_b = 1$, to obtain a reference solution. Numerical convergence of ST-RFM is evaluated in terms of the relative L^2 error, which is depicted in Fig. 3. We use STC to implement the basis function of ST-RFM here. Apparently, spectral accuracy is observed. However, the accuracy is only around 10^{-3}, two orders of magnitude larger than that in the smooth case (3.1), which is attributed to the regularity deficiency.

DOI https://doi.org/10.4208/eajam.2023-065.050423 | Generated on 2024-12-18 23:56:09

3.1.3. Wave equation

Consider the following problem:

$$
\begin{aligned}
&\partial_t^2 u(x,t) - \alpha^2 \partial_x^2 u(x,t) = 0, && x,t \in [x_0,x_1] \times [0,T],\\
&u(x_0,t) = u(x_1,t) = 0, && t \in [0,T],\\
&u(x,0) = g_1(x), && x \in [x_0,x_1],\\
&\partial_t u(x,0) = g_2(x), && x \in [x_0,x_1],
\end{aligned}
$$

where $x_0 = 0$, $x_1 = 6\pi$, $\alpha = 1$ and $T = 10$. The exact solution is chosen to be

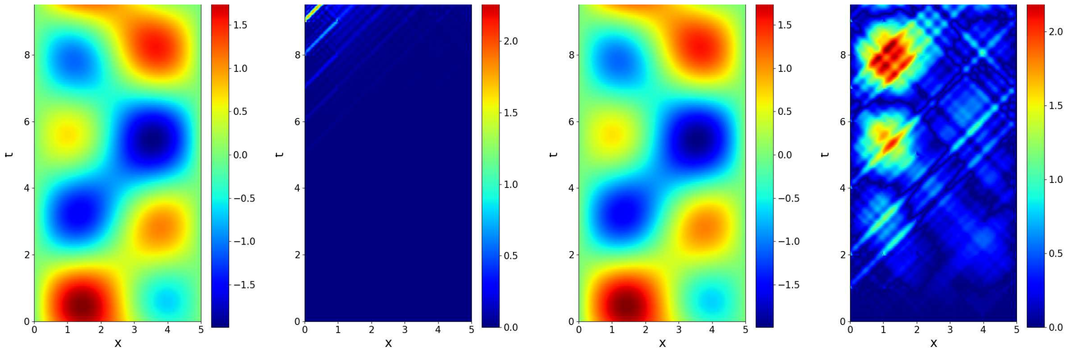
$$
u_e(x,t) = \cos\left(\frac{a\pi}{l}t\right)\sin\left(\frac{\pi}{l}x\right) + \left[\cos\left(\frac{2a\pi}{l}t\right) + \frac{l}{2a\pi}\sin\left(\frac{2a\pi}{l}t\right)\right]\sin\left(\frac{2\pi}{l}x\right),
$$

where $l = x_1 - x_0$. Initial conditions $g_1(x)$ and $g_2(x)$ are chosen accordingly.

Set the default hyper-parameters $N_x = 5$, $N_t = 5$, $Q_x = 30$, $Q_t = 30$, $J_n = 300$ and $N_b = 1$. Numerical solutions and errors of STC and SoV are plotted in Fig. 4. The L^∞ error is smaller than $2.5e-8$.

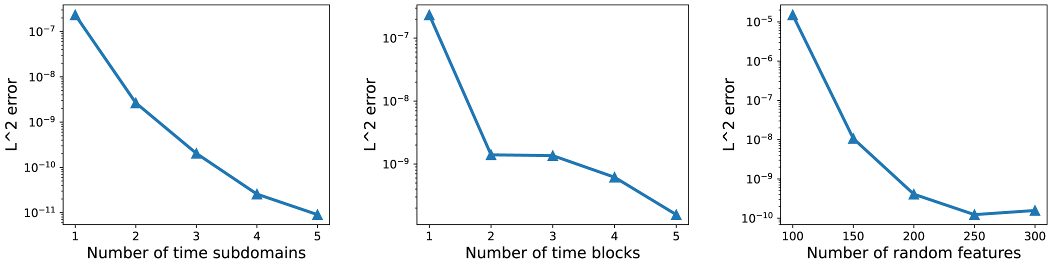
Figs. 5(a)-5(d) show the convergence behavior with respect to different parameters. In Fig. 5(a), we set $N_b = 1$, $N_x = 5$, $J_n = 300$, $Q_x = Q_t = 30$ and $N_t = 1,\ldots,5$ to verify the convergence with respect to N_t. In Fig. 5(b), we set $N_x = 5$, $N_t = 1$, $J_n = 300$, $Q_x = Q_t = 30$ and $N_b = 1,\ldots,5$ to verify the convergence with respect to N_b. In Fig. 5(c), we set $N_b = 5$, $N_x = 5$, $N_t = 1$, $Q_x = Q_t = 30$ and $J_n = 100, 150, 200, 250, 300$ to verify the convergence with respect to J_n. In Fig. 5(d), we set $N_b = 5$, $N_x = 5$, $N_t = 1$, $J_n = 300$ and $Q_x = Q_t = 10, 15, 20, 25, 30$ to verify the convergence with respect to Q_x/Q_t. A clear trend of spectral accuracy is observed for the ST-RFM in both spatial and temporal directions.

Now, we compare STC and SoV in Fig. 5(e), where we set $N_b = 1$, $N_x = 5$, $N_t = 1,\ldots,5$, $Q_x = Q_t = 30$ and $J_n = 300$. For this example, SoV performs better than STC. The comparison between the block time-marching strategy and the ST-RFM is plotted in Fig. 5(f), where we set $N_b = 5$ and $N_t = 1$ for block time-marching strategy and $N_b = 1$

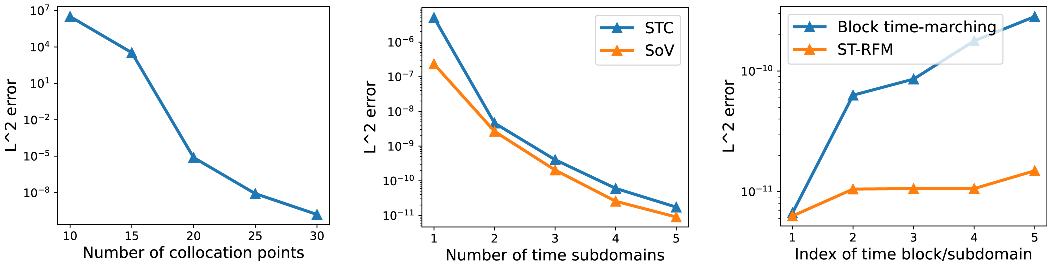


(a) STC (b) SoV

Figure 4: Results for the wave equation. (a) ST-RFM with STC: Left, numerical solution; right, absolute error. (b) ST-RFM with SoV: Left, numerical solution; right, absolute error.



(a) Convergence w.r.t. N_t (b) Convergence w.r.t. N_b (c) Convergence w.r.t. J_n



(d) Convergence w.r.t. Q (e) STC v.s. SoV (f) Block time-marching v.s. ST-RFM

Figure 5: Convergence behavior as different hyper-parameters are varied for the wave equation. (a) the number of time subdomains in the ST-RFM; (b) the number of time blocks in the block time-marching strategy; (c) the number of random features in each direction; (d) the number of collocation points in each direction; (e) comparison of STC and SoV; (f) comparison of the ST-RFM and the block time-marching strategy.

and $N_t = 5$ for ST-RFM. The L^2 error of the solution by the block time-marching strategy increases exponentially fast with respect to the number of blocks, while the error in the ST-RFM remains almost flat over all time subdomains.

3.1.4. Schrödinger equation

Consider the following problem:

$$\begin{aligned}
&i\partial_t\psi(x,t)+0.5\Delta\psi(x,t)=0, && x,t\in[x_0,x_1]\times[0,T],\\
&\psi(x,0)=g(x), && x\in[x_0,x_1],\\
&\psi(x_0,t)=\psi(x_1,t), && t\in[0,T],\\
&\partial_x\psi(x_0,t)=\partial_x\psi(x_1,t), && t\in[0,T],
\end{aligned}$$

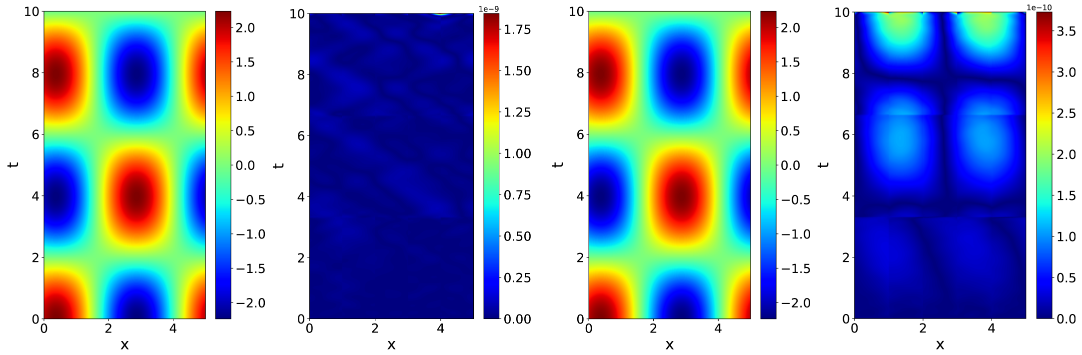
where $x_0=0$, $x_1=5$ and $T=10$. The exact solution is chosen to be

$$\psi(x,t)=e^{-i\omega^2t/2}\big(2\cos(\omega x)+\sin(\omega x)\big), \quad \omega=\frac{2\pi}{b_1-a_1},$$

and $g(x)$ is chosen accordingly.

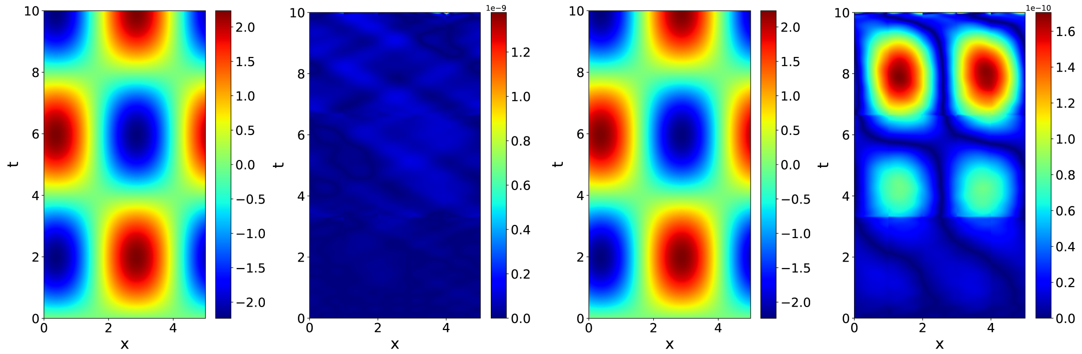
Set the default hyper-parameters $N_x = 5$, $N_t = 3$, $Q_x = 30$, $Q_t = 30$, $J_n = 300$ and $N_b = 1$. Numerical solutions and errors of STC and SoV are plotted in Fig. 6. The L^∞ error in the ST-RFM is smaller than $2.0e-9$.

DOI https://doi.org/10.4208/eajam.2023-065.050423 | Generated on 2024-12-18 23:56:09



(a) Real part: STC

(b) Real part: SoV



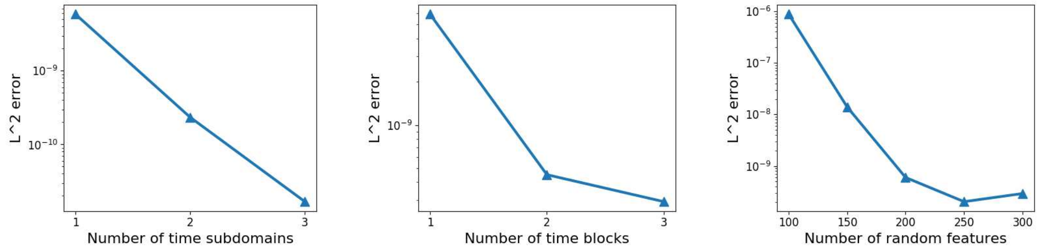
(c) Imaginary part: STC

(d) Imaginary part: SoV

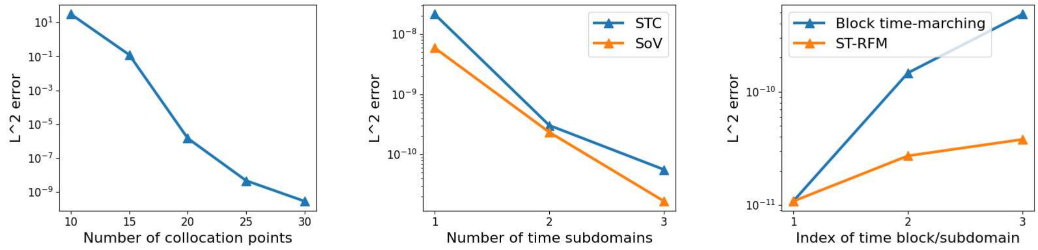
Figure 6: Results for Schrödinger equation. (a), (c) STC: Left, numerical solution; right, absolute error; (b), (d) SoV: Left, numerical solution; right, absolute error.

Figs. 7(a)-7(d) (real part) and Figs. 8(a)-8(d) (imaginary part) show the convergence behavior with respect to different parameters. In Fig. 7(a) and Fig. 8(a), we set $N_b = 1$, $N_x = 5$, $J_n = 300$, $Q_x = Q_t = 30$ and $N_t = 1,2,3$ to verify the convergence with respect to N_t. In Fig. 7(b) and Fig. 8(b), we set $N_x = 5$, $N_t = 1$, $J_n = 300$, $Q_x = Q_t = 30$ and $N_b = 1,2,3$ to verify the convergence with respect to N_b. In Fig. 7(c) and Fig. 8(c), we set $N_b = 3$, $N_x = 5$, $N_t = 1$, $Q_x = Q_t = 30$ and $J_n = 100,150,200,250,300$ to verify the convergence with respect to J_n. In Fig. 7(d) and Fig. 8(d), we set $N_b = 3$, $N_x = 5$, $N_t = 1$, $J_n = 300$ and $Q_x = Q_t = 10,15,20,25,30$ to verify the convergence with respect to Q_x/Q_t. A clear trend of spectral accuracy is observed for the ST-RFM in both spatial and temporal directions.

Now, we compare STC and SoV in Fig. 7(e) and Fig. 8(e), where we set $N_b = 1$, $N_x = 5$, $N_t = 1,2,3$, $Q_x = Q_t = 30$ and $J_n = 300$. For this example, SoV performs better than STC. The comparison between the block time-marching strategy and the ST-RFM is plotted in Fig. 7(f) and Fig. 8(f), where we set $N_b = 3$ and $N_t = 1$ for block time-marching strategy and $N_b = 1$ and $N_t = 3$ for ST-RFM. The L^2 error of the solution by the block time-marching strategy increases exponentially fast with respect to the number of blocks, while the error in the ST-RFM increases slower over all time subdomains.

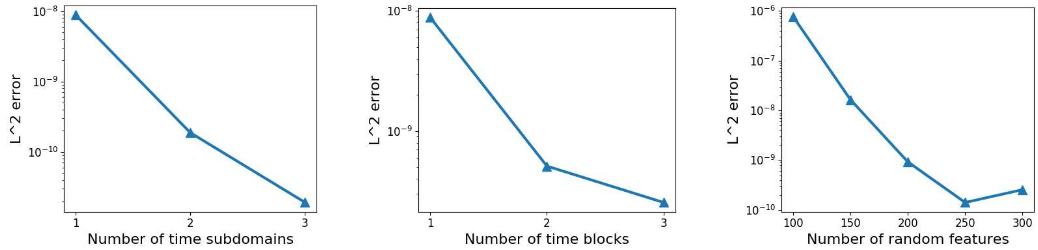


(a) Convergence w.r.t. N_t (b) Convergence w.r.t. N_b (c) Convergence w.r.t. J_n

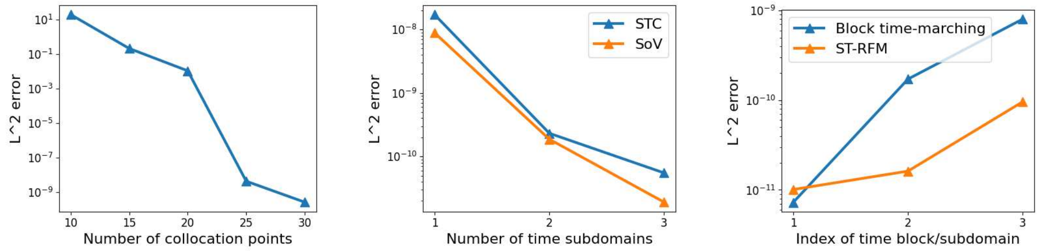


(d) Convergence w.r.t. Q (e) STC v.s. SoV (f) Block time-marching v.s. ST-RFM

Figure 7: Convergence behavior as different hyper-parameters are varied for Schrödinger equation (real part). (a) the number of time subdomains for ST-RFM; (b) the number of time blocks in the block time-marching strategy; (c) the number of random features in each direction; (d) the number of collocation points in each direction; (e) comparison of STC and SoV; (f) comparison of the ST-RFM and the block time-marching strategy.



(a) Convergence w.r.t. N_t (b) Convergence w.r.t. N_b (c) Convergence w.r.t. J_n



(d) Convergence w.r.t. Q (e) STC v.s. SoV (f) Block time-marching v.s. ST-RFM

Figure 8: Convergence behavior as different hyper-parameters are varied for Schrödinger equation (imaginary part). (a) the number of time subdomains for ST-RFM; (b) the number of time blocks in the block time-marching strategy; (c) the number of random features in each direction; (d) the number of collocation points in each direction; (e) comparison of STC and SoV; (f) comparison of the ST-RFM and the block time-marching strategy.

3.1.5. Verification of theoretical results

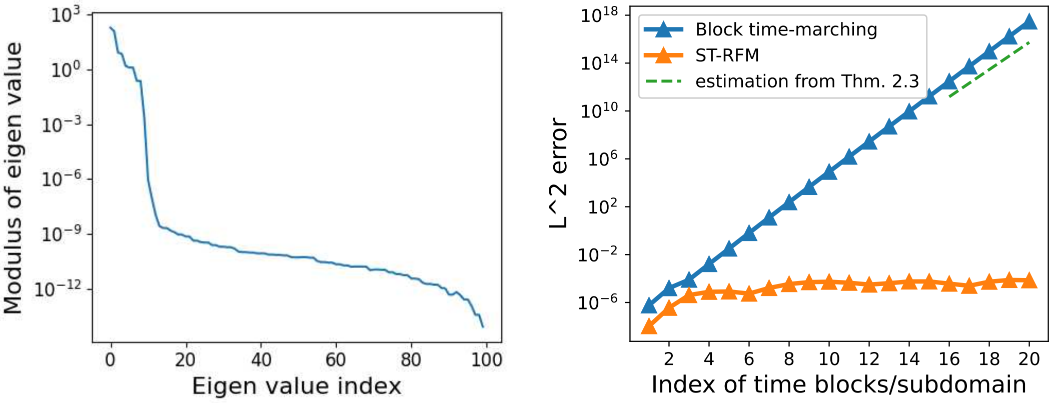
First, we verify Assumption 2.1 for all one-dimensional problems using the sufficient condition that the number of different eigenvalues of $\mathbf{B}$, denoted by #unique λ_k, equals to the number of random features J_n. Results are recorded in Table 1. The number of unique eigenvalues of $\mathbf{B}$ equals to J_n for all one-dimensional problems and Assumption 2.1 is verified.

Table 1: Eigenvalue distribution of $\mathbf{B}$ in one-dimensional problems for the verification of Assumption 2.1.

Heat		Wave		Schrödinger	
J_n	#unique λ_k	J_n	#unique λ_k	J_n	#unique λ_k
100	100	100	100	200	200
150	150	150	150	300	300
200	200	200	200	400	400
250	250	250	250	500	500
300	300	300	300	600	600
350	350	350	350	700	700
400	400	400	400	800	800

Next, we use the wave equation to verify the error estimate in Theorem 2.3. For the block time-marching strategy, we set $N_b = 20, N_x = 2, N_t = 1, J_n = 100, Q_x = 10, Q_t = 10$, and for ST-RFM, we set $N_b = 1, N_x = 2, N_t = 20, J_n = 100, Q_x = 10, Q_t = 10$. Eigenvalues of $\mathbf{B}$ are plotted in Fig. 9(a), and the L^2 error is plotted in Fig. 9(b).

The largest modulus of eigenvalues of $\mathbf{B}$ from Fig. 9(a) is 187.75, indicating that $\beta(x) \approx 13.70^x$, while the L^2 error in the block time-marching strategy increases exponentially with



(a) Modulus of λ_k

(b) L^2 error

Figure 9: (a) Eigenvalue distribution for the matrix $\mathbf{B}$; (b) L^2 error for the wave equation. We see a good agreement between Theorems 2.4-2.5, and the actual numerical results.

DOI https://doi.org/10.4208/eajam.2023-065.050423 | Generated on 2024-12-18 23:56:09

the rate 17.70 from Fig. 9(b). Therefore, the lower bound estimate in Theorem 2.4 is verified. In addition, the L^2 error in the ST-RFM remains almost flat, which is mainly controlled by the approximation power of space-time random features.

3.2. Two-dimensional problems

3.2.1. Membrane vibration over a simple geometry

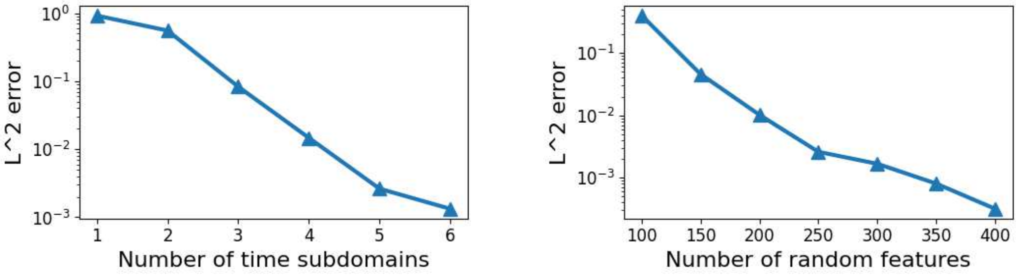
Consider the following problem:

$$\begin{aligned}
&\partial_{tt}u(x,y,t)-\alpha^2\Delta u(x,y,t)=0, && (x,y),t\in\Omega\times[0,T],\\
&u(x,y,0)=\phi(x,y), && (x,y)\in\Omega,\\
&\partial_t u(x,y,0)=\psi(x,y), && (x,y)\in\Omega,\\
&u(x,y,t)=0, && (x,y)\in\partial\Omega\times[0,T],
\end{aligned} \tag{3.3}$$

where $\Omega=[0,5]\times[0,4]$, $\alpha=1$ and $T=10$. The exact solution is chosen to be

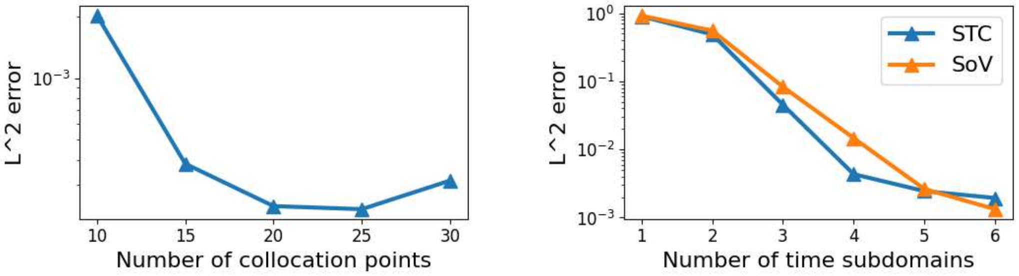
$$u_e(x,y,t)=\sin(\mu x)\sin(\nu y)\big(2\cos(\lambda t)+\sin(\lambda t)\big),$$

where $\mu=2\pi/(x_1-x_0)$, $\nu=2\pi/(y_1-y_0)$, $\lambda=\sqrt{\mu^2+\nu^2}$. The initial conditions $\phi(x,y)$ and $\psi(x,y)$ are chosen accordingly.



(a) Convergence w.r.t. N_t

(b) Convergence w.r.t. J_n



(c) Convergence w.r.t. Q

(d) STC v.s. SoV

Figure 10: Convergence behavior as different hyper-parameters are varied for the membrane vibration equation (3.3).

Set the default hyper-parameters $N_x = N_y = 2$, $N_t = 5$, $Q_x = Q_y = Q_t = 30$, $J_n = 400$ and $N_b = 1$. We report the convergence behavior with respect to different parameters in Figs. 10(a)-10(c). In Fig. 10(a), we set $N_b = 1$, $N_x = N_y = 2$, $J_n = 400$, $Q_x = Q_y = Q_t = 30$ and $N_t = 1,\dots,6$ to verify the convergence with respect to N_t. In Fig. 10(b), we set $N_b = 1$, $N_x = N_y = N_t = 2$, $Q_x = Q_y = Q_t = 30$ and $J_n = 100, 150, 200, 250, 300, 350, 400$ to verify the convergence with respect to J_n. In Fig. 10(c), we set $N_b = 5$, $N_x = N_y = N_t = 5$, $J_n = 400$ and $Q_x = Q_y = Q_t = 10, 15, 20, 25, 30$ to verify the convergence with respect to the number of collocation points. A clear trend of spectral accuracy is observed for the ST-RFM in both spatial and temporal directions. Now, we compare STC and SoV in Fig. 10(d), when $N_b = 1$, $N_x = N_y = 2$, $N_t = 1,\dots,6$, $Q_x = Q_y = Q_t = 30$ and $J_n = 400$. Performances of STC and SoV are close.

3.2.2. Membrane vibration over a complex geometry

Consider a complex geometry Ω in Fig. 11 and the following the membrane vibration problem:

$$\begin{array}{ll} \partial_{tt}u(x,y,t) - \alpha^2 \Delta u(x,y,t) = 1, & (x,y),t \in \Omega \times [0,T], \\ u(x,y,0) = \phi(x,y), & (x,y) \in \Omega, \\ \partial_t u(x,y,0) = \psi(x,y), & (x,y) \in \Omega, \\ u(x,y,t) = 0, & (x,y) \in \partial\Omega \times [0,T], \end{array}$$

where $T = 10.0$, $\alpha = 1$. The same $\phi(x,y)$ and $\psi(x,y)$ are used as in Section 3.2.1. Set the default hyper-parameters $N_x = 5$, $N_t = 1$, $Q_x = 30$, $Q_t = 30$, $J_n = 400$ and $N_b = 5$. Numerical solutions of SoV at different times are plotted in Fig. 12.

No exact solution is available and numerical convergence is shown here. L^2 errors of solution $\|u_M - u_{ref}\|$ with respect to different parameters is recorded in Tables 2-4. The

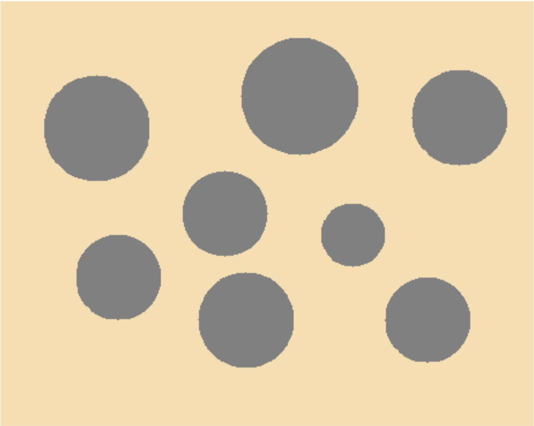
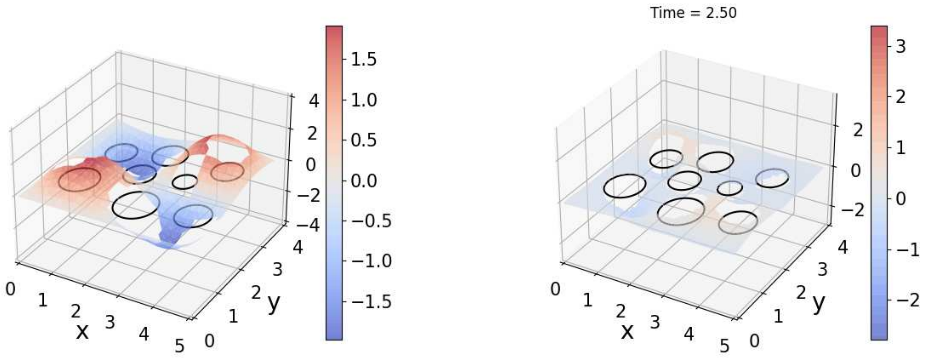
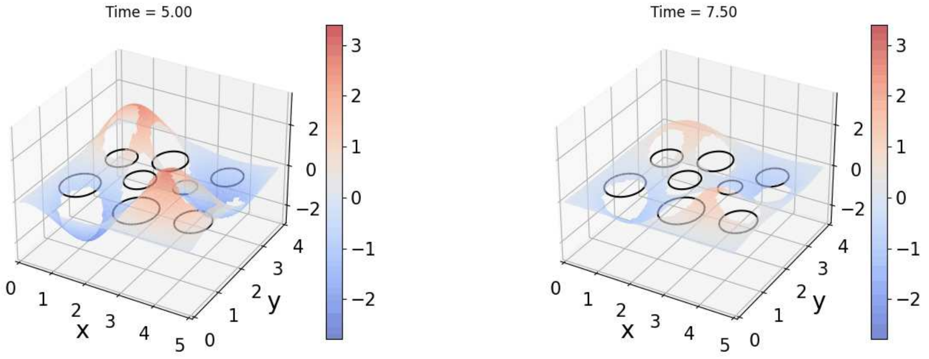


Figure 11: The complex geometry for the membrane vibration problem.

DOI https://doi.org/10.4208/eajam.2023-065.050423 | Generated on 2024-12-18 23:56:09



(a) Time = 0.0 (b) Time = 2.5



(c) Time = 5.0 (d) Time = 7.5

Figure 12: Numerical solution of the membrane vibration problem over a complex geometry at different times.

solution with the largest degrees of freedom is chosen as the reference solution. As the parameter varies, the numerical solution converges to the reference solution, indicating the robustness of the ST-RFM in solving time-dependent partial differential equations over a complex geometry.

Table 2: L^2 error with respect to the number of time sub-domains.

# Time Blocks	N_x/N_y	N_t	J_n	$Q_x/Q_y/Q_t$	Solution error
1	2	1	400	30	5.98e-1
1	2	2	400	30	7.07e-2
1	2	3	400	30	4.57e-2
1	2	4	400	30	3.89e-2
1	2	5	400	30	3.31e-2
1	2	6	400	30	Reference

Table 3: L^2 error with respect to the number of collocation points.

# Time Blocks	N_x/N_y	N_t	J_n	$Q_x/Q_y/Q_t$	Solution error
5	2	2	400	10	7.71e-2
5	2	2	400	15	2.16e-2
5	2	2	400	20	1.44e-2
5	2	2	400	25	1.25e-2
5	2	2	400	30	Reference

Table 4: L^2 error with respect to the number of random features.

# Time Blocks	N_x/N_y	N_t	J_n	$Q_x/Q_y/Q_t$	Solution error
5	2	2	100	30	1.05e-1
5	2	2	150	30	5.25e-2
5	2	2	200	30	3.96e-2
5	2	2	250	30	2.82e-2
5	2	2	300	30	1.80e-2
5	2	2	350	30	1.46e-2
5	2	2	400	30	Reference

4. Concluding Remarks

In this work, we study numerical algorithms for solving time-dependent partial differential equations in the framework of the random feature method. Two types of random feature functions are considered: space-time concatenation random feature functions (STC) and space-time separation-of-variables random feature functions (SoV). A space-time partition of unity is used to piece together local random feature functions to approximate the solution. We tested these ideas for a number of time-dependent problems. Our numerical results show that ST-RFM with both STC and SoV has spectral accuracy in space and time. The error in ST-RFM remains almost flat as the number of time subdomains increases, while the error grows exponentially fast when the block time-marching strategy is used. Consistent theoretical error estimates are also proved. A two-dimensional problem over a complex geometry is used to show that the method is insensitive to the complexity of the underlying domain.

Acknowledgments

The work is supported by the National Key R&D Program of China (2022YFA1005200, 2022YFA1005203), by the NSFC Major Research Plan - Interpretable and General-purpose Next-generation Artificial Intelligence (92270001, 92270205), by the Anhui Center for Applied Mathematics, and by the Major Project of Science & Technology of Anhui Province (202203a05020050).

References

[1] F. Calabrò, G. Fabiani and C. Siettos, *Extreme learning machine collocation for the numerical solution of elliptic pdes with sharp gradients*, Comput. Methods Appl. Mech. Engrg. **387**, 114188 (2021).

[2] J.R. Chen, X.R. Chi, W.N. E and Z.W. Yang, *Bridging traditional and machine learning-based algorithms for solving pdes: The random feature method*, J. Mach. Learn. **1**, 268–298 (2022).

[3] G. Cybenko, *Approximation by superpositions of a sigmoidal function*, Math. Control Signals Systems **2**, 303–314 (1989).

[4] S.C. Dong and Z.W. Li, *Local extreme learning machines and domain decomposition for solving linear and nonlinear partial differential equations*, Comput. Methods Appl. Mech. Engrg. **387**, 114129 (2021).

[5] W.N. E and Y. Bing, *The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems*, Commun. Math. Stat. **6**, 1–12 (2018).

[6] W.N. E, J.Q. Han and A. Jentzen, *Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations*, Commun. Math. Stat. **5**, 349–380 (2017).

[7] W.N. E, J.Q. Han and A. Jentzen, *Algorithms for solving high dimensional PDEs: From nonlinear Monte Carlo to machine learning*, Nonlinearity **35**, 278 (2021).

[8] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press (2016).

[9] J.Q. Han, A. Jentzen and W.N. E, *Solving high-dimensional partial differential equations using deep learning*, Proc. Natl. Acad. Sci. USA **115**, 8505–8510 (2018).

[10] G.B. Huang, Q.Y. Zhu and C.K. Siew, *Extreme learning machine: Theory and applications*, Neurocomputing **70**, 489–501 (2006).

[11] R.J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*, SIAM (2007).

[12] R.M. Neal, *Bayesian Learning for Neural Networks*, Springer-Verlag (1995).

[13] A. Rahimi and B. Recht, *Random features for large-scale kernel machines*, in: *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pp. 1177–1184, Curran Associates, Inc. (2007).

[14] M. Raissi, P. Perdikaris and G.E. Karniadakis, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, J. Comput. Phys. **378**, 686–707 (2019).

[15] J. Shen, T. Tang and L.L. Wang, *Spectral Methods: Algorithms, Analysis and Applications*, Springer Series in Computational Mathematics **41**, Springer-Verlag (2011).

[16] J. Sirignano and K. Spiliopoulos, *Dgm: A deep learning algorithm for solving partial differential equations*, J. Comput. Phys. **375**, 1339–1364 (2018).

[17] V. Thomée, *Galerkin Finite Element Methods for Parabolic Problems*, Springer Series in Computational Mathematics **25**, Springer-Verlag (2007).

[18] Y.L. Yang, M.Z. Hou and J.S. Luo, *A novel improved extreme learning machine algorithm in solving ordinary differential equations by legendre neural network methods*, Adv. Differ. Equ. **2018**, 1–24 (2018).

[19] Y.H. Zang, G. Bao, X.J. Ye and H.M. Zhou, *Weak adversarial networks for high-dimensional partial differential equations*, J. Comput. Phys. **411**, 109409 (2020).