HHS Public Access

Author manuscript

J Comput Phys. Author manuscript; available in PMC 2016 July 01.

Published in final edited form as:

J Comput Phys. 2015 July 1; 292: 43–55. doi:10.1016/j.jcp.2015.03.033.

Semi-implicit Integration Factor Methods on Sparse Grids for High-Dimensional Systems

Dongyong Wang[a,*], **Weitao Chen**[a,*], and **Qing Nie**[a,**]

[a]Department of Mathematics, University of California, Irvine, CA 92697, USA

Abstract

Numerical methods for partial differential equations in high-dimensional spaces are often limited by the curse of dimensionality. Though the sparse grid technique, based on a one-dimensional hierarchical basis through tensor products, is popular for handling challenges such as those associated with spatial discretization, the stability conditions on time step size due to temporal discretization, such as those associated with high-order derivatives in space and stiff reactions, remain. Here, we incorporate the sparse grids with the implicit integration factor method (IIF) that is advantageous in terms of stability conditions for systems containing stiff reactions and diffusions. We combine IIF, in which the reaction is treated implicitly and the diffusion is treated explicitly and exactly, with various sparse grid techniques based on the finite element and finite difference methods and a multi-level combination approach. The overall method is found to be efficient in terms of both storage and computational time for solving a wide range of PDEs in high dimensions. In particular, the IIF with the sparse grid combination technique is flexible and effective in solving systems that may include cross-derivatives and non-constant diffusion coefficients. Extensive numerical simulations in both linear and nonlinear systems in high dimensions, along with applications of diffusive logistic equations and Fokker-Planck equations, demonstrate the accuracy, efficiency, and robustness of the new methods, indicating potential broad applications of the sparse grid-based integration factor method.

Keywords

reaction-diffusion equations; implicit method; sparse grids; high-dimension; Fokker-Planck equation; stiffness

1. Introduction

Consider the following partial differential equation:

**Corresponding author: qnie@uci.edu (Qing Nie).
*Equal contribution.

$$\frac{\partial u(\mathbf{x},t)}{\partial t} = Pu(\mathbf{x},t) + f(u(\mathbf{x},t)), \mathbf{x} = (x_1, \ldots, x_d) \in (0,1)^d, t \in [0,T], \quad (1)$$

where P is an elliptic differential operator with respect to the spatial variable $\mathbf{x}$. This equation has been extensively studied because of its wide application in certain fields. For instance, the formation of the morphogen gradient during the development of the embryo is modeled using reaction-diffusion systems [1], where P denotes the Laplacian operator with respect to $\mathbf{x}$. The stochastic behavior of a gene network can be described using the Fokker-Planck equation [2], also known as the backward Kolmogorov equation, where P is a second-order differential operator containing cross-derivatives. In finance, the Black-Scholes equation adopts a similar form when used to estimate the price of options under several risk factors [3]. In population genetics, the site-frequency spectrum can be modeled using such equations as well [4].

From the numerical perspective, solving Eq. (1) in high dimensions can be extremely challenging. Due to the "curse of dimensionality", achieving good accuracy of $O(1/N_{\mathbf{x}}^{\alpha})$ (for example, $\alpha = 2$ by a second-order central difference formula) usually requires an $O(N_{\mathbf{x}}^{d})$ number of points in uniform grids. The storage and operations on this large number of grid points can be prohibitively expensive when d is large. In addition, spatial discretization due to high-order spatial derivatives and stiff reactions leads to severe stability constraints on the time step for temporal integration. For the explicit methods, such as Runge-Kutta or Euler methods, a severe constraint is placed on the time step, whereas for implicit methods, such as the Crank-Nicolson method, large nonlinear systems are solved at each time step, leading to excessive computational costs.

The sparse grid technique has been shown to be an efficient approach for dealing with high-order spatial dimensions [5]. The discretization for this approach involves an $O(N \cdot (\log N)^{d-1})$ number of points along with an accuracy of $O((\log N)^{d-1}/N^2)$ when the one-dimensional piecewise linear hierarchical basis is applied. Such an approach may be extended to the general piecewise d-linear hierarchal basis by a tensor product construction in d-dimensional spaces [6, 5]. Other hierarchical bases, e.g., high-order polynomials, interpolets, wavelets, have also been developed for a higher order of accuracy [7, 8, 9]. Moreover, sparse grids have been recently applied to stochastic simulations and parameter estimations [10, 11, 12].

A popular approach for addressing the problem associated with dimensionality is the sparse grid finite element method using the Galerkin technique [13, 14] constructed on piecewise linear element using weak formulas. Another approach is the sparse grid finite difference method [15], which employs the regular second-order central difference approximation on the diffusion terms. The finite volume method [16] and the spectral method [17, 18] can also be integrated with the sparse grid technique. Another interesting approach is the so-called *sparse grid combination technique* [19] that uses multi-level regular uniform grids such that the final solution is constructed using a linear combination of the intermediate solutions at the uniform grids, leading to a straightforward implementation, similar to the standard uniform grid approach.

For temporal integration, the integration factor (IF) and exponential time differencing (ETD) methods are effective ways to deal with the temporal stability constraints arising from high-order spatial derivatives on uniform meshes [20, 21, 22]. The IF and ETD methods usually treat linear operators of the highest-order derivatives exactly, and hence, they provide good temporal stability by allowing larger sizes of time step in temporal updates [23, 24, 20]. For addressing the stiffness in reactions, a class of semi-implicit integration factor (IIF) methods, which integrate the differential operators exactly like the IF schemes while treating the reaction terms implicitly, have been developed [25]. In IIF, the calculation of the diffusion and implicit treatment of the reaction is decoupled such that the size of the nonlinear system that needs to be solved at each time step is the same as that of the original continuous PDEs. This property results in good efficiency, in addition to excellent stability conditions (e.g., the second-order IIF is linearly unconditionally stable). Moreover, the IIF method can handle reaction-convection-diffusion equations through an operator splitting technique [26] and can be incorporated with the adaptive meshes and general curvilinear coordinates [27]. Because the exact treatment of the diffusion terms requires computing exponentials of matrices resulting from the discretization of the linear differential operators, a compact representation of IIF (cIIF) [28, 27] and an array representation of IIF (AcIIF) [29] for systems with high spatial dimensions have been introduced. In the compact or array representations, the discretized functions in high spatial dimensions can be represented using multi-dimensional arrays rather than vectors or matrices so that the cost and storage associated with the calculation of the exponentials are significantly reduced.

In this paper, we integrate the sparse grids with the IIF methods to take the advantages provided by both methods to solve temporal PDEs (e.g., Eq. (1)) in high spatial dimensions. In particular, we combine the two temporal schemes, the IIF and AcIIF methods, with the three different sparse grid discretization techniques: finite element, finite difference, and sparse grid combination technique. The combination technique is found to be especially effective in terms of incorporating various implicit integration factor methods, particularly when dealing with systems that include cross-derivatives and non-constant diffusion coefficients.

The paper is organized as follows. In Section 2, we construct the IIF method on sparse grids based on the finite element and Galerkin technique. In Section 3, we construct the AcIIF method on sparse grids using the finite difference approximation. In Section 4, we apply the AcIIF method to the sparse grid combination technique. In Section 5, we describe numerical tests to demonstrate the accuracy, efficiency, and applications of these methods.

2. Semi-implicit integration factor method with finite element method on sparse grids

In this section, we consider the following reaction-diffusion equation:

$$\frac{\partial u}{\partial t}(\mathbf{x},t)=\Delta_{\mathbf{x}}u(\mathbf{x},t)+f(u), \quad (2)$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_d)$ and $\Delta_{\mathbf{x}}$ is the Laplacian operator with respect to $\mathbf{x}$. For simplicity of presentation, we assume the spatial domain to be $(0, 1)^d$ and zero Dirichlet boundary conditions at the boundaries for u. We only focus on the piecewise d-linear hierarchical basis on the sparse grids.

2.1. Weak formula under sparse grid finite element method

Let $N_{\mathbf{x}} = 2^K$, $\mathscr{S}_K = \{(k_1, \ldots, k_d) : k_1 + \ldots + k_d \le K, k_i \ge 0 \text{ for } i = 1, \ldots, d\}$, $\mathscr{I}_{\mathbf{k}} = \{(j_1, \ldots, j_d) : 1 \le j_i \le 2^{k_i} \text{ for } i = 1, \ldots, d\}$ for $\mathbf{k} \in \mathscr{S}_K$. The unknown u can be approximated using sparse grid, by

$$u(\mathbf{x}, t) \approx \sum_{\mathbf{k} \in \mathscr{S}_K} \sum_{\mathbf{j} \in \mathscr{I}_{\mathbf{k}}} v_{\mathbf{k},\mathbf{j}}(t) \phi_{\mathbf{k},\mathbf{j}}(\mathbf{x}), \quad (3)$$

where $\mathbf{k} = (k_1, \ldots, k_d)$ and $\mathbf{j} = (j_1, \ldots, j_d)$. Define the inner product of two functions $\psi_1(\mathbf{x})$ and $\psi_2(\mathbf{x})$:

$$<\psi_1(\mathbf{x}), \psi_2(\mathbf{x})> := \int_{(0,1)^d} \psi_1(\mathbf{x}) \psi_2(\mathbf{x}) d\mathbf{x}. \quad (4)$$

Substitute u in Eq. (2) by Eq. (3), and apply the inner product on both sides using $\varphi_{\bar{\mathbf{k}},\bar{\mathbf{j}}}(\mathbf{x})$, for all possible $\bar{\mathbf{k}}$ and $\bar{\mathbf{j}}$. This leads to the following weak formula on the sparse grids:

$$\begin{aligned} &\sum_{\mathbf{k} \in \mathscr{S}_K} \sum_{\mathbf{j} \in \mathscr{I}_{\mathbf{k}}} <\phi_{\mathbf{k},\mathbf{j}}, \phi_{\bar{\mathbf{k}},\bar{\mathbf{j}}}> \frac{dv_{\mathbf{k},\mathbf{j}}(t)}{dt} \\ &= \sum_{\mathbf{k} \in \mathscr{S}_K} \sum_{\mathbf{j} \in \mathscr{I}_{\mathbf{k}}} <\Delta_{\mathbf{x}} \phi_{\mathbf{k},\mathbf{j}}, \phi_{\bar{\mathbf{k}},\bar{\mathbf{j}}}> v_{\mathbf{k},\mathbf{j}}(t) \\ &+ \sum_{\mathbf{k} \in \mathscr{S}_K} \sum_{\mathbf{j} \in \mathscr{I}_{\mathbf{k}}} <\phi_{\mathbf{k},\mathbf{j}}, \phi_{\bar{\mathbf{k}},\bar{\mathbf{j}}}> f_{\mathbf{k},\mathbf{j}}(t) \end{aligned} \quad (5)$$

where $f_{\mathbf{k},\mathbf{j}}(t)$ is the hierarchical value of function $f(u(\mathbf{x}, t))$:

$$f(u(\mathbf{x}, t)) \approx \sum_{\mathbf{k} \in \mathscr{S}_K} \sum_{\mathbf{j} \in \mathscr{I}_{\mathbf{k}}} f_{\mathbf{k},\mathbf{j}}(t) \phi_{\mathbf{k},\mathbf{j}}(\mathbf{x}). \quad (6)$$

Note that both $< \varphi_{\mathbf{k},\mathbf{j}}, \varphi_{\bar{\mathbf{k}},\bar{\mathbf{j}}} >$ and $< \Delta_{\mathbf{x}} \varphi_{\mathbf{k},\mathbf{j}}, \varphi_{\bar{\mathbf{k}},\bar{\mathbf{j}}} >$ can be exactly calculated in advance because the piecewise d-linear functions are used. We define the following two matrices:

$$M = (<\phi_{\mathbf{k},\mathbf{j}}, \phi_{\bar{\mathbf{k}},\bar{\mathbf{j}}}>), \quad D = (<\Delta_{\mathbf{x}} \phi_{\mathbf{k},\mathbf{j}}, \phi_{\bar{\mathbf{k}},\bar{\mathbf{j}}}>), \quad (7)$$

and let the vectors $V(t) = (v_{\mathbf{k},\mathbf{j}}(t))$ and $F(t) = (f_{\mathbf{k},\mathbf{j}}(t))$ denote the hierarchical values of the unknown function and the reaction, respectively. Then, Eq. (5) becomes the following time-dependent problem:

$$M \frac{dV}{dt} = D \cdot V + M \cdot F, \text{ or } \frac{dV}{dt} = (M^{-1} D) V + F. \quad (8)$$

Instead of solving u, we only need to update the coefficient vector V at each time step according to Eq. (8). By directly applying the IIF method to Eq. (8), we obtain the following second-order (in time) IIF method based on the sparse grid finite element scheme (**IIF2-SG-FEM**):

$$V(t_{n+1})=e^{M^{-1}D\Delta t}\left(V(t_n)+\frac{\Delta t}{2}F(t_n)\right)+\frac{\Delta t}{2}F(t_{n+1}). \quad (9)$$

Note that because the nodal value is usually not equal to the hierarchical value, the above nonlinear system arising from the implicit treatment of the reaction term on the right-hand side cannot be solved locally at each grid point. However, sparse grid discretization allows us to re-order the grid points such that only a localized nonlinear system at each point needs to be solved. The procedure is introduced below.

2.2. Solving the nonlinear system Eq. (9)

To solve Eq. (9), we first calculate the hierarchical value of **k** = (0, …, 0) and **j** = (1, …, 1). Note that at this point, the nodal value equals the hierarchical value, i.e.,

$$v_{(0,\ldots,0),(1,\ldots,1)}(t_{n+1})=u(\frac{1}{2},\ldots,\frac{1}{2},t_{n+1}), \quad (10)$$

so that we can directly solve the following equation:

$$u(\frac{1}{2},\ldots,\frac{1}{2},t_{n+1})=L_{(0,\ldots,0),(1,\ldots,1)}+\frac{\Delta t}{2}f(u(\frac{1}{2},\ldots,\frac{1}{2},t_{n+1})), \quad (11)$$

where $L_{\mathbf{k},\mathbf{j}}$ is the computed value of $e^{M^{-1}D\Delta t}(V(t_n)+\frac{\Delta t}{2}F(t_n))$ at the corresponding grid point. Then, we solve the hierarchical values for **k** such that $k_1 + k_2 + \ldots + k_d = 1$. Notice that there are two ways to compute the nodal value of f at these points: (i) from the hierarchical value of f (left side of Eq. (12)) and (ii) from the nodal value of u (right side of Eq. (12)).

$$f_{\mathbf{k},\mathbf{j}}(t_{n+1})+0.5f_{(0,\ldots,0),(1,\ldots,1)}(t_{n+1})=f(v_{\mathbf{k},\mathbf{j}}(t_{n+1})+0.5v_{(0,\ldots0),(1,\ldots,1)}(t_{n+1})) \text{ for } k_1+\ldots+k_d=1. \quad (12)$$

Because $v_{(0,\ldots,0),(1,\ldots,1)}(t_{n+1})$ and $f_{(0,\ldots,0),(1,\ldots,1)}(t_{n+1})$ are known, substituting (12) into Eq. (9) for $F(t_{n+1})$, we obtain

$$v_{\mathbf{k},\mathbf{j}}(t_{n+1})=L_{\mathbf{k},\mathbf{j}}+\frac{\Delta t}{2}(f(v_{\mathbf{k},\mathbf{j}}(t_{n+1})+0.5v_{(0,\ldots,0),(1,\ldots,1)}(t_{n+1}))-0.5f_{(0,\ldots,0),(1,\ldots,1)}(t_{n+1})). \quad (13)$$

As a result, we can solve the localized nonlinear system (Eq. (13)) to obtain all hierarchical values $v_{\mathbf{k},\mathbf{j}}(t_{n+1})$ such that $k_1 + \ldots + k_d = 1$.

In general, with all $v_{\mathbf{k},\mathbf{j}}$ (and the corresponding $f_{\mathbf{k},\mathbf{j}}$) for which $k_1 + \ldots + k_d < K$ obtained, the equality Eq. (14) follows and can be used to compute the nodal value of f at point $\mathbf{x}_{k,\bar{j}}$ for $\bar{\mathbf{k}}$ satisfying $\overline{k_1}+\ldots+\overline{k_d}=K$,

$$f_{\bar{\mathbf{k}},\bar{\mathbf{j}}}+\sum_{\mathbf{k}\in\mathscr{S}_{K-1}}\sum_{\mathbf{j}\in\mathscr{I}_{\mathbf{k}}} f_{\mathbf{k},\mathbf{j}}\phi_{\mathbf{k},\mathbf{j}}(\mathbf{x}_{\bar{k},\bar{j}})=f\left(v_{\bar{\mathbf{k}},\bar{\mathbf{j}}}+\sum_{\mathbf{k}\in\mathscr{S}_{K-1}}\sum_{\mathbf{j}\in\mathscr{I}_{\mathbf{k}}} v_{\mathbf{k},\mathbf{j}}\phi_{\mathbf{k},\mathbf{j}}(\mathbf{x}_{\bar{k},\bar{j}})\right), \quad (14)$$

where

$$\mathbf{x}_{\bar{k},\bar{j}}:=\left(\frac{2\overline{j_1}-1}{2^{\overline{k_1}+1}},\ldots,\frac{2\overline{j_d}-1}{2^{\overline{k_d}+1}}\right). \quad (15)$$

Putting Eq. (14) into Eq. (9), we derive the local nonlinear system for $v_{\bar{\mathbf{k}},\bar{\mathbf{j}}}$ with $\bar{\mathbf{k}}$ satisfying $\overline{k_1}+\ldots+\overline{k_d}=K$. Recursively, we solve all unknowns in Eq. (9).

2.3. Error estimation, computational cost and stability

The error of the IIF2-SG-FEM method mainly comes from two sources: (i) spatial discretization by the sparse grids and (ii) the second-order IIF method for the temporal integration. Therefore, the estimated overall error is

$$O\left(\frac{(\log N_{\mathbf{x}})^{d-1}}{N_{\mathbf{x}}^2}\right)+O(\Delta t^2). \quad (16)$$

The major cost of the IIF2-SG-FEM method associated with the temporal integration arises from calculating the exponential of $M^{-1}D\Delta t$. Although this calculation only needs to be performed once before the temporal update, it can be very expensive for high dimensions (d) and fine spatial resolution ($N_{\mathbf{x}}$) because of the large size of the matrix of order $O(N_{\mathbf{x}}(\log N_x)d^{-1})$. In addition, at each time step, one must solve the local nonlinear system obtained by substituting Eq. (14) into Eq. (9), leading to higher costs. Later, we discuss other methods that can reduce the size of the exponential matrix to reduce the computational costs.

Because M is the mass matrix and D is associated with the Laplacian operator, $M^{-1}D\Delta t$ only contains non-negative eigenvalues, and the stability of IIF2-SG-FEM can be studied using a scalar case [25]. When the reaction term is linear, showing that IIF2-SG-FEM is A-stable is straightforward.

3. Array-representation semi-implicit integration factor method (AcIIF) with the sparse grids finite difference method

In this section, to reduce the size of the required exponential matrix and associated computational time, we construct the AcIIF method with the sparse grid finite difference scheme to solve Eq. (2), which has the same spatial domain and boundary conditions as that described in Section 2.

3.1. An introduction to the finite difference scheme on sparse grids

Following the methods previously studied in [15], let $V(t) = (v_{\mathbf{k},\mathbf{j}}(t))$ be the hierarchical value of the unknown u and $U(t) = (u(\mathbf{x}_{\mathbf{k},\mathbf{j}}, t))$ be the nodal value. The nodal-hierarchical transformations, denoted by H and M, are obtained using a dimensional splitting scheme [15]. Let H_i be the (nodal to) hierarchical basis transformation along direction x_i. It is equivalent to the one-dimensional (nodal to) hierarchical basis transformation acting on $U_{\mathbf{k}^i\mathbf{j}^i}(t)$ for all possible $\mathbf{k}^i$ and $\mathbf{j}^i$, where

$$\mathbf{k}^i := (k_1, \ldots, k_{i-1}, \cdot, k_{i+1}, \ldots, k_d), \quad \mathbf{j}^i := (j_1, \ldots, j_{i-1}, \cdot, j_{i+1}, \ldots, j_d), \quad (17)$$

and $U_{\mathbf{k}^i\mathbf{j}^i}(t)$ is a sub-vector of $U(t)$ obtained by fixing k_s, j_s, $s \neq i$ and only varying k_i and j_i. The (hierarchical to) nodal basis transformation along direction x_i, M_i, is defined similarly, except it acts on the hierarchical value $V(t)$ by a one-dimensional (hierarchical to) nodal transformation. The nodal-hierarchical transformation is the composition of H_i or M_i:

$$\overline{H}_i\overline{M}_i = I, \ \ \overline{H}U(t) = \overline{H}_1\overline{H}_2 \ldots \overline{H}_d U(t) = V(t), \ \ \overline{M}V(t) = \overline{M}_1\overline{M}_2 \ldots \overline{M}_d V(t) = U(t). \quad (18)$$

Define $H_{I/\{i\}} := H_1 \ldots H_{i-1} H_{i+1} \ldots H_d$ as the (nodal to) hierarchical basis transformation along all directions except for the direction x_i. We also define the finite difference operator D_i acting on any vector $U(t)$ for all possible $\mathbf{k}^i$ and $\mathbf{j}^i$ to be the regular one-dimensional central difference on $U_{\mathbf{k}^i,\mathbf{j}^i}(t)$. Then, the finite difference scheme on sparse grids for $\partial^2/\partial x_i^2$ is $\overline{H}^1_{I/\{i\}} \overline{D}_i \overline{H}_{I/\{i\}} U(t)$, with an estimated error [15] as follows:

$$O\left(\frac{(\log N_{\mathbf{x}})^{d-1}}{N_{\mathbf{x}}^2}\right). \quad (19)$$

With this approximation, Eq. (2) can be written into the following ODE system:

$$\frac{dU(t)}{dt} = \left(\sum_{1 \le i \le d} \overline{H}^{-1}_{I/\{i\}} \overline{D}_i \overline{H}_{I/\{i\}}\right) U(t) + f(U(t)). \quad (20)$$

By applying the second-order IIF method, we obtain the following second-order (in time) IIF with sparse grid finite difference method (**IIF2-SG-FD**):

$$U(t_{n+1}) = e^{A\Delta t}\left(U(t_n) + \frac{\Delta t}{2} f(U(t_n))\right) + \frac{\Delta t}{2} f(U(t_{n+1})), \quad (21)$$

where $A = \sum_{1 \le i \le d} \overline{H}^{-1}_{I/\{i\}} \overline{D}_i \overline{H}_{I/\{i\}}$. In the IIF2-SG-FD scheme, we need to compute and store the exponential of an $O(N_{\mathbf{x}} (\log N_{\mathbf{x}})^{d-1} \times N_{\mathbf{x}} (\log N_{\mathbf{x}})^{d-1})$ matrix. When $N_{\mathbf{x}}$ and d are large, the computation is intractable.

3.2. AcIIF method with the sparse grid finite difference method

Here, we apply the array representation to $U(t)$ and $V(t)$ to decompose them into smaller vectors [29]. Starting from the hierarchical value $V(t)$, for a fixed direction x_i and $\mathbf{k}^i$, $\mathbf{j}^i$, varying k_i and j_i gives a vector of $T(\mathbf{k}^i, \mathbf{j}^i)$ elements. Denote it by $V_{\mathbf{k}^i,\mathbf{j}^i}(t)$, where

$$T(\mathbf{k}^i,\mathbf{j}^i)=\sum_{s=0}^{k_i^{\max}} 2^s = 2^{k_i^{\max}+1}-1,\quad k_i^{\max}=K-\sum_{s\neq i}k_s. \quad (22)$$

Let $\mathbf{k}^i$ and $\mathbf{j}^i$ go through all possible values; the resulting smaller vectors form the original vector $V(t)$. We denote $\{(\mathbf{k}^i, \mathbf{j}^i) : \Sigma_{r\neq i}\, k_r \le K, j_r = 1, 2, \ldots, 2^{k_r}, r \neq i\}$ by $\mathscr{I}_i$ and use the notation $\otimes$ for such a representation:

$$V(t)=\bigotimes_{(\mathbf{k}^i,\mathbf{j}^i)\in\mathscr{I}_i} V_{\mathbf{k}^i,\mathbf{j}^i}(t) \quad (23)$$

Note that this representation is slightly different from the original array representation, where all of the subarrays have the same length [29]. In this case, the different subvectors $V_{\mathbf{k}^i,\mathbf{j}^i}(t)$ have different sizes $T(\mathbf{k}^i, \mathbf{j}^i)$, and the sizes are smaller than $2N_{\mathbf{x}} - 1$. With the array representation, we can use smaller matrices to represent the (hierarchical to) nodal transformation along direction x_i:

$$\overline{M}_i V(t)=\bigotimes_{(\mathbf{k}^i,\mathbf{j}^i)\in\mathscr{I}_i} M_{\mathbf{k}^i,\mathbf{j}^i} V_{\mathbf{k}^i,\mathbf{j}^i}(t), \quad (24)$$

and the central difference scheme $\overline{D}_i$:

$$\overline{D}_i V(t)=\bigotimes_{(\mathbf{k}^i,\mathbf{j}^i)\in\mathscr{I}_i} D_{\mathbf{k}^i,\mathbf{j}^i} V_{\mathbf{k}^i,\mathbf{j}^i}(t). \quad (25)$$

Here, $M_{\mathbf{k}^i,\mathbf{j}^i}$ denotes the one-dimensional (hierarchical to) nodal transformation acting on $V_{\mathbf{k}^i,\mathbf{j}^i}(t)$, and $D_{\mathbf{k}^i,\mathbf{j}^i}$ is the tridiagonal matrix:

$$D_{\mathbf{k}^i,\mathbf{j}^i}=\frac{1}{\mathbf{T}(\mathbf{k}^i,\mathbf{j}^i)^2}\begin{pmatrix} -2 & 1 & & 0 \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & -2 \end{pmatrix}. \quad (26)$$

Both matrices have a size of $T(\mathbf{k}^i, \mathbf{j}^i) \times T(\mathbf{k}^i, \mathbf{j}^i)$, which are at most $(2N_{\mathbf{x}} - 1) \times (2N_{\mathbf{x}} - 1)$. Define $B_{\mathbf{k}^i,\mathbf{j}^i}=M^{-1}_{\mathbf{k}^i,\mathbf{j}^i} D_{\mathbf{k}^i,\mathbf{j}^i} M_{\mathbf{k}^i,\mathbf{j}^i}$, and apply the fact that $\overline{H}^{-1}_{I/\{i\}}\overline{D}_i\overline{H}_{I/\{i\}}U=\overline{H}^{-1}\left(\overline{M}_i^{-1}\overline{D}_i\overline{M}_i\right)\overline{H}U=\overline{H}^{-1}\left(\overline{M}_i^{-1}\overline{D}_i\overline{M}_i\right)V$; under the array representation, the exponential of $\overline{H}^{-1}_{I/\{i\}}\overline{D}_i\overline{H}_{I/\{i\}}\Delta t$ acting on $U(t)$ becomes equivalent to

$$e^{\overline{M}_i^{-1}\overline{D}_i\overline{M}_i\Delta t}V(t)=\bigotimes_{(\mathbf{k}^i,\mathbf{j}^i)\in\mathscr{I}_i} e^{(B_{\mathbf{k}^i,\mathbf{j}^i}\Delta t)}V_{\mathbf{k}^i,\mathbf{j}^i}(t), \quad (27)$$

which is then multiplied by $\bar{H}^{-1}$ to obtain the nodal value. Compared with the IIF method in which the exponential matrix is $O(N_{\mathbf{x}}(\log N_{\mathbf{x}})^{d-1}\times N_{\mathbf{x}}(\log N_{\mathbf{x}})^{d-1})$, the matrices required in the array representation, $B_{\mathbf{k}^i,\mathbf{j}^i i}$, are at most $(2N_{\mathbf{x}}-1)\times(2N_{\mathbf{x}}-1)$.

Through the array representation, i.e., utilizing multiple matrices of much smaller sizes instead of a single large exponential matrix, we obtain the array-representation semi-implicit integration factor method (**AcIIF2-SG-FD**), which can be summarized in the following steps:

i. Transform the nodal value of $(U(t_n)+\frac{\Delta t}{2}f(U(t_n)))$ to the hierarchical value $W^1(t)$,

$$W^1(t_n)=\overline{H}\left(U(t_n)+\frac{\Delta t}{2}f(U(t_n))\right), \quad (28)$$

ii. Adopt array representation to compute the exponential matrices and vector multiplication dimension by dimension:

$$W^2(t_n)=\bigotimes_{(\mathbf{k}^1,\mathbf{j}^1)\in\mathscr{I}_1} e^{(B_{\mathbf{k}^1,\mathbf{j}^1}\Delta t)}\left(\cdots\bigotimes_{(\mathbf{k}^d,\mathbf{j}^d)\in\mathscr{I}_d} e^{(B_{\mathbf{k}^d,\mathbf{j}^d}\Delta t)}W^1_{\mathbf{k}^d,\mathbf{j}^d}(t_n)\right), \quad (29)$$

iii. Update the nodal value by solving the nonlinear system:

$$U(t_{n+1})=\overline{H}^{-1}W^2(t_n)+\frac{\Delta t}{2}f(U(t_{n+1})). \quad (30)$$

3.3. Error estimation, computational cost, stability and high-order (in time) method

Naturally, the errors for the IIF2-SG-FD and AcIIF2-SG-FD methods take forms similar to that described in Eq. (16) because the sources of the discretization errors are similar.

Computing large matrix exponentials are required in IIF2-SG-FD and IIF2-SG-FEM. In contrast, in the AcIIF2-SG-FD method, the computational cost of the matrix exponential is significantly reduced, by a factor of at least $(\log N_{\mathbf{x}})^{d-1}/2$. Even though some additional hierarchical-nodal transformations are introduced at each time step in AcIIF2-SG-FD, the cost is negligible relative to the cost of calculating the matrix exponentials because the fast computation of these transformations is straightforward [18]. Similar to IIF2-SG-FEM, the stability of both the IIF2-SG-FD and AcIIF2-SG-FD methods can be analyzed through the scalar case, and the methods are linearly *A*-stable [25].

To improve the order of accuracy for temporal integration, one can apply the third-order IIF (IIF3) to Eq. (20) in a straightforward way, leading to the IIF3-SG-FD method:

$$U(t_{n+1})=e^{A\Delta t}U(t_n)+\Delta t\left(\frac{5}{12}f(U(t_{n+1}))+\frac{2}{3}e^{A\Delta t}f(U(t_n))-\frac{1}{12}e^{2A\Delta t}f(U(t_{n-1}))\right), \quad (31)$$

with error:

$$O(\frac{(\log N_{\mathbf{x}})^{d-1}}{N_{\mathbf{x}}^2})+O(\Delta t^3). \quad (32)$$

4. AcIIF with sparse grids combination technique

In the previous sections, we incorporated the IIF and AcIIF methods with sparse grids in the finite element or finite difference methods. In both approaches, the formulations are most effective in dealing with the Laplacian differential operators in Eq. (1). Here, we incorporate a sparse grid combination technique with the AcIIF approach for more general cases for solving Eq. (1), where the spatial partial differential operator P may contain cross-derivatives or non-constant diffusion coefficients.

In the sparse grid combination technique [19], instead of using non-uniform grids in a typical sparse grid discretization, the combination technique solves the PDEs on different levels of uniform grids and then combines the solutions for the final solution [19, 30]. The sparse grid combination technique typically consists of two major steps:

Step 1: Choose the spatial resolution $N_{\mathbf{x}} = 2^K$, and solve Eq. (1) on different uniform grids defined as below:

$$(2^{k_1}\times 2^{k_2}\times\ldots\times 2^{k_d}),\quad K\le k_1+k_2+\ldots+k_d\le K+d-1,\quad k_i\ge 1, \quad (33)$$

where 2^{k_i} is the grid number along direction x_i. The result on each grid is denoted by $w_{k_1,\ldots,k_d}(\mathbf{x}, t)$.

Step 2: Combine all the solutions at different uniform grid levels as follows:

$$w_K(\mathbf{x},t)=\sum_{m=K}^{K+d-1}(-1)^{d-1-m+K}\begin{pmatrix} d-1 \\ m-K\end{pmatrix}\sum_{k_1+k_2+\ldots+k_d=m} w_{(k_1,\ldots,k_d)}(\mathbf{x},t). \quad (34)$$

If each individual solution $w_{k_1,\ldots,k_d}(\mathbf{x}, t)$ to the exact solution $u(\mathbf{x}, t)$ satisfies

$$\|u(\mathbf{x},t)-w_{(k_1,\ldots,k_d)}(\mathbf{x},t)\|=\sum_{i,j=1}^{d} O\left(\frac{1}{2^{k_i}2^{k_j}}\right), \quad (35)$$

then the overall error of $w_K(\mathbf{x}, t)$ is [19]

$$\|u(\mathbf{x},t)-w_K(\mathbf{x},t)\|=O\left(\frac{(\log N_{\mathbf{x}})^{d-1}}{N_{\mathbf{x}}^2}\right). \quad (36)$$

In the sparse grid combination technique, each subproblem contains 2^m grid points for $K \le m \le K + d - 1$, and the overall approximated solution has an error similar to the error of a typical sparse grid method. Each subproblem is independent of the other; thus, the implementation is much simpler than that for the finite difference or element approaches, and distributed (or parallel) calculations of each subproblem become straightforward.

If the explicit methods (e.g., RK2) are used to solve each subproblem, due to the stability constraints, the time step must be $\Delta t \sim 1/N_{\mathbf{x}}^2$. Using IIF or AcIIF, the stability constraint for each subproblem is relaxed; in particular, the second-order IIF or AcIIF leads to no constraint on the temporal stability. If we set the uniform time step for each subproblem, e.g., $\Delta t \sim 1/N_{\mathbf{x}}$ in AcIIF, the overall error in both space and time for each subproblem takes the form of Eq. (35).

In particular, on a uniform grid, after spatial discretization, the solution of Eq. (1), U, can be expressed as a d-dimensional array. The array representation for $\mathcal{L}_{x_i x_j}$, the partial differential operator along the x_i and x_j directions, is denoted by $\mathscr{A}_{x_i x_j}^{(k_r),r\neq i,j}$. Thus, we obtain

$$U=\bigotimes_{1\le k_r\le N_r, r\neq i,j} U|_{x_i,x_j}^{(k_r),r\neq i,j}, \quad \mathscr{L}_{x_i x_j}U=\bigotimes_{1\le k_r\le N_r, r\neq i,j} \mathscr{A}_{x_i x_j}^{(k_r),r\neq i,j} U|_{x_i,x_j}^{(k_r),r\neq i,j}, \quad (37)$$

where N_r is the number of grid points in the x_r direction. The exponential of $\mathcal{L}_{x_i x_j}$ is computed by the array representation:

$$e^{\mathscr{L}_{x_i x_j}\Delta t}U=\bigotimes_{1\le k_r\le N_r, r\neq i,j} e^{\mathscr{A}_{x_i x_j}^{(k_r),r\neq i,j}\Delta t} U|_{x_i,x_j}^{(k_r),r\neq i,j}. \quad (38)$$

Applying the implicit integration factor method yields the following second-order AcIIF method for Eq. (1)

$$U(t_{n+1})-\frac{\Delta t}{2}F(U(t_{n+1}))=\bigotimes_{\substack{1\le k_r\le N_{x_r}\\ r\neq 1,2}} e^{\mathscr{A}_{x_1x_2}^{(k_r),r\neq 1,2}\Delta t/2}\cdots\bigotimes_{\substack{1\le k_r\le N_{x_r}\\ r\neq d-1,d}} e^{\mathscr{A}_{x_{d-1}x_d}^{(k_r),r\neq d-1,d}\Delta t/2} \bigotimes_{\substack{1\le k_r\le N_{x_r}\\ r\neq d-1,d}} e^{\mathscr{A}_{x_{d-1}x_d}^{(k_r),r\neq d-1,d}}\cdots\bigotimes_{\substack{1\le k_r\le N_{x_r}\\ r\neq 1,2}} e^{\mathscr{A}_{x_1x_2}^{(k_r),r\neq 1,2}\Delta t/2}V(t_n), \quad (39)$$

where $V(t_n) = U(t_n) + \Delta t/2F(U(t_n))$. The derivation of Eq. (39) is based on direct application of compact implicit integration factor method if $\mathscr{A}_{x_i x_j}^{(k_r),r\neq i,j}$s are commutative. Otherwise, a splitting similar to Strang splitting method [31] needs to be used as shown in our previous work [29] to decompose the exponential matrices of Eq. (38) to obtain Eq. (39) for a second order method. The method is linearly absolute stable regardless of the commutative property

[29]. Overall, the total computational cost for AcIIF along with the sparse grid combination technique is significantly reduced upon utilizing both advantages.

5. Numerical simulations

In this section, we investigate the three different sparse grid methods, all of which are based on the IIF or AcIIF temporal integration approach by applying them to five different PDE systems, where some have exact solutions. In particular, we examine the convergence and computational efficiency of the new methods and compare them with other existing methods for several cases.

5.1. A nonlinear reaction-diffusion system

We first implement the IIF2-SG-FEM method on the following nonlinear reaction-diffusion equation:

$$u_t(x,y,t)=\frac{1}{2\pi^2}(u_{xx}(x,y,t)+u_{yy}(x,y,t))+u(x,y,t)^2-e^{-2t}\sin^2\pi x\sin^2\pi y, \quad (40)$$

with zero Dirichlet boundary conditions and the initial condition:

$$u(x,y,0)=\sin\pi x\sin\pi y. \quad (41)$$

The exact solution for this system is

$$u(x,y,t)=e^{-t}\sin\pi x\sin\pi y. \quad (42)$$

Direct implementation of the weak formula Eq. (5) using the implicit methods (e.g., Crank-Nicolson) requires solving large nonlinear systems at each time step because of the presence of the nonlinear reaction $u(x, y, t)^2$. Instead, we only make a comparison between the IIF2-SG-FEM method and RK2 for this case. In the RK2 method, if one chooses the time step $\Delta t=1/N_x^2$, it is convergent, whereas a larger time step $\Delta t = 1/N_x$ leads to instability. In contrast, numerical results show that the IIF2-SG-FEM, in which the time step is chosen to be $\Delta t = 1/N_x$, is convergent with the expected order of accuracy. The error of the IIF2-SG-FEM method is plotted in Figure 1. Because the discretization error is $O\left(\frac{(\log N_{\mathbf{x}})^{d-1}}{N_{\mathbf{x}}^2}\right)$ analytically, the errors measured under the log-log scale on different spatial resolutions should follow a straight line with a slope of one. This is shown in Figure 1, where the line of errors is almost parallel to the red line when the spatial resolution increases. In general, IIF2 can use a much larger time step in $O(1/N_{\mathbf{x}})$ instead of $O(1/N_{\mathbf{x}}^2)$ for the RK2; thus, the efficiency of IIF2 is better than that of RK2.

5.2. A linear d-dimensional reaction-diffusion equation

We implement the AcIIF2-SG-FD method on the equation

$$u_t(x_1,\ldots,x_d,t)=\frac{1}{\pi^2}\Delta_d u(x_1,\ldots,x_d,t)+(d-1)u(x_1,\ldots,x_d,t), \quad (43)$$

with the initial condition

$$u(x_1,\ldots,x_d,0)=\prod_{i=1}^{d}\sin\pi x_i, \quad (44)$$

and zero Dirichlet boundary condition for d = 2, 3, 4, 6 respectively. The exact solution for the equation is:

$$u(x_1,\ldots,x_d,t)=e^{-t}\prod_{i=1}^{d}\sin\pi x_i. \quad (45)$$

The L^∞ error versus $(\log N_{\mathbf{x}})^{d-1}/N_{\mathbf{x}}^2$ for different spatial mesh sizes is plotted in Figure 2 to show the order of accuracy. The time step is set as $\Delta t = 1/N_{\mathbf{x}}$, and all simulations are terminated at $t = 0.5$ except that $t=\frac{1}{2^4}$ for $d = 6$. Compared with the red reference line with a slope of 1, in different dimensions, the numerical errors always fall on a straight line that is almost parallel to the red line as the spatial resolution increases, showing the expected order of accuracy.

One major advantage of the AcIIF method is the storage. For higher dimensions, such as d = 6, 8, 10, both IIF and RK2 methods rehire much more storage space. For example, when d = 8 and K = 6, IIF2-SG-FD and RK2 methods run out of memory for a computer of 16 GB memory whereas the AcIIF2-SG-FD method only requires matrix exponentials with a size no larger than 128 × 128. Nevertheless, when d = 8, 10, the CPU time taking for the standard sparse grid (mostly due to the transformations on different grid levels) is still very costly, in particular, for large K.

Another advantage of AcIIF is the efficiency. To demonstrate it, we compare it with IIF2 and RK2 based on the sparse grid finite difference method. Constrained by the stability condition, the time step for RK2 must be $\Delta t=1/N_{\mathbf{x}}^2$. For the other two methods, the time step is $1/N_{\mathbf{x}}$. The CPU times consumed by different methods are listed in Table 1 for d = 2, 3, 4. As shown in the table, although RK2 requires less computational time on a coarse mesh, the IIF2 method is more efficient when the mesh is refined in a low-dimensional space. Overall, AcIIF2 performs most efficiently on the fine mesh across different dimensions. A detailed analysis shows that for the IIF2-SG-FD method, the most time-consuming part is calculating the matrix exponential, consuming around 90 percent of the overall CPU time. In the AcIIF2-SG-FD method, the corresponding time spent on this calculation is significantly decreased. We also compare the CPU time by these methods when the error is comparable on different dimensions. For example, the error at N = 32 (or N = 64) for d = 3 is comparable to the error level at N = 64 (or N = 128) for the case of d = 4 in Tables 1 whereas the ratio between the computational time of AcIIF2 and RK2 changes

significantly less than the ratio between $d = 3$ and $d = 4$ at the resolution $N = 64$ (or $N = 128$).

5.3. A diffusive logistic equation

The diffusive logistic equation in the following form:

$$\frac{\partial u}{\partial t}(\mathbf{x},t)=D\Delta_{\mathbf{x}}u+gu-u^2,\mathbf{x}\in\Omega,t>0, \quad (46)$$

describes the population density evolution, in which the unknown u denotes the population density of a species at time t and at location $\mathbf{x}$ [32]. The smooth function g defines the birth rate of the species, which may be positive or negative. Various numerical methods have been developed to solve this type of equation [33, 34]. We use the AcIIF2-SG-FD method to solve the diffusive logistic equation in different dimensions. To study the accuracy of the method, we set up the equation for which an exact solution can be constructed. Let $\Omega_d = [0, 1]^d$,

$$\begin{gathered}\mathbf{x}=(x_1,\ldots,x_d)\in\Omega_d,\quad D=\tfrac{1}{d\pi^2},\quad u(\mathbf{x},0)=\prod_{i=1}^{d}\sin\pi x_i,\\ g(\mathbf{x},t)=e^{-t}\prod_{i=1}^{d}\sin\pi x_i\end{gathered} \quad (47)$$

and the zero Dirichlet boundary condition is imposed. The exact solution for this equation is

$$u(\mathbf{x},t)=e^{-t}\prod_{i=1}^{d}\sin\pi x_i. \quad (48)$$

The implementation is the same as that for Section 5.2, except that the Newton method with a tolerance of 10^{-6} is adopted to solve the small nonlinear system at each grid point, similar to the standard IIF method [25]. The log-log error plot for $d = 2, 3, 4, 6$ is displayed in Figure 3. From these plots, we can see that in different dimensions, the errors always follow a line with a slope close to 1 as the spatial resolution increases, consistent with the expected order of accuracy.

We next compare the CPU time used for AcIIF2-SG-FD and the RK2 method coupled with sparse grid finite difference for $d = 2, 3, 4$. The time step for RK2 is $1/N_{\mathbf{x}}^2$ to guarantee that the stability condition will be satisfied. The CPU times for both methods are provided in Table 2. It is clear that regardless of dimensions of the system, AcIIF2 reduces the computational time significantly, particularly with fine meshes.

5.4. A three-dimensional reaction-diffusion system with cross-derivatives

To deal with cross-derivatives in the differential operators, we implement the AcIIF2 method with the sparse grid combination technique on this three-dimensional PDE:

$$u_t=(0.1u_{xx}-0.15u_{xy}+0.1u_{yy})+(0.1u_{xx}+0.2u_{xz}+0.2u_{zz})+(0.2u_{yy}+0.15u_{yz}+0.1u_{zz})+0.3u \quad (49)$$

where $x, y, z \in (0, 2\pi)$, with a periodic boundary condition and the initial condition:

$$u(x, y, z, 0) = \sin(x+y+z). \quad (50)$$

The exact solution for the system is

$$u(x, y, z, 0) = e^{-0.2t}\sin(x+y+z). \quad (51)$$

The L^∞ error is plotted in Figure 4, which exhibits the correct and expected order of accuracy, i.e., parallel to the line of reference when the spatial resolution becomes large enough. Compared with the IIF method, AcIIF is clearly more effective in dealing with the matrix exponentials, as seen in Table 3, in which the CPU times for calculating the matrix exponentials, recognized to be the dominant cost for both methods, are listed.

5.5. A four-dimensional Fokker-Planck equation

In biology, the Fokker-Planck equation (FPE) is often used to describe and analyze the temporal evolution of the probability density functions for species in biochemical networks [2]. The generalized FPE takes the form:

$$\frac{\partial p(\mathbf{x}, t)}{\partial t} = -\sum_{r=1}^{R}\left\{\sum_{i=1}^{d} n_{ri}\frac{\partial}{\partial x_i}\left(q_r(\mathbf{x}, t) - \frac{1}{2}\sum_{j=1}^{d} n_{rj}\frac{\partial q_r(\mathbf{x}, t)}{\partial x_j}\right)\right\}. \quad (52)$$

In the FPE, R denotes the total number of chemical reactions taking place in the biochemical network, and d denotes the number of different species participating in the reactions. $\mathbf{x} = (x_1, \ldots, x_d)$ denotes a particular biochemical state, where each component x_i denotes the copy number of the i-th reactant. The unknown function, $p(\mathbf{x}, t)$, is the probability density function for each state $\mathbf{x}$ at time t. n_{ri} represents the change in the copy number of reactant i when the r-th reaction occurs once. In addition, one can define

$$q_r(\mathbf{x}, t) = w_r(\mathbf{x}, t)p(\mathbf{x}, t), \quad (53)$$

where $w_r(\mathbf{x}, t)$ is the reaction propensity function for the r-th reaction at state $\mathbf{x}$.

FPE is a d-dimensional differential equation with non-constant diffusion coefficients and second-order cross-derivatives. Previously, the AcIIF scheme coupled with the finite difference method has been applied to the FPE on uniform grids, for which the AcIIF method has good efficiency[29]. Here, we implement the AcIIF method with a sparse grid combination technique to study the following biochemical reaction system [35, 36, 29]. In the metabolite-enzyme system, there are two metabolites A, B and two enzymes E_A, E_B:

$$\begin{array}{c} \varnothing \xrightarrow{\frac{k_A \cdot e_A}{1+a/K_I}} A \quad \varnothing \xrightarrow{\frac{k_B \cdot e_B}{1+b/K_I}} B \\ A+B \xrightarrow{k_2 \cdot a \cdot b} \varnothing \\ A \xrightarrow{\mu \cdot a} \varnothing \quad B \xrightarrow{\mu \cdot b} \varnothing \\ \varnothing \xrightarrow{\frac{k_{E_A}/A}{1+a/K_R}} E_A \quad E_A \xrightarrow{\mu \cdot e_A} \varnothing \\ \varnothing \xrightarrow{\frac{k_{E_B}}{1+b/K_R}} E_B \quad E_B \xrightarrow{\mu \cdot e_B} \varnothing \end{array} \tag{54}$$

where we set $k_A = k_B = 0.3s^{-1}$, $k_2 = 0.001s^{-1}$, $K_I = 60$, $\mu = 0.002s^{-1}$, $k_{E_A} = k_{E_B} = 0.02s^{-1}$ and $K_R = 30$.

To solve this FPE, we choose the domain of the state variables to be [0, 100] × [0, 100] × [0, 50] × [0, 50]. The size of the domain is large enough such that the probability outside of the domain is sufficiently small and can be neglected. No boundary condition is needed for $x_i = 0$ due to the degradation of Eq. (52). Otherwise, the zero Dirichlet boundary condition is imposed.

We start with a Gaussian distribution centered at (30, 40, 15, 12) with a standard deviation of $\sqrt{40}$ as the initial condition for p. There are nine reactions in the system, and for the AcIIF method, we group these reactions if their associated cross-derivative term is present. In this case, the first to fifth reactions are grouped in a one array-representation operator, the sixth and seventh are grouped into one, and the eighth and ninth are grouped into another. In Figure 5, we plot the density distribution at $t = 45$ solved using the sparse grid method. We also plot the trace of the ODE system, which is consistent with the FPE solution, suggesting the correctness of our approach.

To use the sparse grid combination technique, we first set up a spatial resolution $N_{\mathbf{x}} = 2^K$ and then solve the FPE for each of its subproblems. To meet the accuracy requirement, we choose $\Delta t = 1/\max_i N_{x_i}$ for the AcIIF approach. We use multiple cores to run the simulation and then combine all the results. We also implement RK2 to solve each problem, and we observe that, to maintain convergence, for a certain subproblem, e.g., $N_{x_1} = N_{x_2} = \ldots = N_{x_d-1} = 2$ and $N_{x_d} = 2^{K+1-d}$, a very small time step must be used. As a result, RK2 requires significantly more time for these problems than the AcIIF approach, suggesting that AcIIF is a better approach.

6. Conclusions and Discussions

The sparse grid method is one of the major approaches used to reduce the computational cost associated with spatial discretization in solving PDEs of high dimensions. For a time-dependent problem, the size of the time step, dictated by the choice of temporal integrator, becomes critical, particularly for systems with strong stiffness in reactions or large diffusion coefficients. The implicit Integration Factor (IIF), which treats the reaction implicitly and handles diffusion by exact integration, is clearly a natural choice for solving stiff PDEs in high dimensions.

In this paper, we have combined the sparse grid approach with IIF methods for a new class of methods to solve reaction-diffusion equations. We have integrated IIF methods with three different sparse grid approaches: finite element, finite difference, and a multi-level combination technique. In addition, we have employed an array compact representation of IIF (AcIIF) previously developed for uniform and regular grids to reduce the cost associated with storage and computing the exponentials of matrices in sparse grids. After studying various combinations of different sparse grid techniques and IIF methods, we have found that the approach based on AcIIF and the sparse grid combination technique is most effective for solving reaction-diffusion equations, especially for the system containing cross-derivatives and non-constant coefficients.

Because the sparse grid combination technique has intrinsic distributed structures, parallel implementation becomes straightforward, in comparison with the finite difference or the finite element methods in combination with IIF. Therefore, a GPU implementation can be adopted to significantly improve the efficiency of the sparse grid combination technique integrated with IIF. So far, most of the systems we deal with have a Dirichlet boundary condition on a rectangular domain. It is possible to extend the sparse grid-based IIF method to problems with complicated boundary conditions as discussed in [37] or irregular domains. The order of accuracy in space can be improved by introducing higher-order polynomials as the hierarchical bases, particularly for problems with high-order or mixed derivatives. The integration factor methods on sparse grids can likely be used in broad applications involving PDEs in high dimensions such as Fokker-Planck equations in biology and Black-Scholes equations in finance.

Acknowledgments

This work was supported by the National Institutes of Health grants R01GM107264 and P50GM76516 and the National Science Foundation grant DMS1161621.

References

1. Lander A, Nie Q, Wan F. Do Morphogen Gradient Arise by Diffuion? Developmental Cell. 2002; 2:785–796. [PubMed: 12062090]
2. Risken, H. The Fokker-Planck Equation: Methods of Solutions and Applications. Springer; 1996.
3. Zhu, Y.; Wu, X.; Chern, I. Derivative securities and difference methods. Springer Verlag; New York: 2004.
4. Griffths R. The frequency spectrum of a mutation, and its age, in a general diffusion model. Theor Popul Biol. 2003; 64:241–251. [PubMed: 12948685]
5. Bungartz H-J, Griebel M. Sparse grids. Acta Numer. 2004; 13:147–269.
6. Bungartz H-J, Griebel M. A note on the complexity of solving Poisson's equation for spaces of bound mixed derivatives. J Complexity. 1999; 15:167–199.
7. Deslauries G, Dubuc S. Symmetric iterative interpolation processes. Constr Approx. 1989; 5:51–150.
8. Smolyak S. Quadrature and interpolation formulas for tensor products of certain classes of functions. Soviet Math Dokl. 1963; 4:240–243.
9. Bungartz, H-J. Concepts for higher order finite elements on sparse grids. In: Ilin, AV.; Scott, LR., editors. Proceedings of the 3rd Int Conf on Spectral and High Order Metehods, in Houston Journal of Mathematics; 1995.

10. Lin G, Su C-H, Karniadakis G. Stochastic modeling of random roughness in shock scattering problems: Theory and simulations, Computer Methods in Applied Mechanics and Engineering. Stochastic Modeling of Multiscale and Multiphysics Problems. 2008; 197:3420–3434.
11. Foo J, Karniadakis GE. Multi-element probabilistic collocation method in high dimensions. J Comput Phys. 2010; 229:1536–1557.
12. Zhang Z, Tretyakov M, Rozovskii B, Karniadakis G. A recursive sparse grid collocation method for differential equations with white noise. SIAM Journal on Scientific Computing. 2014; 36:A1652–A1677.
13. Zenger, C. Sparse grids. In: Hackbusch, W., editor. Parallel Algorithms for Partial Differential Equations, volume 31, Notes on Numerical Fluid Mechanics. 1991.
14. Griebel, M. A parallelizable of vectorizble multi-level algorithm on sparse grids. In: Hackbusch, W., editor. Parallel Algorithms for Partial Differential Equations, volume 31, Notes on Numerical Fluid Mechanics. 1991.
15. Griebel M. Adaptive sparse grid multilevel methods for elliptic PDEs based on finite differences. Computing. 1998; 61:151–179.
16. Hemker P. Sparse-grid finite-volume multi grid for 3D problems. Adv Comput Math. 1995; 4:83–110.
17. Kupka, F. PhD thesis. Insitut fur Mathematik, Universitat Wien; 1997. Sparse grid spectral methods for the numerical solution of partial differential equations with periodic boundary conditions.
18. Shen J, Yu H. Efficient spectral sparse grid methods and applications to high-dimensional elliptic problems. SIAM JSciComput. 2010; 32:3228–3250.
19. Griebel, M.; Schneider, M.; Zenger, C. A combination technique for the solution of sparse grid problems. In: Groen, P.; Beauwens, R., editors. Iterative Methods in Linear Algebra. 1992. p. 263-281.
20. Kassam A, Lloyd, Trefethen N. Fourth-order time stepping for stiff PDEs. SIAM J Sci Comput. 2005; 26:1214–1233.
21. Kleefeld B, Khaliq A, Wade B. An ETD Crank-Nicolson method for reaction-diffusion systems. Numer Methods Partial Differential Eq. 2012; 28:1309–1335.
22. Cox S, Matthews P. Exponential time differencing for stiff systems. J Comput Phys. 2002; 176:430–455.
23. Du Q, Zhu W. Stability analysis and application of the exponential time differencing schemes. J Comput Math. 2004; 22
24. Krogstad S. Generalized integrating factor methods for stiff PDEs. J Comput Phy. 2005; 203:72–88.
25. Nie Q, Zhang Y-T, Zhao R. Efficient semi-implicit schemes for stiff systems. J Comput Phys. 2006; 214:521–537.
26. Zhao S, Ovadia J, Liu X, Zhang Y-T, Nie Q. Operator splitting implicit integration factor methods for stiff reaction-diffusion-advection systems. J Comput Phys. 2011; 230:5996–6009. [PubMed: 21666863]
27. Liu X, Nie Q. Compact integration factor methods for complex domains and adaptive mesh refinement. J Comput Phys. 2010; 229:5692–5706. [PubMed: 20543883]
28. Nie Q, Wan F, Zhang Y-T, Liu X. Compact integration factor methods in high spatial dimensions. J Comput Phys. 2008; 227:5238–5255. [PubMed: 19809596]
29. Wang D, Zhang L, Nie Q. Array-representation integration factor method for high-dimensional systems. J Comput Phys. 2014; 258:585–600.
30. Leentvaar C, Oosterlee C. On coordinate transformation and grid stretching for sparse grid pricing of basket options. Journal of Computational and Applied Mathematics. 2008; 222:193–209.
31. Strang G. On the construction and comparison of difference schemes. SIAM J Numer Anal. 1968; 5:506–517.
32. Hadeler K, Lewis M. Spatial dynamics of the diffusive logistic equation with a sedentary compartment. Canadian Appl Math Quart. 2002; 10:473–499.

33. Afrouzi G, Khademloo S. Numerical solutions of diffusive logistic equation. Chaos, Solitons and Fractals. 2007; 31:112–118.
34. Afrouzi G, Naghizadeh Z, Mahdavi S. A numerical method for finding positive solution of logistic equation. Applied Mathematics and Computation. 2007; 186:1497–1501.
35. Ferm L, Lotstedt P, Sjoberg P. Conservative solution of the fokker-planck equation for stochastic chemical equation. Technical Report. 2004
36. Sjoberg P, Lotstedt P, Elf J. Fokker-planck approximation of the master equation in molecular biology. Comput Visual Sci. 2009; 12:37–50.
37. Ju L, Liu X, Leng W. Compact implicit integration factor methods for a family of semilinear fourth-order parabolic equations. Discrete and Continuous Dynamical Systems - Series B. 2014; 19:1667–1687.

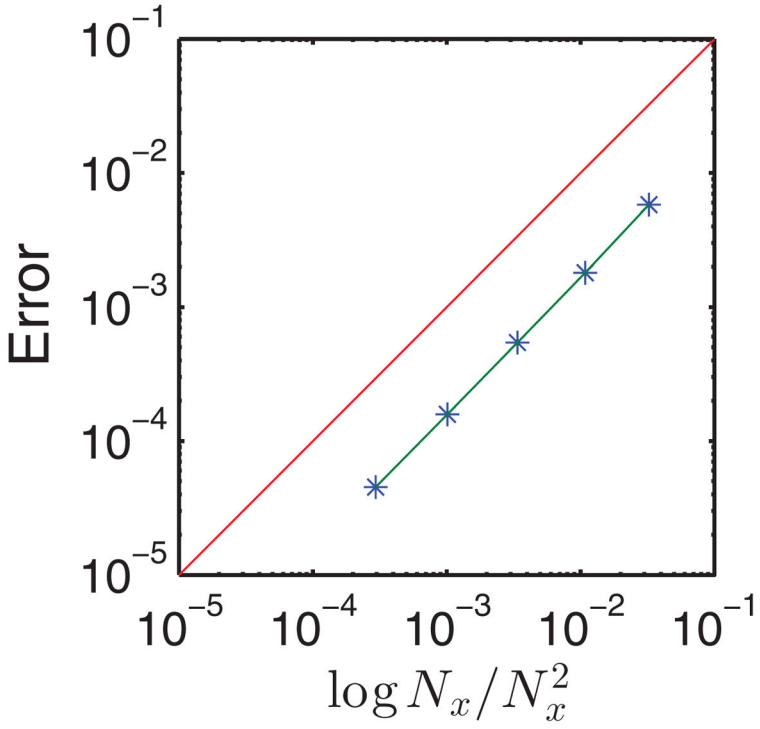

Figure 1.

The log-log plot of L^∞ error versus the quantity $\log N_x/N_x^2$ of IIF2-SG-FEM in Eq. (40). The spatial resolutions are N_x = 8, 16, 32, 64, 128, the time step is $\Delta t = 1/N_x$, and all simulations end at t = 1. The blue markers denote numerical errors, whereas the red line is a straight line with a slope of 1.

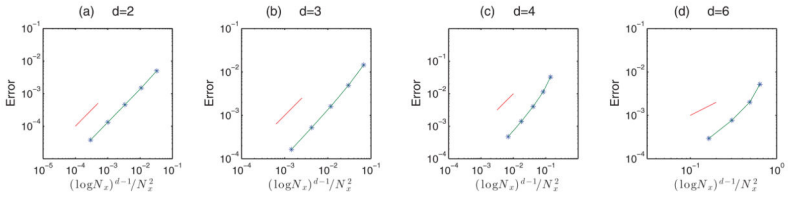

Figure 2.

The log-log plot of L^∞ error versus $(\log N_x)^{d-1}/N_x^2$ for the AcIIF2-SG-FD method for Eq. (43) for d = 2, 3, 4, 6. The spatial resolutions are N_x = 8, 16, 32, 64, 128 for d = 2, 3, 4 and N_x = 16, 32, 64, 128 for d = 6, and the time step is $\Delta t = 1/N_x$. The green line with blue markers denotes the solution error at final time, and the red one is a reference line with a slope of 1.

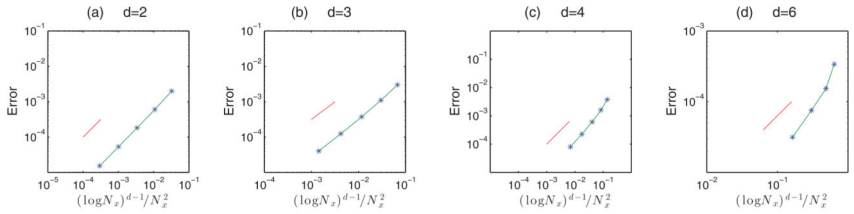

Figure 3.

The log-log plot of L^∞ error versus $(\log N_{\mathbf{x}})^{d-1}/N_{\mathbf{x}}^2$ for the AcIIF2-SG-FD method for Eq. (46) for d = 2, 3, 4, 6. The spatial resolutions are $N_{\mathbf{x}}$ = 8, 16, 32, 64, 128 for d = 2, 3, 4 and $N_{\mathbf{x}}$ = 16, 32, 64, 128 for d = 6, and the time step is $\Delta t = 1/N_{\mathbf{x}}$. The green lines with blue markers denote the solution errors at final time, and the red one is a reference line with a slope of 1.

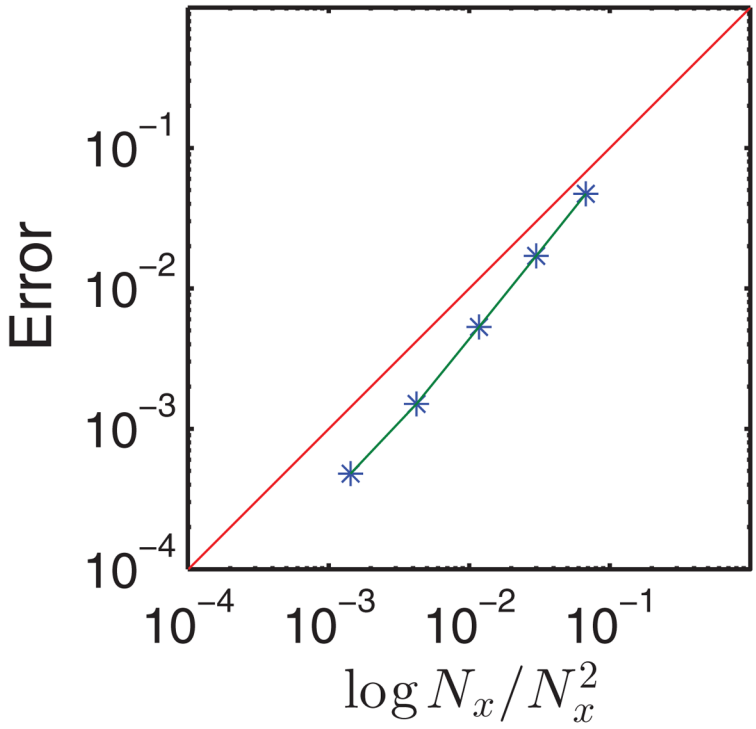

Figure 4.

The log-log plot of L^∞ error versus the quantity $(\log N_{\mathbf{x}})/N_{\mathbf{x}}^2$ for the AcIIF2 method with the sparse grid combination technique for Eq. (49), which contains cross derivatives. The spatial resolutions are $N_{\mathbf{x}} = 8, 16, 32, 64, 128$, the time step is $\Delta t = 1/N_{\mathbf{x}}$, and all simulations end at $t = 0.5$. The blue markers denote numerical errors, and the red line is a straight line with a slope of 1.

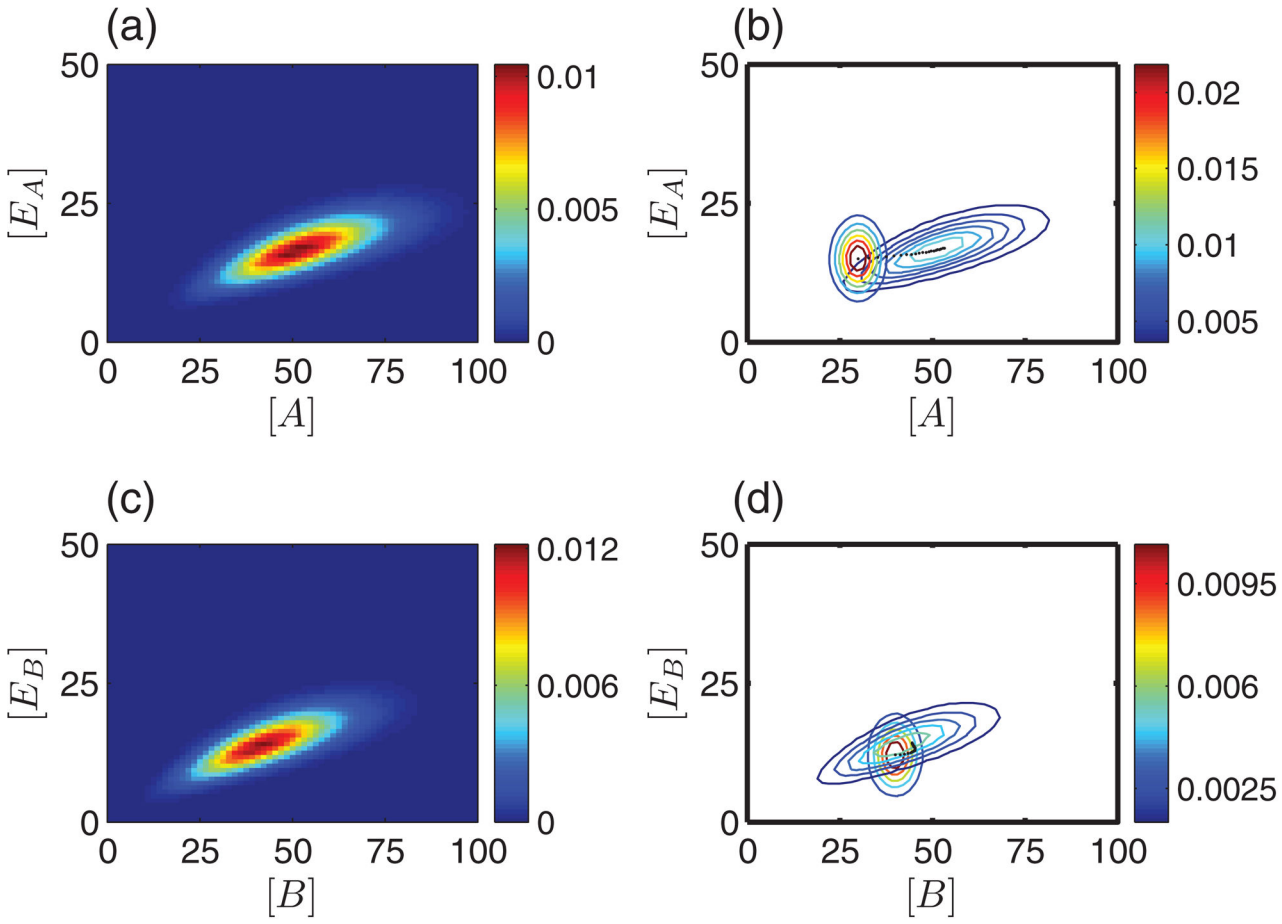

Figure 5.

Numerical solution of the biochemical system (54) using AcIIF2 with a sparse grid combination technique. The final time is $t = 45s$, and the spatial resolution is $N_{\mathbf{x}} = 2^{15}$. (a) The distribution of metabolite A and related enzyme E_A at $t = 45$. (b) The contour plot of the initial distribution and final distribution of A and E_A. The black dotted line is the trace of the center by solving the corresponding ODE system. (c) The distribution of metabolite B and related enzyme E_B at $t = 45$. (d) The contour plot of the initial distribution and final distribution of B and E_B. The black dotted line is the trace of the center by solving the corresponding ODE system.

Table 1

The CPU time (in seconds) of the AcIIF2-SG-FD, IIF2-SG-FD and RK2-SG-FD methods for different spatial mesh sizes $N_x = 2^3, 2^4, 2^5, 2^6, 2^7$ for $d = 2, 3, 4$. For AcIIF2-SG-FD and IIF2-SG-FD, the time step is $\Delta t = 1/N_{\mathbf{x}}$. For RK2-SG-FD, the time step is $\Delta t = 1/N_{\mathbf{x}}^2$. All data are shown for the solutions calculated at $t = 0.5$.

	d=2			d=3			d=4		
N_x	AcIIF2	IIF2	RK2	AcIIF2	IIF2	RK2	AcIIF2	IIF2	RK2
8	0.016	0.025	0.004	0.044	0.011	0.009	0.122	0.042	0.022
16	0.04	0.018	0.047	0.206	0.162	0.198	0.85	1.44	0.66
32	0.17	0.11	0.60	1.21	3.19	4.01	7.19	46.5	18.5
64	0.65	1.17	11.0	8.1	64.5	84.3	85.9	1268.9	509.3
128	3.47	11.46	153.02	68.5	902.2	1846.3	1910.6	25729.2	12068.8

Table 2

The CPU time (in seconds) for AcIIF2-SG-FD and RK2-SG-FD for different spatial mesh sizes $N_{\mathbf{x}} = 2^3, 2^4, 2^5, 2^6, 2^7$ for $d = 2, 3, 4$ respectively. For AcIIF2-SG-FD, the time step is $\Delta t = 1/N_{\mathbf{x}}$. In RK2-SG-FD, the time step is $\Delta t = 1/N_{\mathbf{x}}^2$. All simulations end at $t = 0.5$.

	d=2		d=3		d=4	
N_x	AcIIF2-SG-FD	RK2-SG-FD	AcIIF2-SG-FD	RK2-SG-FD	AcIIF2-SG-FD	RK2-SG-FD
8	0.011	0.004	0.06	0.01	0.146	0.023
16	0.047	0.045	0.232	0.202	0.94	0.68
32	0.196	0.63	1.42	4.17	8.46	19.1
64	0.92	12.19	10.7	83.8	106.2	533.6
128	5.53	166.14	99.1	2016.8	2163.7	12123.7

Table 3

The CPU time for computing all matrix exponentials for both the AcIIF and IIF methods. The time step is $\Delta t = 1/\max_i N_{x_i}$ to meet the error requirement described by Eq. (35) for the sparse grid combination technique.

(d, K)	$(N_{x_1}, \ldots, N_{x_d})$	CPU time for AcIIF on matrix exponentials	CPU time for IIF on matrix exponential
(3, 11)	$(2^2, 2^2, 2^7)$	0.3s	3.5s
(3, 11)	$(2^3, 2^4, 2^4)$	0.2s	5.4s
(3, 12)	$(2^2, 2^2, 2^8)$	4.3s	51s
(3, 12)	$(2^4, 2^4, 2^4)$	0.4s	24s
(4, 11)	$(2^2, 2^2, 2^2, 2^5)$	0.05s	5.6s
(4, 12)	$(2^2, 2^2, 2^2, 2^6)$	0.12s	68s