

Krylov implicit integration factor methods for spatial discretization on high dimensional unstructured meshes: application to discontinuous Galerkin methods

Shanqin Chen¹ and Yong-Tao Zhang^{2*}

*Corresponding author. Tel.: +1 574 631 6079. E-mail: yzhang10@nd.edu

Abstract

Integration factor methods are a class of “exactly linear part” time discretization methods. In [Q. Nie, Y.-T. Zhang, and R. Zhao, Efficient semi-implicit schemes for stiff systems. *Journal of Computational Physics*, 2006. v214: 521-537], a class of efficient implicit integration factor (IIF) methods are developed for solving systems with both stiff linear and nonlinear terms, arising from numerical spatial discretization of time-dependent partial differential equations (PDEs) with linear high order terms and stiff lower order nonlinear terms. A novel property of the scheme is that the exact evaluation of the linear part is decoupled from the implicit treatment of the nonlinear part. As a result, the size of the nonlinear system arising from the implicit treatment is independent of the number of spatial grid points, when the scheme is used for solving reaction-diffusion PDEs. The tremendous challenge in applying IIF temporal discretization for time-dependent PDEs on high spatial dimensions is how to evaluate the matrix exponential operator efficiently. On rectangular meshes, the compact IIF methods [Q. Nie, F. Wan, Y.-T. Zhang, and X.-F. Liu, Compact integration factor methods in high spatial dimensions. *Journal of Computational Physics*, 2008. v227: 5238-5255] were developed to deal with this issue. For spatial discretization on unstructured meshes to solve PDEs on complex geometrical domains, how to efficiently apply the IIF temporal discretization was open. In this paper, we solved this problem by applying the Krylov subspace approximations to the matrix exponential operator. Then we apply this novel time discretization technique to discontinuous Galerkin (DG) methods on unstructured meshes for solving reaction-diffusion equations. Numerical examples are shown to demonstrate the accuracy, efficiency and robustness of the method in resolving the stiffness of the DG spatial operator for reaction-diffusion PDEs. Application of the method to a mathematical model in pattern formation during zebrafish embryo development shall be shown.

Key Words: Discontinuous Galerkin finite element methods · Integration factor methods · Krylov subspace approximation · Triangular meshes · Morphogenesis

AMS(MOS) subject classification: 65M99

¹Department of Mathematical Sciences, Indiana University South Bend, South Bend, IN 46634-7111, USA. E-mail: chen39@iusb.edu. Research supported by IU South Bend Research and Development grant.

²Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556, USA. E-mail: yzhang10@nd.edu. Research partially supported by NSF grant DMS-0810413 and Oak Ridge Associated Universities (ORAU) Ralph E. Powe Junior Faculty Enhancement Award.

1 Introduction

High order accuracy numerical methods (e.g. weighted essentially non-oscillatory (WENO) methods, discontinuous Galerkin (DG) methods, spectral methods, etc) are especially efficient for numerically solving problems which contain complex solution structures, and have been applied extensively in computational fluid dynamics. Efficient and high order temporal numerical schemes are important for the performance of high accuracy numerical simulations. A lot of state-of-the-art high order time-stepping methods were developed. Here we just give a few examples and it is *not* a complete list. For example, the total variation diminishing (TVD) Runge-Kutta (RK) schemes [76, 77, 40, 41]; high order implicit-explicit (IMEX) multistep / Runge-Kutta methods and their applications [4, 5, 48, 51, 64, 84, 73, 93]; spectral deferred correction (SDC) methods [14, 34, 47, 56, 57, 61]; hybrid methods of SDC and high order RK schemes [19]; etc.

Integration factor (IF) methods are a broad class of time discretization methods which have been combined with spatial discretization methods especially spectral methods [35, 13, 16, 80] for solving various partial differential equations (PDEs). Especially many important mathematical models in fluid dynamics or biological problems involve nonlinear PDEs with the linear highest spatial derivatives. The method of lines generates stiff systems of ordinary differential equations (ODEs) with the stiff linear terms (nonlinear terms could also be stiff, depending on different problems). Integration factor methods are a class of “exactly linear part” time discretization methods for the solution of this type of stiff systems. This class of methods perform the time evolution of the stiff linear operator via evaluation of an exponential function of the corresponding matrix. Hence for PDEs with linear high order derivatives, the integration factor type time discretization can remove both the stability constrain and numerical errors from the high order derivatives.

Traditional integration factor methods have a disadvantage that large error coefficients are produced when the linear term has large norm [13]. Recently S. Krogstad developed a method which can improve the accuracy of the traditional integration factor methods significantly [53]. Another class of “integration factor” type methods, the exponential time differencing (ETD) methods [10, 11, 29, 50], can also obtain a better accuracy than the traditional integration factor methods. For ETD methods, extra treatments are needed for consistent order of accuracy [50, 30].

The explicit integration factor methods use explicit linear multi-step methods or explicit Runge-Kutta methods to treat the nonlinear terms in the system (e.g. the explicit ETD-RK schemes [29]), which are very efficient for systems with stiff linear part but nonstiff or mildly stiff nonlinear part. In [45], explicit exponential Runge-Kutta methods for the time integration of semilinear parabolic problems were analyzed. For systems with both stiff linear and nonlinear terms, an implicit treatment of nonlinear terms is desirable. In [66], we developed a class of efficient implicit integration factor (IIF) methods for solving systems with both stiff linear and nonlinear terms. A novel property of the methods is that the implicit terms are free of the exponential operation of the linear terms. Hence when the methods are applied to PDEs with stiff nonlinear reactions (e.g. the reaction-diffusion systems arising from mathematical models in computational developmental biology), the exact evaluation of the linear part is decoupled from the implicit treatment of the nonlinear reaction terms. As a result, the size of the nonlinear system arising from the implicit treatment is independent of the

number of spatial grid points; it only depends on the number of the original equations. This distinguishes our IIF methods [66] from implicit ETD methods in [11]. Another efficient approach to decouple the stiff nonlinear terms from the linear terms is to use operator splitting integration factor methods (e.g. [60]). We would like to point out that our approach is different from the operator splitting methods, hence is free of operator splitting errors.

The tremendous difficulty in implementing integration factor type method for high dimensional problems is how to efficiently evaluate the product of the matrix exponential with a vector. The differential matrix from the high dimensional spatial discretization is usually very large and sparse, but its exponential matrix will be dense. CPU cost and storage of such exponential matrix are prohibitive for high dimensional problems, although it can be handled well for 1D problems [66]. On rectangular meshes, the compact integration factor methods [67] can be used to deal with this issue. By introducing a compact representation for the matrix approximating the differential operator, the compact IIF methods apply matrix exponential operations sequentially in every spatial direction. As the results, exponential matrices which are calculated and stored have small sizes, as those in the 1D problem. For example for a 3D problem, if we have N grid points in every direction of the x, y and z directions, then we do *not* need to work on the full $N^3 \times N^3$ matrix and its exponential as in the original non-compact IIF methods [66], but just need to consider several $N \times N$ matrices and their exponentials. The storage and CPU time required for compact IIF schemes are smaller by orders of magnitude than the non-compact ones. The operation count of compact integration factor schemes is $O(N^4)$ vs. $O(N^6)$ for non-compact ones. See more details in [67]. But how to apply this approach to spatial discretization on high dimensional unstructured meshes for dealing with complex domain geometry is still unclear. In this paper, we address this problem by using the Krylov subspace approximations to the matrix exponential operator and develop the Krylov implicit integration factor (Krylov IIF) methods for spatial discretization on triangular meshes.

The Krylov subspace methods were used and analyzed by Gallopoulos and Saad [36], Saad [74] for the approximation of a product of a exponential matrix of a large sparse differential matrix and a given vector, in the finite difference discretization of a time-dependent diffusion equation. Since in many applications including the integration factor methods, one does not need the full exponential matrix, but only the matrix-vector product. Such Krylov approximations were first used in chemical physics [65, 69], and are also related to early work e.g. [15, 31, 38, 79]. The methods were studied further by Hochbruck and Lubich [43], Hochbruck, Lubich, and Selhofer [44], van Den Eshof and Hochbruck [82], Druskin and Knizhnerman [32], Druskin, Greenbaum, and Knizhnerman [33], Stewart and Leyk [78], Moret and Novati [63]. See also the review paper of Moler and Van Loan [62]. As discovered in [43], the convergence to $e^{A\Delta t}v$ is substantially faster than that of corresponding Krylov methods for the solution of linear system $(I - \Delta tA)x = v$, which arises in the implicit treatment of diffusion or other high order derivative terms, at least unless a good preconditional is available.

On the discretization methods for the spatial directions, discontinuous Galerkin (DG) finite element methods are a class of popular high order accuracy spatial discretization methods for numerically solving various partial differential equations (PDEs) arising from computational fluid dynamics and other computational physics problems. DG methods confer several advantages that make them attractive for applications.

These include common advantages shared by all finite element methods such as their ability for easy handling of complicated geometry and boundary conditions. Since DG methods use completely discontinuous piecewise polynomial space for the numerical solution and the test functions, this property makes DG methods have lots of flexibility which is not shared by continuous Galerkin finite element methods, such as their flexibility for easy h-p adaptivity including changes of approximation orders between neighboring elements and allowing general meshes with hanging nodes, their compactness hence efficient parallel implementation [12], and their easy coordination with finite volume techniques for computing problems with discontinuous or sharp gradient solutions. The first DG method was introduced by Reed and Hill [70], in the framework of neutron transport. A major development of the DG method was carried out by Cockburn et al. in a series of papers [23, 24, 21, 25, 26], in which they established a framework to easily solve nonlinear time dependent hyperbolic conservation laws.

The DG method has found rapid applications in many diverse areas. Good references for the DG method and its recent development include the survey paper [22], other papers in that Springer volume, and the review papers [20, 27]. In recent years, DG methods have been generalized to solve time dependent PDEs containing higher spatial derivatives. It was designed to solve a convection diffusion equation (containing second derivatives) by Cockburn and Shu [28], motivated by the successful numerical experiments of Bassi and Rebay [8] for the compressible Navier-Stokes equations. This method is termed local discontinuous Galerkin (LDG) method because the auxiliary variables introduced to approximate spatial derivatives can be eliminated locally. For more references on DG methods for diffusion problems, see for example [2, 3, 6, 7, 9, 17, 39, 46, 49, 52, 59, 68, 71, 72]. Later, LDG methods were developed to solve various nonlinear time dependent PDEs with higher order derivatives in [58, 86, 87, 88, 89, 90, 91]. In a recent paper [18], Cheng and Shu developed a new DG method for solving time dependent PDEs with higher order spatial derivatives, based on [1, 37, 83]. The scheme is formulated by repeated integration by parts of the original equation and then replacing the interface values of the solution by carefully chosen numerical fluxes. Compared to the LDG method, this new DG method can be applied without introducing any auxiliary variables or rewriting the original equation into a larger system, hence it is easier to formulate and implement, has a smaller effective stencil, and may save storage and computational cost. On the other hand, the disadvantages of this new DG method against the LDG method include the fact that the new DG method is in general inconsistent for lower order P^k elements when $k + 1$ is smaller than the highest order of the spatial derivative in the PDE; and the fact that extra penalty terms must be added to the numerical flux to ensure stability for even order derivative terms [18].

When DG methods are applied to discretize the spatial variables for convection-diffusion problems or other PDEs which have high order derivatives, a large coupled stiff ordinary differential equations (ODEs) system is generated. An explicit time discretization for the resulting stiff ODE system will suffer from extremely small time step restriction for stability. In [85], time discretization techniques including semi-implicit spectral deferred correction method, the additive Runge-Kutta method and the exponential time differencing (ETD) method for solving the stiff ODEs resulting from a LDG spatial discretization to PDEs with high order spatial derivatives on 1D spatial domain were compared and studied.

To test our new Krylov IIF schemes in this paper, we apply them to solve the stiff ODE system resulting from a DG spatial discretization for reaction-diffusion problems on 2D triangular meshes. This DG spatial discretization is based on the DG methods for PDEs with high order derivatives by Cheng and Shu [18]. In [94], we extended the methods in [18] to solve reaction-diffusion problems on 2D triangular meshes. Due to the stiffness of the spatial operators, the implicit discretization was formulated for the P^1 case in [94] and an operator-splitting approach was used to enhance the computation efficiency. In this paper, we will formulate the implicit DG discretization for both P^1 and P^2 cases, and apply the new Krylov IIF schemes for the time-marching. Numerical examples are shown to demonstrate the accuracy, efficiency and robustness of the method in resolving the stiffness of the DG spatial operator for reaction-diffusion PDEs which have higher than first order spatial derivatives. Application of the method to a mathematical model in spatial pattern formation during zebrafish embryo development shall be shown.

The rest of the paper is organized as following. In Section 2, we derive and formulate the Krylov IIF methods. In Section 3, we describe in detail the P^1 and P^2 DG spatial discretization for reaction-diffusion equations on 2D triangular meshes; then apply the new Krylov IIF methods to resulting stiff ODE system. Numerical experiments to test the algorithms are presented in Section 4. Discussions and conclusions are given in Section 5.

2 Implicit integration factor methods based on Krylov subspace approximation

We first review the original IIF methods developed in [66]. Then we propose the new IIF methods based on Krylov subspace approximations to the Matrix exponential operator.

2.1 Review of the original IIF methods

Assume that we need to solve a system of ODEs which arises after a spatial discretization of a PDE system whose highest order term is linear and has higher than first order derivative (e.g. a reaction-diffusion equation):

$$\frac{dU(t)}{dt} = \mathbf{A}U(t) + \mathbf{F}(U(t)), \quad (1)$$

where $U(t) \in R^N$, $\mathbf{A} \in R^{N \times N}$ is the constant approximation matrix for the linear differential operator of the highest order derivative, $\mathbf{F}(U(t)) \in R^N$ is the nonlinear term. The matrix \mathbf{A} is usually a sparse matrix when a finite difference or finite element method is used for the spatial discretization.

To construct the IIF methods for (1), we multiply it by the integration factor $e^{-\mathbf{A}t}$ and integrate over one time step from t_n to $t_{n+1} \equiv t_n + \Delta t$ to obtain

$$U(t_{n+1}) = e^{\mathbf{A}\Delta t}U(t_n) + e^{\mathbf{A}\Delta t} \int_0^{\Delta t} e^{-\mathbf{A}\tau} \mathbf{F}(U(t_n + \tau))d\tau. \quad (2)$$

Then we approximate the integrand in (2) using an $r-1$ th order Lagrange interpolation polynomial with interpolation points at $t_{n+1}, t_n, \dots, t_{n+2-r}$, and obtain the r th order IIF scheme

$$U_{n+1} = e^{\mathbf{A}\Delta t}U_n + \Delta t \left(\alpha_1 \mathbf{F}(U_{n+1}) + \sum_{i=0}^{r-2} \alpha_{-i} e^{(i+1)\mathbf{A}\Delta t} \mathbf{F}(U_{n-i}) \right), \quad (3)$$

where

$$\alpha_{-i} = \frac{1}{\Delta t} \int_0^{\Delta t} \prod_{\substack{k=-1 \\ k \neq i}}^{r-2} \frac{\tau + k\Delta t}{(k-i)\Delta t} d\tau, \quad -1 \leq i \leq r-2. \quad (4)$$

See [66, 67] for the values of coefficients α_j for the schemes with different orders. For example, the second order scheme is of the following form

$$U_{n+1} = e^{\mathbf{A}\Delta t} \left(U_n + \frac{\Delta t}{2} \mathbf{F}(U_n) \right) + \frac{\Delta t}{2} \mathbf{F}(U_{n+1}), \quad (5)$$

and the third order scheme is

$$U_{n+1} = e^{\mathbf{A}\Delta t}U_n + \Delta t \left(\frac{5}{12} \mathbf{F}(U_{n+1}) + \frac{2}{3} e^{\mathbf{A}\Delta t} \mathbf{F}(U_n) - \frac{1}{12} e^{2\mathbf{A}\Delta t} \mathbf{F}(U_{n-1}) \right). \quad (6)$$

The distinct feature of the scheme (3) is that the nonlinear implicit term $\mathbf{F}(U_{n+1})$ does *not* involve the matrix exponential operator, unlike the implicit ETD schemes [11]. Hence if the nonlinear operator \mathbf{F} itself does *not* involve the coupling of the numerical values at the spatial grid points (e.g. the nonlinear reactions), the size of nonlinear system resulted from the implicit treatment is independent of the number of the spatial grid points, and it only depends on the number of the original PDEs.

Remark: we consider the time step size Δt to be uniform in the time evolution in this paper. The methods to efficiently deal with non-uniform time step sizes (e.g., adaptive time step computation; solving PDEs which involves hyperbolic terms and needs the CFL condition constraint) will be reported in the future.

2.2 IIF methods based on Krylov Subspace Approximation

The efficiency of the IIF schemes (3) largely depends on the methods to evaluate the product of the matrix exponential with a vector, for example $e^{\mathbf{A}\Delta t}U_n$. For PDEs defined on high spatial dimensions (two spatial dimensions (2D) and above), the method of lines (MOL) with a specific spatial discretization will generate a large and sparse matrix \mathbf{A} in (1). For example if a finite difference method is used on a 2D rectangular mesh with N grid points in both x and y directions, then the matrix \mathbf{A} has size $N^2 \times N^2$. And the exponential matrix $e^{\mathbf{A}\Delta t}$ is dense. In [66], for a given spatial and temporal numerical resolution, the exponential matrices such as $e^{\mathbf{A}\Delta t}$ are pre-computed and stored for later use at every time step. For one-dimensional problems, this works very well since the size of the matrix \mathbf{A} is manageable. For 2D and 3D problems, direct computation and storage of such exponential matrix are prohibitive in terms of both CPU cost and computer memory.

On rectangular meshes, the compact IIF methods [67] can be used to deal with this issue for some differential operators (e.g. the Laplacian operator). The compact IIF methods introduce a compact representation for the matrix approximating the differential operator. The compact form involves storage only proportional to the number of unknowns, i.e. the size of U , unlike the non-compact approach, which is proportional to the square of the unknowns. As the results, exponential matrices which are calculated and stored have small sizes, as those in the 1D problem. For example, for a 3D problem if we have N_x, N_y and N_z grid points in x, y and z directions, then instead of considering the full $(N_x N_y N_z) \times (N_x N_y N_z)$ matrix and its exponential, we just consider $N_x \times N_x$, $N_y \times N_y$ and $N_z \times N_z$ matrices and their exponentials. The storage and CPU time required for compact IIF schemes are smaller by orders of magnitude than the original non-compact ones in [66]. The operation count of compact IIF schemes is $O(N_x^2 N_y N_z + N_x N_y^2 N_z + N_x N_y N_z^2)$ vs. $O(N_x^2 N_y^2 N_z^2)$ for non-compact IIF schemes.

For spatial discretization on high dimensional unstructured meshes (e.g. triangular meshes), how to formulate a compact matrix representation of the spatial operations and apply this approach is still unclear. Moreover, application of compact IIF methods to more general spatial operators (e.g. the general diffusion operators involving mixed derivatives) is not straightforward. So we re-consider the non-compact IIF methods (3) and look for another way to efficiently compute the matrix exponential operations.

Although the matrix \mathbf{A} is sparse for many spatial discretizations, the exponential matrix $e^{\mathbf{A}\Delta t}$ is dense. Directly computing and storing $e^{\mathbf{A}\Delta t}$ for spatial discretization on high dimensional unstructured meshes is not practical for a typical machine. Fortunately we do *not* need the full exponential matrix $e^{\mathbf{A}\Delta t}$ itself, but only the products of the exponential matrix with some vectors in (3). The Krylov subspace approximations to the matrix exponential operator is an excellent choice in terms of both accuracy and efficiency. For example, in [36, 74], the Krylov subspace methods were used for the approximation of $e^{\mathbf{A}\Delta t}v$ where \mathbf{A} is a large sparse matrix and v is a given vector, in the finite difference discretization of a time-dependent diffusion equation. In the following we apply the Krylov subspace method to approximate the products of the exponential matrix with vectors in our IIF schemes (3) and derive the new Krylov IIF methods. First we describe the Krylov subspace methods to approximate $e^{\mathbf{A}\Delta t}v$, following the literature (e.g. [36, 62]).

The large sparse matrix \mathbf{A} is projected to the Krylov subspace

$$K_M = \text{span}\{v, \mathbf{A}v, \mathbf{A}^2v, \dots, \mathbf{A}^{M-1}v\}. \quad (7)$$

The dimension M of the Krylov subspace is **much** smaller than the dimension N of the large sparse matrix \mathbf{A} . In all numerical computations of this paper, we take $M = 25$ for different N , and accurate results are obtained as shown in Section 4. An orthonormal basis $V_M = [v_1, v_2, v_3, \dots, v_M]$ of the Krylov subspace K_M is generated by the well-known Arnoldi algorithm [81] as the following.

1. Compute the initial vector: $v_1 = v/\|v\|_2$.
2. Perform iterations: Do $j = 1, 2, \dots, M$:
 - 1) Compute the vector $w = \mathbf{A}v_j$.
 - 2) Do $i = 1, 2, \dots, j$:
 - (a) Compute the inner product $h_{i,j} = (w, v_i)$.
 - (b) Compute the vector $w = w - h_{i,j}v_i$.

- 3) Compute $h_{j+1,j} = \|w\|_2$.
- 4) If $h_{j+1,j} \equiv 0$, then
 - stop the iteration;
 - else
 - compute the next basis vector $v_{j+1} = w/h_{j+1,j}$.

In the Arnoldi algorithm, if $h_{j+1,j} \equiv 0$ for some $j < M$, it means that the convergence has occurred and the Krylov subspace $K_M = \text{span}\{v_1, v_2, \dots, v_j\}$, so the iteration can be stopped at this step j , and we assign the value of this j to M . This algorithm will produce an orthonormal basis V_M of the Krylov subspace K_M . Denote the $M \times M$ upper Hessenberg matrix consisting of the coefficients $h_{i,j}$ by H_M . Since the columns of V_M are orthogonal, we have

$$H_M = V_M^T \mathbf{A} V_M. \quad (8)$$

This means that the very small Hessenberg matrix H_M represents the projection of the large sparse matrix \mathbf{A} to the Krylov subspace K_M , with respect to the basis V_M . Also since V_M is orthonormal, the vector $V_M V_M^T e^{\mathbf{A}\Delta t} v$ is the orthogonal projection of $e^{\mathbf{A}\Delta t} v$ on the Krylov subspace K_M , namely, it is the closest approximation to $e^{\mathbf{A}\Delta t} v$ from K_M . Therefore

$$e^{\mathbf{A}\Delta t} v \simeq V_M V_M^T e^{\mathbf{A}\Delta t} v = \beta V_M V_M^T e^{\mathbf{A}\Delta t} v_1 = \beta V_M V_M^T e^{\mathbf{A}\Delta t} V_M e_1,$$

where $\beta = \|v\|_2$, and e_1 denotes the first column of the $M \times M$ identity matrix I_M . Use the fact of (8), we have the approximation

$$e^{\mathbf{A}\Delta t} v \simeq \beta V_M e^{H_M \Delta t} e_1. \quad (9)$$

Thus the large $e^{\mathbf{A}\Delta t}$ matrix exponential problem is replaced with a much smaller $e^{H_M \Delta t}$ problem. The small matrix exponential $e^{H_M \Delta t}$ will be computed using a scaling and squaring algorithm with a Padé approximation with only computational cost of $O(M^2)$, see [42, 62, 36]. Applying the Krylov subspace approximation (9) to (3), we obtain the Krylov IIF schemes

$$U_{n+1} = \Delta t \alpha_1 \mathbf{F}(U_{n+1}) + \beta_{0,n} V_{M,0,n} e^{H_{M,0,n} \Delta t} e_1 + \Delta t \left(\sum_{i=1}^{r-2} \alpha_{-i} \beta_{i,n} V_{M,i,n} e^{(i+1)H_{M,i,n} \Delta t} e_1 \right), \quad (10)$$

where $\beta_{0,n} = \|U_n + \alpha_0 \Delta t \mathbf{F}(U_n)\|_2$, $\beta_{i,n} = \|\mathbf{F}(U_{n-i})\|_2$, $V_{M,0,n}$ and $H_{M,0,n}$ are orthonormal basis and upper Hessenberg matrix generated by the Arnoldi algorithm with the initial vector $U_n + \alpha_0 \Delta t \mathbf{F}(U_n)$. $V_{M,i,n}$ and $H_{M,i,n}$ are orthonormal basis and upper Hessenberg matrix generated by the Arnoldi algorithm with the initial vectors $\mathbf{F}(U_{n-i})$, for $i = 1, 2, \dots, r-2$. We would like to point out that $V_{M,0,n}$ and $V_{M,i,n}$, $i = 1, 2, \dots, r-2$ are orthonormal bases of different Krylov subspaces for the same matrix \mathbf{A} , which are generated with different initial vectors in the Arnoldi algorithm. The value of M is taken to be large enough such that the error of Krylov subspace approximations is much less than the truncation errors of the numerical schemes (3). From our numerical experiments in this paper (Section 4), we can see that our numerical schemes have already given a clear accuracy order with a very small size $M = 25$, and M does *not* need to be increased when the spatial-temporal resolution is refined. Specifically the

second order Krylov IIF scheme has the form

$$U_{n+1} = \frac{1}{2}\Delta t \mathbf{F}(U_{n+1}) + \beta_n V_{M,n} e^{H_{M,n}\Delta t} e_1, \quad (11)$$

where $\beta_n = \|U_n + \frac{1}{2}\Delta t \mathbf{F}(U_n)\|_2$, $V_{M,n}$ and $H_{M,n}$ are orthonormal basis and upper Hessenberg matrix generated by the Arnoldi algorithm with the initial vector $U_n + \frac{1}{2}\Delta t \mathbf{F}(U_n)$. And the third order Krylov IIF scheme has the form

$$U_{n+1} = \frac{5}{12}\Delta t \mathbf{F}(U_{n+1}) + \beta_{0,n} V_{M,0,n} e^{H_{M,0,n}\Delta t} e_1 - \frac{1}{12}\Delta t \beta_{1,n} V_{M,1,n} e^{2H_{M,1,n}\Delta t} e_1, \quad (12)$$

where $\beta_{0,n} = \|U_n + \frac{2}{3}\Delta t \mathbf{F}(U_n)\|_2$, $\beta_{1,n} = \|\mathbf{F}(U_{n-1})\|_2$, $V_{M,0,n}$ and $H_{M,0,n}$ are orthonormal basis and upper Hessenberg matrix generated by the Arnoldi algorithm with the initial vector $U_n + \frac{2}{3}\Delta t \mathbf{F}(U_n)$. $V_{M,1,n}$ and $H_{M,1,n}$ are orthonormal basis and upper Hessenberg matrix generated by the Arnoldi algorithm with the initial vector $\mathbf{F}(U_{n-1})$.

3 Implicit DG formulations and application of Krylov IIF methods

To test our Krylov IIF methods, we apply them to solve the stiff ODE system arising from a DG discretization of reaction-diffusion equations on 2D triangular meshes. Let Ω be an open, bounded domain on which the reaction-diffusion system

$$\frac{\partial \mathbf{u}}{\partial t} = D\Delta \mathbf{u} + F(\mathbf{u}) \quad (13)$$

is defined, where \mathbf{u} often represents concentrations of a group of chemical molecules, D is the constant diffusion matrix, $\Delta \mathbf{u}$ is the Laplacian associated with the diffusion of the molecules \mathbf{u} , and $F(\mathbf{u})$ describes the nonlinear chemical reactions. For the simplicity of the description, we consider the scalar case of (13). The numerical formulae for the scalar case can be straightforwardly extended to solve the system case component by component. We consider a triangulation Ω_h of Ω which consists of nonoverlapping triangles $\{\Delta_m\}_{m=1}^N$. Let $h_{\min} = \min_{1 \leq m \leq N} \rho_m$, where ρ_m is the diameter of the inscribed circle of the triangle Δ_m .

Define the finite element space $V_h^k = \{v : v|_{\Delta_m} \in P^k(\Delta_m), m = 1, \dots, N\}$, where $P^k(\Delta_m)$ denotes the set of all polynomials of degree at most k on Δ_m .

We apply the DG formulation [18] to discretize the reaction-diffusion equations (13) in the spatial direction, but keep the time variable continuous. The semi-discrete scheme is: find $u \in V_h^k$, such that

$$\begin{aligned} \int_{\Delta_m} u_t v dx - D \int_{\Delta_m} u \nabla^2 v dx + D \int_{\partial \Delta_m} \hat{u} \nabla v \cdot \vec{n}_{\partial \Delta_m} dS - D \int_{\partial \Delta_m} v \widetilde{\nabla} u \cdot \vec{n}_{\partial \Delta_m} dS \\ = \int_{\Delta_m} F(u) v dx \end{aligned} \quad (14)$$

holds true for any $v \in V_h^k$ and $m = 1, \dots, N$. The numerical fluxes on the element edges $\partial \Delta_m$ are chosen as

$$\hat{u} = \frac{u^{in} + u^{ext}}{2}, \quad (15)$$

$$\widetilde{\nabla}u = \frac{(\nabla u)^{in} + (\nabla u)^{ext}}{2} + \beta[u], \quad (16)$$

where the jump term

$$[u] = (u^{ext} - u^{in})|_{\partial\Delta_m} \cdot \vec{n}_{\partial\Delta_m}, \quad (17)$$

u^{in} and u^{ext} are the limits of u at $x \in \partial\Delta_m$ taken from the interior and the exterior of Δ_m respectively, $\vec{n}_{\partial\Delta_m}$ is the outward unit normal to the element Δ_m at $x \in \partial\Delta_m$, and β is a positive constant that is of the order $\mathcal{O}(h_{\min}^{-1})$. In all of computations of this paper, we take $\beta = 10/h_{\min}$. The choice of numerical fluxes (15)-(17) is crucial for the stability and convergence of the DG scheme (14). See [27, 18] for more discussions about the choice of numerical fluxes.

To apply the Krylov IIF schemes (10) to the DG spatial discretization (14), we will need the matrix expression (i.e., the implicit DG formulae) as (1). This is a different step from the explicit DG methods for hyperbolic conservation laws. The implicit DG formulae for the P^1 case has been given in [94]. In this paper, we will present the implicit DG formulae for both the P^1 and P^2 cases. For each element Δ_m , denote its three neighboring elements by i_m , j_m , and k_m . To simplify notations in the following presentation, we will omit the subscript m and just use i , j , k to represent the neighboring cells of Δ_m .

The implicit DG formulae for the P^1 case has been derived in [94]. But to make the paper self-contained, we will re-present the P^1 case in the section 3.1. The new formulae for the P^2 case will be given in the section 3.2.

3.1 The P^1 case

The linear polynomial on Δ_m is represented by

$$u(x, y, t) = a_m(t) + b_m(t)\xi_m + c_m(t)\eta_m \quad (18)$$

where

$$\xi_m = \frac{x - x_m}{h_m}, \quad (19)$$

$$\eta_m = \frac{y - y_m}{h_m}, \quad (20)$$

and (x_m, y_m) is the barycenter of the element Δ_m , $h_m = \sqrt{|\Delta_m|}$ with $|\Delta_m|$ denoting the area of the triangle Δ_m . By taking $v = 1, \xi_m, \eta_m$ on Δ_m and $v = 0$ elsewhere, the DG formulation (14) can be converted from the integral form to the following

semi-discretized ODE system, for $m = 1, \dots, N$:

$$q_{11}a'_m(t) + q_{12}b'_m(t) + q_{13}c'_m(t) = D \sum_{l=m,i,j,k} [w_{al1}a_l(t) + w_{bl1}b_l(t) + w_{cl1}c_l(t)] \\ + (q_{11}/3) \sum_{l=i,j,k} F(u(x_{m,l}, y_{m,l})), \quad (21)$$

$$q_{21}a'_m(t) + q_{22}b'_m(t) + q_{23}c'_m(t) = D \sum_{l=m,i,j,k} [w_{al2}a_l(t) + w_{bl2}b_l(t) + w_{cl2}c_l(t)] \\ + (q_{11}/3) \sum_{l=i,j,k} F(u(x_{m,l}, y_{m,l}))\xi_m(x_{m,l}, y_{m,l}), \quad (22)$$

$$q_{31}a'_m(t) + q_{32}b'_m(t) + q_{33}c'_m(t) = D \sum_{l=m,i,j,k} [w_{al3}a_l(t) + w_{bl3}b_l(t) + w_{cl3}c_l(t)] \\ + (q_{11}/3) \sum_{l=i,j,k} F(u(x_{m,l}, y_{m,l}))\eta_m(x_{m,l}, y_{m,l}), \quad (23)$$

where the coefficients $\{q_{rs}\}_{r,s=1}^3$, $\{\{w_{alr}\}_{r=1}^3, \{w_{blr}\}_{r=1}^3, \{w_{clr}\}_{r=1}^3\}_{l=m,i,j,k}$ are constants which depend on the local geometry of the mesh (i.e., triangle Δ_m and its neighboring cells i, j, k and $\vec{n}_{\partial\Delta_m}$), the local basis functions $1, \{\xi_l, \eta_l\}_{l=m,i,j,k}$, and the constant β . $\{(x_{m,l}, y_{m,l})\}_{l=i,j,k}$ are the mid-points of the three edges $\{e_l\}_{l=i,j,k}$ of Δ_m which serve as Gaussian quadrature points for the P^1 case in the integral involving the nonlinear reaction terms in (14). The detailed formulae for computing these constants are presented in the Appendix. In our implementation, these mesh-dependent constants are pre-calculated and stored before the time evolution since they don't depend on the numerical solution u . Rewrite equations (21)-(23) to the matrix-vector form

$$\mathbf{Q}_m \vec{V}'_m(t) = D \sum_{l=m,i,j,k} \mathbf{W}_l \vec{V}_l(t) + \vec{F}_m(\vec{V}_m) \quad m = 1, \dots, N, \quad (24)$$

where $\mathbf{Q}_m = \begin{pmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{pmatrix}$, $\mathbf{W}_l = \begin{pmatrix} w_{al1} & w_{bl1} & w_{cl1} \\ w_{al2} & w_{bl2} & w_{cl2} \\ w_{al3} & w_{bl3} & w_{cl3} \end{pmatrix}$, $\vec{V}_m = \begin{pmatrix} a_m(t) \\ b_m(t) \\ c_m(t) \end{pmatrix}$, $\vec{V}_l = \begin{pmatrix} a_l(t) \\ b_l(t) \\ c_l(t) \end{pmatrix}$, and $\vec{F}_m(\vec{V}_m) = \begin{pmatrix} (q_{11}/3) \sum_{l=i,j,k} F(u(x_{m,l}, y_{m,l})) \\ (q_{11}/3) \sum_{l=i,j,k} F(u(x_{m,l}, y_{m,l}))\xi_m(x_{m,l}, y_{m,l}) \\ (q_{11}/3) \sum_{l=i,j,k} F(u(x_{m,l}, y_{m,l}))\eta_m(x_{m,l}, y_{m,l}) \end{pmatrix}$.

3.2 The P^2 case

The procedure to construct the P^2 implicit DG spatial discretizations with third order accuracy will follow that of the P^1 case, with a larger semi-discretized system. The number of local equations on each triangle depends on the degree of freedoms of the approximation polynomial. On each triangle Δ_m , a quadratic polynomial

$$u(x, y, t) = a_m(t) + b_m(t)\xi_m + c_m(t)\eta_m + d_m(t)\xi_m\eta_m + e_m(t)\xi_m^2 + f_m(t)\eta_m^2 \quad (25)$$

will be constructed. In (14), we take $v = 1, \xi_m, \eta_m, \xi_m\eta_m, \xi_m^2, \eta_m^2$ on Δ_m and $v = 0$ elsewhere and convert the integral formulation to the semi-discretized ODE system

$$\mathbf{Q}_m \vec{V}'_m(t) = D \sum_{l=m,i,j,k} \mathbf{W}_l \vec{V}_l(t) + \vec{F}_m(\vec{V}_m) \quad m = 1, \dots, N, \quad (26)$$

$$\begin{aligned}
\text{where } \mathbf{Q}_m &= \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} & q_{15} & q_{16} \\ q_{21} & q_{22} & q_{23} & q_{24} & q_{25} & q_{26} \\ q_{31} & q_{32} & q_{33} & q_{34} & q_{35} & q_{36} \\ q_{41} & q_{42} & q_{43} & q_{44} & q_{45} & q_{46} \\ q_{51} & q_{52} & q_{53} & q_{54} & q_{55} & q_{56} \\ q_{61} & q_{62} & q_{63} & q_{64} & q_{65} & q_{66} \end{pmatrix}, \mathbf{W}_l = \begin{pmatrix} w_{al1} & w_{bl1} & w_{cl1} & w_{dl1} & w_{el1} & w_{fl1} \\ w_{al2} & w_{bl2} & w_{cl2} & w_{dl2} & w_{el2} & w_{fl2} \\ w_{al3} & w_{bl3} & w_{cl3} & w_{dl3} & w_{el3} & w_{fl3} \\ w_{al4} & w_{bl4} & w_{cl4} & w_{dl4} & w_{el4} & w_{fl4} \\ w_{al5} & w_{bl5} & w_{cl5} & w_{dl5} & w_{el5} & w_{fl5} \\ w_{al6} & w_{bl6} & w_{cl6} & w_{dl6} & w_{el6} & w_{fl6} \end{pmatrix}, \\
\vec{V}_m &= \begin{pmatrix} a_m(t) \\ b_m(t) \\ c_m(t) \\ d_m(t) \\ e_m(t) \\ f_m(t) \end{pmatrix}, \vec{V}_l = \begin{pmatrix} a_l(t) \\ b_l(t) \\ c_l(t) \\ d_l(t) \\ e_l(t) \\ f_l(t) \end{pmatrix}, \text{ and} \\
\vec{F}_m(\vec{V}_m) &= \begin{pmatrix} q_{11} \sum_{\nu=1}^7 \omega_\nu F(u(x_{m_\nu}, y_{m_\nu})) \\ q_{11} \sum_{\nu=1}^7 \omega_\nu F(u(x_{m_\nu}, y_{m_\nu})) \xi_m(x_{m_\nu}, y_{m_\nu}) \\ q_{11} \sum_{\nu=1}^7 \omega_\nu F(u(x_{m_\nu}, y_{m_\nu})) \eta_m(x_{m_\nu}, y_{m_\nu}) \\ q_{11} \sum_{\nu=1}^7 \omega_\nu F(u(x_{m_\nu}, y_{m_\nu})) \xi_m(x_{m_\nu}, y_{m_\nu}) \eta_m(x_{m_\nu}, y_{m_\nu}) \\ q_{11} \sum_{\nu=1}^7 \omega_\nu F(u(x_{m_\nu}, y_{m_\nu})) \xi_m^2(x_{m_\nu}, y_{m_\nu}) \\ q_{11} \sum_{\nu=1}^7 \omega_\nu F(u(x_{m_\nu}, y_{m_\nu})) \eta_m^2(x_{m_\nu}, y_{m_\nu}) \end{pmatrix}.
\end{aligned}$$

Again the detailed formulae for elements of matrices \mathbf{Q}_m , \mathbf{W}_m and \mathbf{W}_l , $l = i, j, k$ are given in the Appendix. These mesh-dependent constants don't depend on the numerical solution u and are pre-calculated and stored before the time evolution. (x_{m_ν}, y_{m_ν}) , $\nu = 1, \dots, 7$ are the Gaussian quadrature points in the triangle Δ_m for computing the integral involving the nonlinear reaction terms for the P^2 case. Their values are also given in the Appendix.

3.3 Application of Krylov IIF schemes

Finally we have the ODE system resulting from the P^1 (24) or P^2 (26) DG spatial discretization:

$$\vec{V}'_m(t) = D \sum_{l=m,i,j,k} \widetilde{\mathbf{W}}_l \vec{V}_l(t) + \vec{F}_m(\vec{V}_m), \quad m = 1, \dots, N, \quad (27)$$

where $\widetilde{\mathbf{W}}_l = \mathbf{Q}_m^{-1} \mathbf{W}_l$, $\vec{F}_m = \mathbf{Q}_m^{-1} \vec{F}_m$. \mathbf{Q}_m^{-1} and $\widetilde{\mathbf{W}}_l$ are mesh-dependent data and they don't depend on the numerical solution u . So we pre-calculate and store these data before the time evolution. Note that in this paper we consider that the nonlinear term $\vec{F}_m(\vec{V}_m)$ arises from the nonlinear reaction or source terms and no spatial derivative is involved in it, so the nonlinear operator \vec{F}_m is local, namely it only depends on the unknowns \vec{V}_m on the element Δ_m .

Assembling the local systems (27), we get the global ODE system arising from the implicit DG spatial discretization for a reaction-diffusion equation on a triangular mesh

$$U'(t) = \mathbf{A}U + \mathbf{F}(U) \quad (28)$$

where $U = (V_1^T, V_2^T, \dots, V_N^T)^T$, \mathbf{A} is a $p \cdot N \times p \cdot N$ sparse matrix with block structures, where p is the degrees of freedom on each triangular element. The nonzero elements

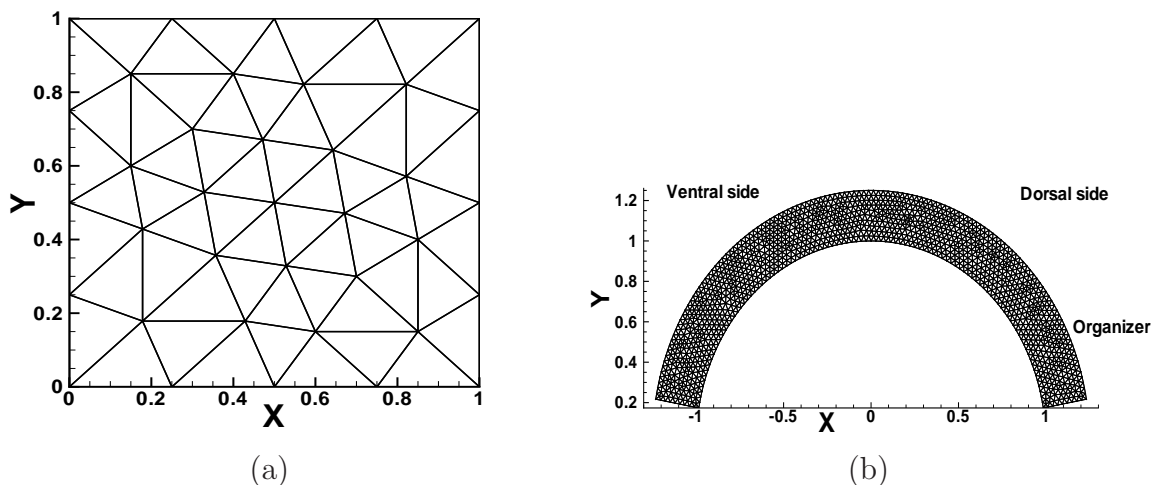


Figure 1: (a): A coarse mesh with 56 triangles in the convergence studies in Section 4.1; (b): a coarse mesh with 2415 triangles in the numerical simulations for the zebrafish model in Section 4.2.

in \mathbf{A} are $p \times p$ sub-matrices $D\widetilde{\mathbf{W}}_l$ distributed on corresponding locations, and

$$\mathbf{F}(U) = (\vec{F}_1(\vec{V}_1)^T, \vec{F}_2(\vec{V}_2)^T, \dots, \vec{F}_N(\vec{V}_N)^T)^T.$$

Remark: The sparse matrix \mathbf{A} is the P^1 or P^2 DG discretization (14) of the Laplacian operator $D\nabla^2$. For the sparse matrix \mathbf{A} we only store the nonzero elements (those $D\widetilde{\mathbf{W}}_l$ blocks) and their locations in the matrix [75]. In our implementation we avoid the operations involving zero elements in the sparse matrix \mathbf{A} .

Now for the ODE system (28), we can directly apply the Krylov IIF methods (10) for the time evolution. In our numerical experiments, we apply the second order Krylov IIF method (11) for the P^1 DG spatial discretization, and the third order Krylov IIF method (12) for the P^2 DG spatial discretization to achieve consistent spatial-temporal accuracy orders. As the original IIF methods, the Krylov IIF methods will result in a local nonlinear algebraic system on every triangular element. The number of algebraic equations of the local system is the product of the degrees of freedom p and the number of equations in the PDEs system. We use the Newton method to solve the small nonlinear algebraic system on every triangular element. In the Newton iterations to compute U_{n+1} , we use the numerical values U_n at time step t_n as the initial guess. And the threshold value for judging Newton iteration convergence is set to be 10^{-15} .

4 Numerical Experiments

In this section we present numerical examples to show the stability, accuracy and efficiency of the Krylov IIF methods in resolving the stiffness of the DG spatial operator for reaction-diffusion PDEs on 2D triangular meshes. The methods are firstly tested on a set of problems with exact solutions. Then application of the method to long-time simulation of a mathematical model which describes the dorsal-ventral pattern formation during the zebrafish embryo development will be shown. From numerical experiments

we can observe that large time step sizes are achieved in numerical computations of these parabolic PDEs by the DG method.

In this paper, all of the reaction-diffusion systems we considered are subject to no-flux boundary conditions. If the element edge e_l of Δ_m is aligned with the domain boundary $\partial\Omega$, we take $u^{in}|_{e_l} = u^{ext}|_{e_l}$, and $(\nabla u)^{in}|_{e_l} \cdot \vec{n}_{e_l} = (\nabla u)^{ext}|_{e_l} \cdot \vec{n}_{e_l} = 0$ in the numerical fluxes (15)-(17). Hence we have

$$\widehat{u}|_{e_l} = u^{in}|_{e_l}, \quad \widetilde{\nabla}u|_{e_l} \cdot \vec{n}_{e_l} = 0$$

in the scheme (14).

All computations in this paper are performed on a 2.39GHz, 8GB RAM Linux computer.

4.1 Numerical examples with exact solutions

In this subsection, we perform convergence studies for the second and third order Krylov IIF methods which are applied to P^1 and P^2 DG spatial discretizations respectively on triangular meshes, as shown in Fig. 1(a) which is a coarse one with 56 elements. The refinement of the triangular meshes is done in a uniform way, namely by cutting each triangle into four smaller similar ones.

To show the stiffness of the DG spatial discretization matrix \mathbf{A} for the diffusion operator ∇^2 , we study the eigenvalues and condition numbers related to \mathbf{A} . In Table 1, we list the negative largest eigenvalue $\lambda_0(\mathbf{A})$ of P^1 or P^2 DG discretization matrices \mathbf{A} for the diffusion operator ∇^2 on the domain $[0, 1]^2$ with no-flux boundary conditions, on successively refined meshes. The matrices \mathbf{A} have the 0 eigenvalue due to the no-flux boundary condition. We can see that the matrices \mathbf{A} have quite large magnitude eigenvalues for not so refined meshes. If an regular implicit method is used for the time discretization (e.g. the backward Euler method), a linear system with the coefficient matrix $I - \Delta t \mathbf{A}$ needs to be solved. In Table 1, we also list the condition numbers of the matrices $I - \Delta t \mathbf{A}$ for successively refined spatial-temporal meshes, with $\Delta t = h_{min}$. These condition numbers are quite large for not so refined 2D meshes. Next we will show that the Krylov IIF methods developed in this paper work quite well for this DG spatial discretization.

Example 1: consider the two-dimensional diffusion problem

$$\begin{cases} u_t = u_{xx} + u_{yy}, & (x, y) \in (0, 1) \times (0, 1) \\ u(x, y, 0) = \cos(\pi x) \cos(\pi y), \end{cases} \quad (29)$$

with the no-flux boundary conditions. The exact solution is $u(x, y, t) = e^{-2\pi^2 t} \cos(\pi x) \cos(\pi y)$. The computation is carried up to $T = 0.6$ with $M = 25$ at which the L^1 , L^2 and L^∞ errors are measured. The time step size $\Delta t = h_{min}$. CPU time, errors and order of accuracy for the second order Krylov IIF method with P^1 DG spatial discretization are reported in Table 2, and for the third order Krylov IIF method with P^2 DG spatial discretization are reported in Table 3. We can observe that we obtained desired accuracy orders for both cases. In the Tables, N_{ee} is the number of triangular elements of the computational meshes, and $N = 3 \cdot N_{ee}$ (P^1 case) or $N = 6 \cdot N_{ee}$ (P^2 case) is the number of unknown degrees of freedom in the spatial direction. If the original IIF schemes are used, it is computationally prohibitive to compute the matrix exponential

Table 1: Eigenvalue and condition number study. $\lambda_0(\mathbf{A})$ is the negative largest eigenvalue of P^1 or P^2 DG discretization matrices \mathbf{A} for the diffusion operator ∇^2 on the domain $[0, 1]^2$ with no-flux boundary conditions, on successively refined meshes. $\text{cond}(I - \Delta t \mathbf{A})$ are condition numbers for the linear operator $I - \Delta t \mathbf{A}$, which will arise if an implicit treatment of the diffusion term is used. Time step size $\Delta t = h_{min}$ is used. N_{ee} is the number of triangular elements of the computational meshes.

N_{ee}	Δt	P^1		P^2	
		$\text{cond}(I - \Delta t \mathbf{A})$	$\lambda_0(\mathbf{A})$	$\text{cond}(I - \Delta t \mathbf{A})$	$\lambda_0(\mathbf{A})$
14	0.2079	2.7549E+03	-3.1490E+03	2.2734E+04	-4.2638E+03
56	0.1039	4.8715E+03	-1.3522E+04	3.6438E+04	-1.7467E+04
224	0.0520	9.0636E+03	-5.5484E+04	6.3633E+04	-7.0285E+04
896	0.0260	1.9658E+04	-2.2407E+05	1.3349E+05	-2.8164E+05
3584	0.0130	3.6801E+04	-8.9920E+05	2.4045E+05	-1.1277E+06

for such huge $N \times N$ matrices, especially when the spatial meshes are refined. The Krylov subspace approximations we used in this paper only need to compute the matrix exponentials of 25×25 matrices, which are independent of the degrees of freedom N . As shown in the CPU time of the numerical Tables, we can see that the computations are very efficient. When the spatial mesh is refined, the number of degrees of freedom N increase 4 times and the total time steps double. From Tables 2 and 3, we see that the CPU time approximately increases 8 times when the mesh is refined once. So The CPU time approximately linearly depends on the number of degrees of freedom. Also, we obtain a clean second and third order accuracy in the computations for a fixed Krylov space dimension $M = 25$ for all meshes, and this indicates that the errors generated by the Krylov subspace approximations are much smaller than the truncation errors of the numerical schemes. This is also confirmed by the numerical results in Table 4, in which we show numerical errors if different dimensions M of the Krylov subspace are used in the second order Krylov IIF method with P^1 DG spatial discretization, for a fixed spatial mesh. We can see that the numerical errors are comparable for both smaller M and larger ones. So the numerical errors shown in Table 4 are mainly due to the truncation errors of the DG spatial discretizations (14)(the temporal truncation errors from the IIF part (5) are 0 since this problem only has the linear diffusion part), and the numerical errors by the Krylov subspace approximations are negligible in this example. We will further perform theoretical error analysis on this in our future work. The numerical results also show that even for this parabolic PDE, we can use a large time step size proportional to the spatial grid size for a stable and accurate computation. Actually for this problem which only has linear diffusion terms, in the time direction the Krylov IIF method can evolve the linear DG spatial discretization operator almost “exactly” up to the numerical errors of the Krylov subspace approximations. This is shown in both Table 4 and Table 5. From these two Tables, we can see that even if we use a very large time step size $\Delta t = 0.6$, namely we just use one time-step to reach the final time T , we can still obtain comparable numerical errors as those in the computations by using $\Delta t = h_{min}$ and a clear second order accuracy as shown in Table 5.

Table 2: Example 1. CPU time, error, and order of accuracy of the second order Krylov IIF method with P^1 DG spatial discretization. Final time $T = 0.6$.

N_{ee}	N	CPU(s)	L_1 error	order	L_2 error	order	L_∞ error	order
14	42	0.00	2.60E-06	-	3.17E-06	-	6.47E-06	-
56	168	0.02	1.20E-06	1.12	1.48E-06	1.10	2.70E-06	1.26
224	672	0.12	3.70E-07	1.70	4.54E-07	1.70	9.00E-07	1.59
896	2688	0.69	9.86E-08	1.91	1.21E-07	1.91	2.41E-07	1.90
3584	10752	5.24	2.49E-08	1.98	3.06E-08	1.98	6.09E-08	1.98
14336	43008	40.23	6.24E-09	2.00	7.67E-09	2.00	1.52E-08	2.00
57344	172032	368.65	1.56E-09	2.00	1.92E-09	2.00	3.81E-09	2.00

Table 3: Example 1. CPU time, error, and order of accuracy of the third order Krylov IIF method with P^2 DG spatial discretization. Final time $T = 0.6$.

N_{ee}	N	CPU(s)	L_1 error	order	L_2 error	order	L_∞ error	order
14	84	0.01	3.46E-07	-	4.02E-07	-	6.02E-07	-
56	336	0.05	1.12E-08	4.95	1.47E-08	4.78	6.34E-08	3.25
224	1344	0.26	8.19E-10	3.78	1.06E-09	3.79	8.25E-09	2.94
896	5376	1.99	7.24E-11	3.50	9.38E-11	3.50	1.04E-09	2.99
3584	21504	16.05	8.11E-12	3.16	1.06E-11	3.14	1.30E-10	3.00
14336	86016	134.62	9.83E-13	3.04	1.30E-12	3.03	1.63E-11	3.00
57344	344064	994.91	1.25E-13	2.98	1.65E-13	2.98	2.06E-12	2.98

Example 2: consider the two-dimensional problem with a linear reaction term

$$\begin{cases} u_t = u_{xx} + u_{yy} - u + 2\pi^2 e^{-t} \cos(\pi x) \cos(\pi y), & (x, y) \in (0, 1) \times (0, 1) \\ u(x, 0) = \cos(\pi x) \cos(\pi y), \end{cases} \quad (30)$$

with the no-flux boundary conditions. The exact solution is $u(x, y, t) = e^{-t} \cos(\pi x) \cos(\pi y)$. The computation is carried up to $T = 2.0$ with $M = 25$ at which the L^1 , L^2 and L^∞ errors are measured. The time step size $\Delta t = h_{min}$, and the larger ones $\Delta t = 2h_{min}$, $\Delta t = 4h_{min}$. CPU time, errors and order of accuracy for the second order Krylov IIF method with P^1 DG spatial discretization are reported in Table 6, and for the third order Krylov IIF method with P^2 DG spatial discretization are reported in Table 7. We can observe that we obtained desired accuracy orders for all cases. In the Tables, N_{ee} is the number of triangular elements of the computational meshes, and $N = 3 \cdot N_{ee}$ (P^1 case) or $N = 6 \cdot N_{ee}$ (P^2 case) is the number of unknown degrees of freedom in the spatial direction. As in the example 1, the Krylov IIF methods demonstrate excellent efficiency, accuracy and stability properties in this example. The stable and accurate computation results by using large time step sizes show that the methods have quite large linear stability region.

Table 4: Example 1. Numerical errors if different dimensions M of the Krylov subspace are used in the second order Krylov IIF method with P^1 DG spatial discretization. Final time $T = 0.6$. Two different time step sizes $\Delta t = 0.6$ and $\Delta t = h_{min}$ are used. The spatial mesh is the one which has $N_{ee} = 3584$ triangles.

M	$\Delta t = 0.6$			$\Delta t = h_{min}$		
	L_1 error	L_2 error	L_∞ error	L_1 error	L_2 error	L_∞ error
10	3.06E-08	3.77E-08	7.74E-08	2.45E-08	3.02E-08	5.98E-08
25	2.69E-08	3.32E-08	6.78E-08	2.49E-08	3.06E-08	6.09E-08
100	2.44E-08	3.01E-08	5.99E-08	2.49E-08	3.06E-08	6.09E-08
250	2.48E-08	3.05E-08	6.03E-08	2.49E-08	3.06E-08	6.09E-08

Table 5: Example 1. Error, and order of accuracy of the second order Krylov IIF method with P^1 DG spatial discretization. Final time $T = 0.6$. A large time step $\Delta t = 0.6$ is used, namely, only one step to the final time for this diffusion problem.

N_{ee}	N	L_1 error	order	L_2 error	order	L_∞ error	order
14	42	2.60E-06	-	3.18E-06	-	6.50E-06	-
56	168	1.20E-06	1.12	1.48E-06	1.11	2.71E-06	1.26
224	672	3.69E-07	1.70	4.54E-07	1.70	8.99E-07	1.59
896	2688	1.11E-07	1.73	1.37E-07	1.73	2.74E-07	1.72
3584	10752	2.69E-08	2.04	3.32E-08	2.04	6.78E-08	2.01
14336	43008	6.64E-09	2.02	8.19E-09	2.02	1.69E-08	2.00
57344	172032	1.70E-09	1.96	2.10E-09	1.96	4.35E-09	1.96

Example 3: consider the two-dimensional problem with a nonlinear reaction term

$$\begin{cases} u_t = u_{xx} + u_{yy} - u^2 + e^{-2t} \cos^2 \pi x \cos^2 \pi y + (2\pi^2 - 1)e^{-t} \cos \pi x \cos \pi y, & (x, y) \in (0, 1) \times (0, 1) \\ u(x, 0) = \cos(\pi x) \cos(\pi y), \end{cases} \quad (31)$$

with the no-flux boundary conditions. The exact solution is $u(x, y, t) = e^{-t} \cos(\pi x) \cos(\pi y)$. The computation is carried up to $T = 2.0$ with $M = 25$ at which the L^1 , L^2 and L^∞ errors are measured. The time step size $\Delta t = h_{min}$. CPU time, errors and order of accuracy for the second order Krylov IIF method with P^1 DG spatial discretization are reported in Table 8, and for the third order Krylov IIF method with P^2 DG spatial discretization are reported in Table 9. Again we obtained desired accuracy orders for both cases. In the Tables, N_{ee} is the number of triangular elements of the computational meshes, and $N = 3 \cdot N_{ee}$ (P^1 case) or $N = 6 \cdot N_{ee}$ (P^2 case) is the number of unknown degrees of freedom in the spatial direction. For this example with a nonlinear reaction term, we still draw the same conclusion as the last two examples, from the numerical convergence study about the efficiency, accuracy and stability properties of the Krylov IIF methods.

Table 6: Example 2. CPU time, error, and order of accuracy of the second order Krylov IIF method with P^1 DG spatial discretization. Final time $T = 2$.

$\Delta t = h_{min}$								
N_{ee}	N	CPU(s)	L_1 error	order	L_2 error	order	L_∞ error	order
14	42	0.01	2.10E-02	-	2.59E-02	-	9.31E-02	-
56	168	0.10	7.60E-03	1.47	9.56E-03	1.44	3.33E-02	1.48
224	672	0.46	2.26E-03	1.75	2.80E-03	1.77	9.35E-03	1.83
896	2688	2.82	8.25E-04	1.45	1.02E-03	1.46	3.04E-03	1.62
3584	10752	21.95	2.07E-04	1.99	2.56E-04	1.99	7.80E-04	1.96
14336	43008	171.41	5.23E-05	1.99	6.45E-05	1.99	1.99E-04	1.97
57344	172032	1662.97	1.33E-05	1.98	1.64E-05	1.98	5.12E-05	1.96
$\Delta t = 2h_{min}$								
N_{ee}	N	CPU(s)	L_1 error	order	L_2 error	order	L_∞ error	order
14	42	0.00	1.00E-01	-	1.30E-01	-	3.40E-01	-
56	168	0.05	2.33E-02	2.14	2.87E-02	2.15	7.57E-02	2.18
224	672	0.27	9.22E-03	1.34	1.14E-02	1.34	2.70E-02	1.49
896	2688	1.58	2.68E-03	1.78	3.30E-03	1.78	7.63E-03	1.82
3584	10752	11.89	9.44E-04	1.50	1.17E-03	1.50	2.59E-03	1.56
14336	43008	89.35	2.37E-04	2.00	2.92E-04	2.00	6.52E-04	1.99
57344	172032	741.47	5.95E-05	1.99	7.34E-05	1.99	1.65E-04	1.98
$\Delta t = 4h_{min}$								
N_{ee}	N	CPU(s)	L_1 error	order	L_2 error	order	L_∞ error	order
56	168	0.03	1.00E-01	-	1.30E-01	-	2.90E-01	-
224	672	0.16	2.44E-02	2.06	3.01E-02	2.06	6.53E-02	2.14
896	2688	0.95	9.61E-03	1.34	1.19E-02	1.34	2.48E-02	1.40
3584	10752	6.63	2.80E-03	1.78	3.45E-03	1.78	7.17E-03	1.79
14336	43008	48.03	9.74E-04	1.52	1.20E-03	1.52	2.47E-03	1.54
57344	172032	409.27	2.44E-04	2.00	3.01E-04	2.00	6.19E-04	2.00

4.2 Application to a morphogenesis system

Many mathematical models in developmental biology take the form of reaction-diffusion equations (13). In such systems, both diffusion and reaction terms are often very stiff. And long time numerical simulations are often needed for these systems. Hence robust and accurate methods which permit large time step sizes are desired. As demonstrated in one-dimensional systems [66], the IIF methods are quite efficient for such applications. In this section, We apply the second order Krylov IIF method with P^1 DG spatial discretization on a triangular mesh to a model for the study of embryonic patterning. The model is re-defined on a two dimensional domain which has a realistic shape as part of the embryo during its development.

One of the central problems in developmental biology is how uniform fields of cells are transformed into tissues with highly specialized cell types at distinct anatomical positions. In this process, diffusible morphogens produced by certain cells pattern the

Table 7: Example 2. CPU time, error, and order of accuracy of the third order Krylov IIF method with P^2 DG spatial discretization. Final time $T = 2$.

$\Delta t = h_{min}$								
N_{ee}	N	CPU(s)	L_1 error	order	L_2 error	order	L_∞ error	order
14	84	0.07	1.49E-02	-	1.84E-02	-	4.54E-02	-
56	336	0.31	5.11E-03	1.55	6.29E-03	1.54	1.31E-02	1.80
224	1344	1.81	9.44E-04	2.44	1.16E-03	2.43	2.36E-03	2.47
896	5376	14.21	1.99E-04	2.25	2.45E-04	2.25	4.92E-04	2.26
3584	21504	115.10	2.71E-05	2.88	3.34E-05	2.88	6.70E-05	2.88
14336	86016	940.71	3.59E-06	2.91	4.43E-06	2.91	8.88E-06	2.91
57344	344064	7898.45	4.76E-07	2.92	5.87E-07	2.92	1.18E-06	2.92
$\Delta t = 2h_{min}$								
N_{ee}	N	CPU(s)	L_1 error	order	L_2 error	order	L_∞ error	order
14	84	0.03	7.87E-02	-	9.73E-02	-	0.21	-
56	336	0.16	1.51E-02	2.38	1.86E-02	2.39	3.78E-02	2.48
224	1344	0.97	5.11E-03	1.56	6.31E-03	1.56	1.27E-02	1.58
896	5376	7.55	9.44E-04	2.44	1.16E-03	2.44	2.33E-03	2.44
3584	21504	58.61	1.99E-04	2.25	2.45E-04	2.25	4.91E-04	2.25
14336	86016	500.34	2.71E-05	2.88	3.34E-05	2.88	6.69E-05	2.88
57344	344064	4288.04	3.59E-06	2.91	4.43E-06	2.91	8.86E-06	2.92
$\Delta t = 4h_{min}$								
N_{ee}	N	CPU(s)	L_1 error	order	L_2 error	order	L_∞ error	order
56	336	0.09	7.89E-02	-	9.73E-02	-	2.00E-01	-
224	1344	0.54	1.51E-02	2.38	1.86E-02	2.38	3.73E-02	2.39
896	5376	4.03	5.12E-03	1.56	6.31E-03	1.56	1.26E-02	1.56
3584	21504	30.85	9.44E-04	2.44	1.16E-03	2.44	2.33E-03	2.44
14336	86016	249.01	1.99E-04	2.25	2.45E-04	2.25	4.91E-04	2.25
57344	344064	1971.68	2.71E-05	2.88	3.34E-05	2.88	6.69E-05	2.88

surrounding tissue through interactions with certain proteins on the cell membrane. Morphogens are signaling molecules that, when bound to cell receptors, assign different cell fates at different concentrations. This role of morphogens has been the prevailing thought in tissue patterning for over half a century; but only recently have there been sufficient experimental data for us to begin to understand how various morphogens interact and patterns emerge. Now, mathematical modelling, analysis and computations have been very helpful to understand and identify underline biological mechanisms in morphogenesis systems [54, 55].

The example we are considering is a system of reaction-diffusion equations arising from mathematical modeling of a morphogenesis problem during the dorsal-ventral patterning of zebrafish embryo development [92]. The model predicts that the dorsal organizer of the zebrafish embryo plays a key role in forming a stable non-homogeneous morphogen gradient, and the prediction agrees well with the existing biological exper-

Table 8: Example 3. CPU time, error, and order of accuracy of the second order Krylov IIF method with P^1 DG spatial discretization. Final time $T = 2$.

N_{ee}	N	CPU(s)	L_1 error	order	L_2 error	order	L_∞ error	order
14	42	0.06	0.55	-	0.55	-	0.67	-
56	168	0.11	1.45E-02	5.24	1.70E-02	5.01	4.87E-02	3.78
224	672	0.52	2.44E-03	2.58	3.02E-03	2.50	9.93E-03	2.29
896	2688	3.19	8.72E-04	1.48	1.08E-03	1.49	3.19E-03	1.64
3584	10752	24.60	2.19E-04	1.99	2.70E-04	1.99	8.18E-04	1.96
14336	43008	197.79	5.52E-05	1.99	6.80E-05	1.99	2.09E-04	1.97
57344	172032	1670.53	1.40E-05	1.98	1.72E-05	1.98	5.36E-05	1.96

Table 9: Example 3. CPU time, error, and order of accuracy of the third order Krylov IIF method with P^2 DG spatial discretization. Final time $T = 2$.

N_{ee}	N	CPU(s)	L_1 error	order	L_2 error	order	L_∞ error	order
14	84	0.08	0.15	-	0.15	-	0.20	-
56	336	0.35	6.57E-03	4.49	7.98E-03	4.23	1.83E-02	3.46
224	1344	2.08	1.01E-03	2.69	1.25E-03	2.67	2.58E-03	2.83
896	5376	16.18	2.10E-04	2.27	2.59E-04	2.27	5.28E-04	2.29
3584	21504	127.09	2.86E-05	2.88	3.53E-05	2.88	7.17E-05	2.88
14336	86016	1139.88	3.77E-06	2.92	4.65E-06	2.92	9.44E-06	2.93
57344	344064	8864.08	4.91E-07	2.94	6.05E-07	2.94	1.22E-06	2.95

iments. First we briefly describe the reaction-diffusion model in the following.

Consider the zebrafish embryo from the end of blastula period to the middle of gastrula period (approximately 4.5-7.5 hours after fertilization). We are interested in the patterning of the dorsal-ventral tissues. On a two dimensional approximation, the dorsal-ventral region has a shape as shown in Fig. 1(b), with the dorsal organizer being located at the corner of the dorsal region. The whole domain Ω can be represented by

$$\Omega = \{(x, y) | r_{\min}^2 \leq x^2 + y^2 \leq r_{\max}^2, \quad \cos 170^\circ \leq \frac{x}{\sqrt{x^2 + y^2}} \leq \cos 10^\circ\},$$

and the dorsal organizer region Ω_O is

$$\Omega_O = \{(x, y) | r_{\min}^2 \leq x^2 + y^2 \leq r_{\max}^2, \quad \cos 30^\circ \leq \frac{x}{\sqrt{x^2 + y^2}} \leq \cos 10^\circ\},$$

where r_{\min} and r_{\max} are radii of the inner and outer boundaries.

Let $[L]$ and $[LR]$ denote the concentration of the morphogen BMP and the concentration of BMP-receptor complexes (to which BMP signaling is assumed to be proportional), respectively. The concentration of the free molecule Chordin (an inhibitor of BMP) and the concentration of BMP-Chordin complex are denoted by $[C]$

and $[LC]$, respectively. Let R_0 denote the total receptor concentration and let D_L , D_C , and D_{LC} represent the three diffusion coefficients for BMP, Chordin, BMP-Chordin complexes, respectively. The values k_{on} , k_{off} , k_{deg} , j_{on} , j_{off} , and τ are the binding and degradation rates for BMP, Chordin, and their complexes. With this notation, the reaction-diffusion model that is formulated in [92] is as follows:

$$\begin{cases} \frac{\partial[L]}{\partial t} = D_L \nabla^2[L] - k_{on}[L](R_0 - [LR]) + k_{off}[LR] - j_{on}[L][C] + (j_{off} + \tau)[LC] + V_L; \\ \frac{\partial[LR]}{\partial t} = k_{on}[L](R_0 - [LR]) - (k_{off} + k_{deg})[LR]; \\ \frac{\partial[LC]}{\partial t} = D_{LC} \nabla^2[LC] + j_{on}[L][C] - (j_{off} + \tau)[LC]; \\ \frac{\partial[C]}{\partial t} = D_C \nabla^2[C] - j_{on}[L][C] + j_{off}[LC] + V_C, \end{cases} \quad (32)$$

where V_C and V_L are the production rates of molecules Chordin and BMP, respectively, defined by

$$V_C = V_{Cmin} + \frac{V_{Cmax} - V_{Cmin}}{1 + \gamma_C[LR]} + \begin{cases} V_{Corg}e^{-at}, & \text{if } (x, y) \in \Omega_O; \\ 0, & \text{otherwise.} \end{cases} \quad (33)$$

$$V_L = V_{Lmin} + \frac{V_{Lmax} - V_{Lmin}}{1 + \gamma_L[LR]^{-1}} + V_{Lmat}e^{-bt}. \quad (34)$$

The terms $V_{Corg}e^{-at}$ and $V_{Lmat}e^{-bt}$ represent the maternal production rates of Chordin and BMP [92].

The system (32) is subjected to the no-flux boundary conditions

$$\frac{\partial[L]}{\partial n} = \frac{\partial[LC]}{\partial n} = \frac{\partial[C]}{\partial n} = 0 \text{ for } (x, y) \in \partial\Omega. \quad (35)$$

Parameters in the model are the following biological reasonable parameters [92]

$$\begin{aligned} r_{min} &= 0.028cm, r_{max} = 0.088cm, \\ D_L = D_{LC} = D_C &= 8.5 \times 10^{-7}cm^2s^{-1}, & k_{on} &= 0.4\mu Ms^{-1}, \\ R_0 &= 3.0\mu M, & k_{off} &= 4 \times 10^{-6}s^{-1}, \\ j_{on} &= 10\mu Ms^{-1}, & j_{off} &= 1.0 \times 10^{-5}s^{-1}, \\ \tau &= 0.01s^{-1}, & k_{deg} &= 5.0 \times 10^{-4}s^{-1}, \\ V_{Cmin} &= 8.0 \times 10^{-4}\mu Ms^{-1}, & V_{Cmax} &= 8.0 \times 10^{-2}\mu Ms^{-1}, \\ V_{Corg} &= 6.68 \times 10^{-1}\mu Ms^{-1}, & \gamma_C &= 10\mu M^{-1}, \\ a &= 0.0167s^{-1}, & V_{Lmin} &= 1.0 \times 10^{-5}\mu Ms^{-1}, \\ V_{Lmax} &= 6.0 \times 10^{-3}\mu Ms^{-1}, & \gamma_L &= 10\mu M, \\ V_{Lmat} &= 5.01 \times 10^{-2}\mu Ms^{-1}, & b &= 0.0167s^{-1}, \end{aligned} \quad (36)$$

The system (32) can be non-dimensionalized with the following normalized quantities

$$T = \frac{D}{r_{min}^2}t, \quad X = \frac{x}{r_{min}}, \quad Y = \frac{y}{r_{min}}, \quad (37)$$

$$\{f_L, g_L, h_L, f_S, h_S, \tau_S\} = \frac{r_{min}^2}{D} \{k_{off}, k_{deg}, k_{on}R_0, j_{off}, j_{on}R_0, \tau\}, \quad (38)$$

$$\{A, B, U, S\} = \frac{1}{R_0} \{[L], [LR], [LC], [C]\}, \quad (39)$$

$$\begin{aligned} &\{W_{Cmin}, W_{Cmax}, W_{Corg}, W_{Lmin}, W_{Lmax}, W_{Lmat}\} \\ &= \frac{r_{min}^2}{DR_0} \{V_{Cmin}, V_{Cmax}, V_{Corg}, V_{Lmin}, V_{Lmax}, V_{Lmat}\}, \end{aligned} \quad (40)$$

$$\{P, Q\} = \frac{r_{min}^2}{D} \{a, b\}, \quad (41)$$

$$\{d_L, d_{LC}, d_C\} = \frac{1}{D} \{D_L, D_{LC}, D_C\}, \quad \overline{\gamma_C} = \gamma_C R_0, \quad \overline{\gamma_L} = \frac{\gamma_L}{R_0}, \quad (42)$$

where D is the maximum of D_L , D_{LC} and D_C in (36).

In terms of the normalized quantities, the system (32) becomes to the following dimensionless form

$$\begin{cases} \frac{\partial A}{\partial T} = d_L \nabla^2 A - h_L A(1 - B) + f_L B - h_S A S + (f_S + \tau_S)U + W_L; \\ \frac{\partial B}{\partial T} = h_L A(1 - B) - (f_L + g_L)B; \\ \frac{\partial U}{\partial T} = d_{LC} \nabla^2 U + h_S A S - (f_S + \tau_S)U; \\ \frac{\partial S}{\partial T} = d_C \nabla^2 S - h_S A S + f_S U + W_C, \end{cases} \quad (43)$$

where

$$W_C = W_{Cmin} + \frac{W_{Cmax} - W_{Cmin}}{1 + \overline{\gamma_C} B} + \begin{cases} W_{Corg} e^{-PT}, & \text{if } X \geq \frac{7}{8}; \\ 0, & \text{otherwise.} \end{cases} \quad (44)$$

$$W_L = W_{Lmin} + \frac{W_{Lmax} - W_{Lmin}}{1 + \overline{\gamma_L} B^{-1}} + W_{Lmat} e^{-QT}. \quad (45)$$

The non-dimensionalized parameters corresponding to those in (36) are:

$$\begin{aligned} d_L = d_{LS} = d_C &= 1, & h_L &= 1.1068 \times 10^3, \\ f_L &= 3.6894 \times 10^{-3}, & h_S &= 2.7671 \times 10^4, \\ f_S &= 9.2235 \times 10^{-3}, & \tau_S &= 9.2235, \\ g_L &= 0.4612, & W_{Cmin} &= 0.24596, \\ W_{Cmax} &= 24.596, & W_{Corg} &= 2.0538 \times 10^2, \\ \overline{\gamma_C} &= 30, & P &= 15.4032, \\ W_{Lmin} &= 3.0745 \times 10^{-3}, & W_{Lmax} &= 1.8447, \\ \overline{\gamma_L} &= 3.3333, & W_{Lmat} &= 15.4032, & Q &= 15.4032. \end{aligned} \quad (46)$$

We use our new second order Krylov IIF method with P^1 DG spatial scheme to simulate the system (43)-(46), to the normalized final time $T = 10$. The spatial triangular mesh is a much more refined one based on the mesh shown in Figure 1(b). It has 154560 triangular elements. The sparse matrix \mathbf{A} has the huge size 463680×463680 . The original IIF methods can *not* be applied to such big size problem directly on a regular computer. But our Krylov IIF method designed in this paper can simulate the system stably and efficiently. The time step size is taken to be $\Delta t = 0.5h_{min} = 6.416 \times 10^{-4}$ which is the advantage of the implicit methods for a parabolic problem. The simulation results of the normalized concentrations of morphogen molecules are shown in Figure 2. The Figure 2(b) shows a morphogen gradient of the BMP-receptor complex which induces the cell differentiation and tissue patterns. In the zebrafish dorsal-ventral patterning, the high concentration region of the BMP-receptor complex will develop into the ventral tissue, and the low concentration region of the BMP-receptor complex will develop into the dorsal tissue as shown in the Figure 2.

5 Discussions and Conclusions

In this paper, we contributed two aspects in high order numerical methods. One is the development of a new Krylov subspace based implicit integration factor method for solving large ODE systems with both stiff linear and nonlinear terms, arising from numerical spatial discretization of time-dependent partial differential equations (PDEs) with linear high order terms and stiff lower order nonlinear terms. The method can preserve the nice property of the original IIF scheme [66] that the exact evaluation of the linear part is decoupled from the implicit treatment of the nonlinear part. Furthermore, it can efficiently and accurately solve the large ODE systems arising from spatial discretization (e.g. DG methods) on high dimensional unstructured meshes for PDEs defined on complex geometrical domains. At the same time, this method provides an efficient and robust time discretization technique for DG methods on unstructured meshes for solving PDEs which have high order spatial derivatives. Development of efficient high-order accurate time-stepping methods for DG methods applied to solve PDEs which have high order derivatives and defined on multi-dimensional spatial domains is still an active area. This is the second aspect in which this paper is trying to contribute.

By using numerical examples of solving reaction-diffusion PDEs with exact solutions and a realistic mathematical model in morphogenesis, we show the nice efficiency, stability and accuracy of the Krylov IIF-DG method. For this method, we can use a large time step size which is proportional to the spatial grid size to solve parabolic PDEs such as the reaction-diffusion equations. Numerical examples also show that the error generated by the Krylov subspace approximation with a quite small dimension (the dimension $M = 25$ for all mesh sizes) does *not* affect the accuracy orders of the IIF time discretizations ((5) or (6)) or the DG spatial discretizations (14), hence it is already much smaller than the DG spatial and IIF temporal truncation errors.

Because the computations of matrix exponential by the scaling and squaring method [42] are only needed for very small matrices (25×25 matrices in this paper), operations required with the original huge matrix are a few matrix-by-vector multiplications, which makes the algorithm easy to parallelize.

In this paper, we only used the numerical experiments to test the algorithm. No

analysis results are reported. The current ongoing work includes the analysis of the error generated by the Krylov subspace approximation and its effects on the truncation errors of the IIF and DG discretizations. The error analysis techniques in [36] will be used. In the numerical experiments of this paper, reaction-diffusion PDEs are used as examples. The Krylov IIF methods developed in this paper can be straightforwardly extended to solve more complex equations such as advection-reaction-diffusion PDEs and also applied to other DG methods such as the LDG methods [90, 88] for solving PDEs involving high order spatial derivatives on multi-dimensional unstructured meshes. These are our planned research at the next stage.

Appendix: Formulae for mesh-dependent constants in the implicit DG schemes

We introduce the notation

$$t_{i_1 i_2 i_3 i_4} \stackrel{\text{def}}{=} \int_{e_l} \xi_m^{i_1} \xi_l^{i_2} \eta_m^{i_3} \eta_l^{i_4} ds,$$

where i_1, i_2, i_3 , and i_4 are integers from 0 to 4. For example $t_{1203} = \int_{e_l} \xi_m^1 \xi_l^2 \eta_m^0 \eta_l^3 ds = \int_{e_l} \xi_m \xi_l^2 \eta_l^3 ds$.

A) The P^1 case

1. The matrix $\mathbf{Q}_m = \begin{pmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{pmatrix}$:

$$\begin{aligned} q_{11} &= \int_{\Delta_m} dx, \\ q_{12} = q_{21} &= \int_{\Delta_m} \xi_m dx, \\ q_{13} = q_{31} &= \int_{\Delta_m} \eta_m dx, \\ q_{22} &= \int_{\Delta_m} \xi_m^2 dx, \\ q_{23} = q_{32} &= \int_{\Delta_m} \xi_m \eta_m dx, \\ q_{33} &= \int_{\Delta_m} \eta_m^2 dx. \end{aligned}$$

2. The matrix $\mathbf{W}_m = \begin{pmatrix} w_{am1} & w_{bm1} & w_{cm1} \\ w_{am2} & w_{bm2} & w_{cm2} \\ w_{am3} & w_{bm3} & w_{cm3} \end{pmatrix}$:

$$\begin{aligned}
w_{am1} &= \sum_{l=i,j,k} (-\beta t_{0000}), \\
w_{bm1} &= \sum_{l=i,j,k} \left(\frac{t_{0000} n_{l,x}}{2h_m} - \beta t_{1000} \right), \\
w_{cm1} &= \sum_{l=i,j,k} \left(\frac{t_{0000} n_{l,y}}{2h_m} - \beta t_{0010} \right), \\
w_{am2} &= \sum_{l=i,j,k} \left(-\frac{t_{0000} n_{l,x}}{2h_m} - \beta t_{1000} \right), \\
w_{bm2} &= \sum_{l=i,j,k} (-\beta t_{2000}), \\
w_{cm2} &= \sum_{l=i,j,k} \left(\frac{t_{1000} n_{l,y} - t_{0010} n_{l,x}}{2h_m} - \beta t_{1010} \right), \\
w_{am3} &= \sum_{l=i,j,k} \left(-\frac{t_{0000} n_{l,y}}{2h_m} - \beta t_{0010} \right), \\
w_{bm3} &= \sum_{l=i,j,k} \left(\frac{t_{0010} n_{l,x} - t_{1000} n_{l,y}}{2h_m} - \beta t_{1010} \right), \\
w_{cm3} &= \sum_{l=i,j,k} (-\beta t_{0020}).
\end{aligned}$$

3. The matrix $\mathbf{W}_l = \begin{pmatrix} w_{al1} & w_{bl1} & w_{cl1} \\ w_{al2} & w_{bl2} & w_{cl2} \\ w_{al3} & w_{bl3} & w_{cl3} \end{pmatrix}$, $l = i, j, k$:

$$\begin{aligned}
w_{al1} &= \beta t_{0000}, \\
w_{bl1} &= \frac{t_{0000} n_{l,x}}{2h_l} + \beta t_{0100}, \\
w_{cl1} &= \frac{t_{0000} n_{l,y}}{2h_l} + \beta t_{0001}, \\
w_{al2} &= -\frac{t_{0000} n_{l,x}}{2h_m} + \beta t_{1000}, \\
w_{bl2} &= \frac{t_{1000} n_{l,x}}{2h_l} - \frac{t_{0100} n_{l,x}}{2h_m} + \beta t_{1100}, \\
w_{cl2} &= \frac{t_{1000} n_{l,y}}{2h_l} - \frac{t_{0001} n_{l,x}}{2h_m} + \beta t_{1001}, \\
w_{al3} &= -\frac{t_{0000} n_{l,y}}{2h_m} + \beta t_{0010}, \\
w_{bl3} &= \frac{t_{0010} n_{l,x}}{2h_l} - \frac{t_{0100} n_{l,y}}{2h_m} + \beta t_{0110}, \\
w_{cl3} &= \frac{t_{0010} n_{l,y}}{2h_l} - \frac{t_{0001} n_{l,y}}{2h_m} + \beta t_{0011}.
\end{aligned}$$

B) The P^2 case

1. The matrix $\mathbf{Q}_m = \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} & q_{15} & q_{16} \\ q_{21} & q_{22} & q_{23} & q_{24} & q_{25} & q_{26} \\ q_{31} & q_{32} & q_{33} & q_{34} & q_{35} & q_{36} \\ q_{41} & q_{42} & q_{43} & q_{44} & q_{45} & q_{46} \\ q_{51} & q_{52} & q_{53} & q_{54} & q_{55} & q_{56} \\ q_{61} & q_{62} & q_{63} & q_{64} & q_{65} & q_{66} \end{pmatrix} :$

$$\begin{aligned} q_{11} &= \int_{\Delta_m} dx, \\ q_{12} = q_{21} &= \int_{\Delta_m} \xi_m dx, \\ q_{13} = q_{31} &= \int_{\Delta_m} \eta_m dx, \\ q_{14} = q_{23} = q_{32} = q_{41} &= \int_{\Delta_m} \xi_m \eta_m dx, \\ q_{15} = q_{51} = q_{22} &= \int_{\Delta_m} \xi_m^2 dx, \\ q_{16} = q_{61} = q_{33} &= \int_{\Delta_m} \eta_m^2 dx, \\ q_{24} = q_{42} = q_{35} = q_{53} &= \int_{\Delta_m} \xi_m^2 \eta_m dx, \\ q_{25} = q_{52} &= \int_{\Delta_m} \xi_m^3 dx, \\ q_{26} = q_{62} = q_{34} = q_{43} &= \int_{\Delta_m} \xi_m \eta_m^2 dx, \\ q_{36} = q_{63} &= \int_{\Delta_m} \eta_m^3 dx, \\ q_{44} = q_{56} = q_{65} &= \int_{\Delta_m} \xi_m^2 \eta_m^2 dx, \\ q_{45} = q_{54} &= \int_{\Delta_m} \xi_m^3 \eta_m dx, \\ q_{46} = q_{64} &= \int_{\Delta_m} \xi_m \eta_m^3 dx, \\ q_{55} &= \int_{\Delta_m} \xi_m^4 dx, \\ q_{66} &= \int_{\Delta_m} \eta_m^4 dx, \end{aligned}$$

2. The matrix $\mathbf{W}_m = \begin{pmatrix} w_{am1} & w_{bm1} & w_{cm1} & w_{dm1} & w_{em1} & w_{fm1} \\ w_{am2} & w_{bm2} & w_{cm2} & w_{dm2} & w_{em2} & w_{fm2} \\ w_{am3} & w_{bm3} & w_{cm3} & w_{dm3} & w_{em3} & w_{fm3} \\ w_{am4} & w_{bm4} & w_{cm4} & w_{dm4} & w_{em4} & w_{fm4} \\ w_{am5} & w_{bm5} & w_{cm5} & w_{dm5} & w_{em5} & w_{fm5} \\ w_{am6} & w_{bm6} & w_{cm6} & w_{dm6} & w_{em6} & w_{fm6} \end{pmatrix} :$

The formulae for w_{am1} , w_{bm1} , w_{cm1} , w_{am2} , w_{bm2} , w_{cm2} , w_{am3} , w_{bm3} and w_{cm3} are the same as those in the P^1 case.

$$\begin{aligned}
w_{dm1} &= \sum_{l=i,j,k} \left(\frac{n_{l,x}t_{0010} + n_{l,y}t_{1000}}{2h_m} - \beta t_{1010} \right), \\
w_{em1} &= \sum_{l=i,j,k} \left(\frac{n_{l,x}t_{1000}}{h_m} - \beta t_{2000} \right), \\
w_{fm1} &= \sum_{l=i,j,k} \left(\frac{n_{l,y}t_{0010}}{h_m} - \beta t_{0020} \right), \\
w_{dm2} &= \sum_{l=i,j,k} \left(\frac{n_{l,y}t_{2000}}{2h_m} - \beta t_{2010} \right), \\
w_{em2} &= \sum_{l=i,j,k} \left(\frac{-n_{l,x}t_{2000} + 2n_{l,x}t_{2000}}{2h_m} - \beta t_{3000} \right), \\
w_{fm2} &= \sum_{l=i,j,k} \left(\frac{-n_{l,x}t_{0020} + 2n_{l,y}t_{1010}}{2h_m} - \beta t_{1020} \right), \\
w_{dm3} &= \sum_{l=i,j,k} \left(\frac{n_{l,x}t_{0020}}{2h_m} - \beta t_{1020} \right), \\
w_{em3} &= \sum_{l=i,j,k} \left(\frac{-n_{l,y}t_{2000} + 2n_{l,x}t_{1010}}{2h_m} - \beta t_{2010} \right), \\
w_{fm3} &= \sum_{l=i,j,k} \left(\frac{-n_{l,y}t_{0020} + 2n_{l,y}t_{0020}}{2h_m} - \beta t_{0030} \right), \\
w_{am4} &= \sum_{l=i,j,k} \left(\frac{-n_{l,x}t_{0010} - n_{l,y}t_{1000}}{2h_m} - \beta t_{1010} \right), \\
w_{bm4} &= \sum_{l=i,j,k} \left(-\frac{n_{l,y}t_{2000}}{2h_m} - \beta t_{2010} \right), \\
w_{cm4} &= \sum_{l=i,j,k} \left(-\frac{n_{l,x}t_{0020}}{2h_m} - \beta t_{1020} \right), \\
w_{dm4} &= \sum_{l=i,j,k} (-\beta t_{2020}), \\
w_{em4} &= \sum_{l=i,j,k} \left(\frac{n_{l,x}t_{2010} - n_{l,y}t_{3000}}{2h_m} - \beta t_{3010} \right), \\
w_{fm4} &= \sum_{l=i,j,k} \left(\frac{-n_{l,x}t_{0030} + n_{l,y}t_{1020}}{2h_m} - \beta t_{1030} \right),
\end{aligned}$$

$$\begin{aligned}
w_{am5} &= \sum_{l=i,j,k} \left(\frac{2q_{11}}{h_m^2} - \frac{n_{l,x}t_{1000}}{h_m} - \beta t_{2000} \right), \\
w_{bm5} &= \sum_{l=i,j,k} \left(\frac{2q_{12}}{h_m^2} - \frac{n_{l,x}t_{2000}}{2h_m} - \beta t_{3000} \right), \\
w_{cm5} &= \sum_{l=i,j,k} \left(\frac{2q_{13}}{h_m^2} - \frac{n_{l,x}t_{1010}}{h_m} + \frac{n_{l,y}t_{2000}}{2h_m} - \beta t_{2010} \right), \\
w_{dm5} &= \sum_{l=i,j,k} \left(\frac{2q_{14}}{h_m^2} - \frac{n_{l,x}t_{2010}}{2h_m} + \frac{n_{l,y}t_{3000}}{2h_m} - \beta t_{3010} \right), \\
w_{em5} &= \sum_{l=i,j,k} \left(\frac{2q_{15}}{h_m^2} - \beta t_{4000} \right), \\
w_{fm5} &= \sum_{l=i,j,k} \left(\frac{2q_{16}}{h_m^2} - \frac{n_{l,x}t_{1020}}{h_m} + \frac{n_{l,y}t_{2010}}{h_m} - \beta t_{2020} \right), \\
w_{am6} &= \sum_{l=i,j,k} \left(\frac{2q_{11}}{h_m^2} - \frac{n_{l,y}t_{0010}}{h_m} - \beta t_{0020} \right), \\
w_{bm6} &= \sum_{l=i,j,k} \left(\frac{2q_{12}}{h_m^2} - \frac{n_{l,y}t_{1010}}{h_m} + \frac{n_{l,x}t_{0020}}{2h_m} - \beta t_{1020} \right), \\
w_{cm6} &= \sum_{l=i,j,k} \left(\frac{2q_{13}}{h_m^2} - \frac{n_{l,y}t_{0020}}{2h_m} - \beta t_{0030} \right), \\
w_{dm6} &= \sum_{l=i,j,k} \left(\frac{2q_{14}}{h_m^2} - \frac{n_{l,y}t_{1020}}{2h_m} + \frac{n_{l,x}t_{0030}}{2h_m} - \beta t_{1030} \right), \\
w_{em6} &= \sum_{l=i,j,k} \left(\frac{2q_{15}}{h_m^2} - \frac{n_{l,y}t_{2010}}{h_m} + \frac{n_{l,x}t_{1020}}{h_m} - \beta t_{2020} \right), \\
w_{fm6} &= \sum_{l=i,j,k} \left(\frac{2q_{16}}{h_m^2} - \beta t_{0040} \right).
\end{aligned}$$

3. The matrix $\mathbf{W}_l = \begin{pmatrix} w_{al1} & w_{bl1} & w_{cl1} & w_{dl1} & w_{el1} & w_{fl1} \\ w_{al2} & w_{bl2} & w_{cl2} & w_{dl2} & w_{el2} & w_{fl2} \\ w_{al3} & w_{bl3} & w_{cl3} & w_{dl3} & w_{el3} & w_{fl3} \\ w_{al4} & w_{bl4} & w_{cl4} & w_{dl4} & w_{el4} & w_{fl4} \\ w_{al5} & w_{bl5} & w_{cl5} & w_{dl5} & w_{el5} & w_{fl5} \\ w_{al6} & w_{bl6} & w_{cl6} & w_{dl6} & w_{el6} & w_{fl6} \end{pmatrix}$, $l = i, j, k$:

The formulae for w_{al1} , w_{bl1} , w_{cl1} , w_{al2} , w_{bl2} , w_{cl2} , w_{al3} , w_{bl3} and w_{cl3} are the same as those in the P^1 case.

$$\begin{aligned}
w_{dl1} &= \frac{t_{0001}n_{l,x}}{2h_l} + \frac{t_{0100}n_{l,y}}{2h_l} + \beta t_{0101}, \\
w_{el1} &= \frac{t_{0100}n_{l,x}}{h_l} + \beta t_{0200}, \\
w_{fl1} &= \frac{t_{0001}n_{l,y}}{h_l} + \beta t_{0002}, \\
w_{dl2} &= -\frac{t_{0101}n_{l,x}}{2h_m} + \frac{t_{1001}n_{l,x} + t_{1100}n_{l,y}}{2h_l} + \beta t_{1101}, \\
w_{el2} &= \frac{t_{1100}n_{l,x}}{h_l} - \frac{t_{0200}n_{l,x}}{2h_m} + \beta t_{1200}, \\
w_{fl2} &= \frac{t_{1001}n_{l,y}}{h_l} - \frac{t_{0002}n_{l,x}}{2h_m} + \beta t_{1002}, \\
w_{dl3} &= -\frac{t_{0101}n_{l,y}}{2h_m} + \frac{t_{0011}n_{l,x} + t_{0110}n_{l,y}}{2h_l} + \beta t_{0111}, \\
w_{el3} &= \frac{t_{0110}n_{l,x}}{h_l} - \frac{t_{0200}n_{l,y}}{2h_m} + \beta t_{0210}, \\
w_{fl3} &= \frac{t_{0011}n_{l,y}}{h_l} - \frac{t_{0002}n_{l,y}}{2h_m} + \beta t_{0012}, \\
w_{al4} &= -\frac{t_{0010}n_{l,x} + t_{1000}n_{l,y}}{2h_m} + \beta t_{1010}, \\
w_{bl4} &= -\frac{t_{0110}n_{l,x} + t_{1100}n_{l,y}}{2h_m} + \frac{t_{1010}n_{l,x}}{2h_l} + \beta t_{1110}, \\
w_{cl4} &= -\frac{t_{0011}n_{l,x} + t_{1001}n_{l,y}}{2h_m} + \frac{t_{1010}n_{l,y}}{h_l} + \beta t_{1011}, \\
w_{dl4} &= -\frac{t_{0111}n_{l,x} + t_{1101}n_{l,y}}{2h_m} + \frac{t_{1011}n_{l,x} + t_{1110}n_{l,y}}{2h_l} + \beta t_{1111}, \\
w_{el4} &= -\frac{t_{0210}n_{l,x} + t_{1200}n_{l,y}}{2h_m} + \frac{t_{1110}n_{l,x}}{h_l} + \beta t_{1210}, \\
w_{fl4} &= -\frac{t_{0012}n_{l,x} + t_{1002}n_{l,y}}{2h_m} + \frac{t_{1011}n_{l,y}}{h_l} + \beta t_{1012}, \\
w_{al5} &= -\frac{t_{1000}n_{l,x}}{h_m} + \beta t_{2000}, \\
w_{bl5} &= -\frac{t_{1100}n_{l,x}}{h_m} + \frac{t_{2000}n_{l,x}}{2h_l} + \beta t_{2100}, \\
w_{cl5} &= -\frac{t_{1001}n_{l,x}}{h_m} + \frac{t_{2000}n_{l,y}}{2h_l} + \beta t_{2001}, \\
w_{dl5} &= -\frac{t_{1101}n_{l,x}}{h_m} + \frac{t_{2001}n_{l,x} + t_{2100}n_{l,y}}{2h_l} + \beta t_{2101}, \\
w_{el5} &= -\frac{t_{1200}n_{l,x}}{h_m} + \frac{t_{2100}n_{l,x}}{h_l} + \beta t_{2200}, \\
w_{fl5} &= -\frac{t_{1002}n_{l,x}}{h_m} + \frac{t_{2001}n_{l,y}}{h_l} + \beta t_{2002}, \\
w_{al6} &= -\frac{t_{0010}n_{l,y}}{h_m} + \beta t_{0020}, \\
w_{bl6} &= -\frac{t_{0110}n_{l,y}}{h_m} + \frac{t_{0020}n_{l,x}}{2h_l} + \beta t_{0120},
\end{aligned}$$

$$\begin{aligned}
w_{cl6} &= -\frac{t_{0011}n_{l,y}}{h_m} + \frac{t_{0020}n_{l,y}}{2h_l} + \beta t_{0021}, \\
w_{dl6} &= -\frac{t_{0111}n_{l,y}}{h_m} + \frac{t_{0021}n_{l,x} + t_{0120}n_{l,y}}{2h_l} + \beta t_{0121}, \\
w_{el6} &= -\frac{t_{0210}n_{l,y}}{h_m} + \frac{t_{0120}n_{l,x}}{h_l} + \beta t_{0220}, \\
w_{fl6} &= -\frac{t_{0012}n_{l,y}}{h_m} + \frac{t_{0021}n_{l,y}}{h_l} + \beta t_{0022}.
\end{aligned}$$

C) The Gaussian quadratures

The Gaussian quadrature for computing an integral $\int_{\Delta} f(x, y) dx dy$ on a triangle element Δ for the P^1 case has been given in the main text. Here we give the 7 points Gaussian quadrature for the P^2 case used in the computations of this paper. Suppose that the triangle element Δ has vertexes (x_i, y_i) , (x_j, y_j) and (x_k, y_k) , then the 7 Gaussian quadrature points are

$$\begin{aligned}
(x_1, y_1) &= (1/3 \cdot (x_i + x_j + x_k), 1/3 \cdot (y_i + y_j + y_k)), \\
(x_2, y_2) &= (ax_i + b(x_j + x_k), ay_i + b(y_j + y_k)), \\
(x_3, y_3) &= (ax_j + b(x_i + x_k), ay_j + b(y_i + y_k)), \\
(x_4, y_4) &= (ax_k + b(x_i + x_j), ay_k + b(y_i + y_j)), \\
(x_5, y_5) &= (cx_i + d(x_j + x_k), cy_i + d(y_j + y_k)), \\
(x_6, y_6) &= (cx_j + d(x_i + x_k), cy_j + d(y_i + y_k)), \\
(x_7, y_7) &= (cx_k + d(x_i + x_j), cy_k + d(y_i + y_j)),
\end{aligned}$$

where $a = 0.059715871789770$, $b = 0.470142064105115$, $c = 0.797426985353087$, $d = 0.101286507323456$. The Gaussian quadrature is

$$\begin{aligned}
\int_{\Delta} f(x, y) dx dy &\approx |\Delta| \{w_1 f(x_1, y_1) + w_2 [f(x_2, y_2) + f(x_3, y_3) + f(x_4, y_4)] \\
&\quad + w_3 [f(x_5, y_5) + f(x_6, y_6) + f(x_7, y_7)]\}, \tag{47}
\end{aligned}$$

where $|\Delta|$ is the area of the triangle element Δ , $w_1 = 0.225$, $w_2 = 0.132394152788506$, and $w_3 = 0.125939180544827$.

References

- [1] Adjerid, S. and Temimi, H., *A discontinuous Galerkin method for higher-order ordinary differential equations*. Computer Methods in Applied Mechanics and Engineering, 2007. 197: p. 202-218.
- [2] Arnold, D., *An interior penalty finite element method with discontinuous elements*. SIAM J. Numer. Anal., 1982. 19: p. 742-760.

- [3] Arnold, D., Brezzi, F., Cockburn, B., and Marini, L.D., *Unified analysis of discontinuous Galerkin methods for elliptic problems*. SIAM J. Numer. Anal., 2002. 39: p. 1749-1779.
- [4] Ascher, U., Ruuth, S., and Wetton, B., *Implicit-explicit methods for time-dependent PDE's*. SIAM J. Numer. Anal., 1995. 32: p. 797-823.
- [5] Ascher, U.M., Ruuth, S.J., and Spiteri, R.J., *Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations*. Applied Numerical Mathematics, 1997. 25: p. 151-167.
- [6] Atkins, H. and Shu, C.-W. *Analysis of the discontinuous Galerkin method applied to the diffusion operator*. in 14th AIAA Computational Fluid Dynamics Conference AIAA. 1999.
- [7] Baker, G., *Finite element methods for elliptic equations using nonconforming elements*. Math. Comp., 1977. 31: p. 45-59.
- [8] Bassi, F. and Rebay, S., *A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations*. Journal of Computational Physics, 1997. 131: p. 267-279.
- [9] Baumann, C.E. and Oden, J.T., *A discontinuous hp finite element method for convection-diffusion problems*. Computer Methods in Applied Mechanics and Engineering, 1999. 175: p. 311-341.
- [10] Beylkin, G., and Keiser, J.M., *On the adaptive numerical solution of nonlinear partial differential equations in wavelet bases*. J. Comput. Phys., 1997. 132: p. 233-259.
- [11] Beylkin, G., Keiser, J.M., and Vozovoi, L., *A new class of time discretization schemes for the solution of nonlinear PDEs*. J. Comput. Phys., 1998. 147: p. 362-387.
- [12] Biswas, R., Devine, K.D., and Flaherty, J., *Parallel, adaptive finite element methods for conservation laws*. Applied Numerical Mathematics, 1994. 14: p. 255-283.
- [13] Boyd, J.P., *Chebyshev and Fourier Spectral Methods*, Dover, New York, 2001.
- [14] Bourlioux, A., Layton, A.T., and Minion, M.L., *High-order multi-implicit spectral deferred correction methods for problems of reactive flow*. Journal of Computational Physics, 2003. 189: p. 651-675.
- [15] Brown, P.N. and Hindmarsh, A.C., *Matrix-free methods for stiff systems of ODEs*. SIAM J. Numer. Anal., 1986. 23: p. 610-638.
- [16] Canuto, C., Hussaini, M.Y., Quarteroni, A., and Zang, T.A., *Spectral Methods in Fluid Dynamics*. Springer Series in Computational Physics. 1988, Berlin: Springer-Verlag.
- [17] Castillo, P., *Performance of discontinuous Galerkin methods for elliptic problems*. SIAM J. Numer. Anal., 2002. 24(2): p. 524-547.
- [18] Cheng, Y. and Shu, C.-W., *A discontinuous Galerkin finite element method for time dependent partial differential equations with higher order derivatives*. Mathematics of Computation, v77, 2008, p.699-730.

- [19] Christlieb, A., Ong, B., and Qiu, J.-M., *Integral deferred correction methods constructed with high order Runge-Kutta integrators*. Mathematics of Computation, to appear.
- [20] Cockburn, B., *Discontinuous Galerkin methods for convection-dominated problems*, in *High-Order Methods for Computational Physics*, T.J. Barth and H. Deconinck, Editors, Lecture Notes in Computational Science and Engineering, v9, 1999, Springer. p. 69-224.
- [21] Cockburn, B., Hou, S., and Shu, C.-W., *The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case*. Mathematics of Computation, 1990. 54: p. 545-581.
- [22] Cockburn, B., Karniadakis, G., and Shu, C.-W., The development of discontinuous Galerkin methods, in *Discontinuous Galerkin Methods: Theory, Computation and Applications*, B. Cockburn, G. Karniadakis, and C.-W. Shu, Editors, Lecture Notes in Computational Science and Engineering, v11, 2000, Springer, Part I: Overview. p. 3-50.
- [23] Cockburn, B., Lin, S.-Y., and Shu, C.-W., *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems*. Journal of Computational Physics, 1989. 84: p. 90-113.
- [24] Cockburn, B. and Shu, C.-W., *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework*. Mathematics of Computation, 1989. 52: p. 411-435.
- [25] Cockburn, B. and Shu, C.-W., *The Runge-Kutta local projection P1-discontinuous Galerkin finite element method for scalar conservation laws*. Mathematical Modelling and Numerical Analysis, 1991. 25: p. 337-361.
- [26] Cockburn, B. and Shu, C.-W., *The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems*. Journal of Computational Physics, 1998. 141: p. 199-224.
- [27] Cockburn, B. and Shu, C.-W., *Runge-Kutta discontinuous Galerkin method for convection-dominated problems*. Journal of Scientific Computing, 2001. 16: p. 173-261.
- [28] Cockburn, B. and Shu, C.-W., *The local discontinuous Galerkin method for time-dependent convection-diffusion systems*. SIAM Journal on Numerical Analysis, 1998. 35: p. 2440-2463.
- [29] Cox, S.M., and Matthews, P.C., *Exponential time differencing for stiff systems*. J. Comput. Phys., 2002. 176: p. 430-455.
- [30] Du, Q. and Zhu, W., *Modified exponential time differencing schemes: analysis and applications*. BIT, Numer. Math., 2005. 45(2): p. 307-328.
- [31] Druskin, V., and Knizhnerman, L.A., *Two polynomial methods of calculating functions of symmetric matrices*. Comput. Math. Math. Phys., 1989. 29: p. 112-121.
- [32] Druskin, V.L., and Knizhnerman, L.A., *Krylov space approximations of eigenpairs and matrix functions in exact and computer arithmetic*. Numer. Linear Algebra Appl., 1995. 2: p. 205-217.

- [33] Druskin, V., Greenbaum, A., and Knizhnerman, L.A., *Using nonorthogonal Lanczos vectors in the computation of matrix functions*. SIAM J. Sci. Comp., 1998. 19: p. 38-54.
- [34] Dutt, A., Greengard, L., and Rokhlin, V., *Spectral deferred correction methods for ordinary differential equations*. BIT, 2000. 40(2): p. 241-266.
- [35] Fornberg, B. and Driscoll, T.A., *A fast spectral algorithm for nonlinear wave equations with linear dispersion*. Journal of Computational Physics, 1999. 155: p. 456-467.
- [36] Gallopoulos, E. and Saad, Y., *Efficient solution of parabolic equations by Krylov approximation methods*. SIAM J. Sci. Stat. Comput., 1992. 13(5): p. 1236-1264.
- [37] Gassner, G., Lorcher, F., and Munz, C.-D., *A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes*. Journal of Computational Physics, 2007. 224: p. 1049-1063.
- [38] Gear, C.W. and Saad, Y., *Iterative solution of linear equations in ODE codes*. SIAM J. Sci. Statist. Comput., 1983. 4: p. 583-601.
- [39] Georgoulis, E.H., Hall, E., and Houston, P., *Discontinuous Galerkin methods for advection-diffusion-reaction problems on anisotropically refined meshes*. SIAM Journal on Scientific Computing, 2007. 30: p. 246-271.
- [40] Gottlieb, S., and Shu, C.-W., *Total variation diminishing Runge-Kutta schemes*. Math. Comp., 1998. 67: p.73-85.
- [41] Gottlieb, S., Shu, C.-W., and Tadmor, E., *Strong stability preserving high order time discretization methods*. SIAM Review, 2001. 43: p. 89-112.
- [42] Higham, N. J., *The scaling and squaring method for the matrix exponential revisited*. SIAM Review, 2009. 51(4): p. 747-764.
- [43] Hochbruck, M., and Lubich, C., *On Krylov subspace approximations to the matrix exponential operator*. SIAM J. Numer. Anal., 1997. 34(5): p. 1911-1925.
- [44] Hochbruck, M., Lubich, C., and Selhofer, H., *Exponential integrators for large systems of differential equations*. SIAM J. Sci. Comp., 1998. 19(5): p. 1552-1574.
- [45] Hochbruck, M., and Ostermann, A., *Explicit exponential Runge-Kutta methods for semilinear parabolic problems*. SIAM J. Numer. Anal., 2005. 43(3): p. 1069-1090.
- [46] Houston, P., Schwab, C., and Suli, E., *Discontinuous hp-finite element methods for advection-diffusion-reaction problems*. SIAM J. Numer. Anal., 2002. 39(6): p. 2133-2163.
- [47] Huang, J., Jia, J., and Minion, M., *Arbitrary order Krylov deferred correction methods for differential algebraic equations*. J. Comp. Phys., 2007. 221(2): p. 739-760.
- [48] Kanevsky, A., Carpenter, M.H., Gottlieb, D., and Hesthaven, J.S., *Application of implicit-explicit high order Runge-Kutta methods to Discontinuous-Galerkin schemes*. J. Comp. Phys., 2007. 225(2): p. 1753-1781.
- [49] Karakashian, O.A. and Pascal, F., *Convergence of adaptive discontinuous Galerkin approximations of second-order elliptic problems*. SIAM J. Numer. Anal., 2007. 45(2): p. 641-665.

- [50] Kassam, A.-K., and Trefethen, L.N., *Fourth-order time stepping for stiff PDEs*. SIAM J. Sci. Comp., 2005. 26(4): p. 1214–1233.
- [51] Kennedy, C.A. and Carpenter, M.H., *Additive Runge-Kutta schemes for convection-diffusion-reaction equations*. Applied Numerical Mathematics, 2003. 44: p. 139-181.
- [52] Kirby, R.M. and Karniadakis, G.E., *Selecting the numerical flux in Discontinuous Galerkin methods for diffusion problems*. J. Sci. Comput., 2005. 22-23: p. 385-411.
- [53] Krogstad, S., *Generalized integrating factor methods for stiff PDEs*. J. Comp. Phys., 2005. 203: p. 72-88.
- [54] Lander, A., Nie, Q., and Wan, F., *Do morphogen gradients arise by diffusion?* Development Cell, 2002. 2(6): p. 785-796.
- [55] Lander, A., Nie, Q., Wan, F., and Zhang, Y.-T., *Localized ectopic expression of Dpp receptors in a Drosophila embryo*. Studies in Applied Mathematics, 2009. 123: p. 175-214.
- [56] Layton, A.T., and Minion, M.L., *Conservative multi-implicit spectral deferred correction methods for reacting gas dynamics*. J. Comp. Phys., 2004. 194(2): p. 697-715.
- [57] Layton, A.T., and Minion, M.L., *Implications of the choice of predictors for semi-implicit Picard integral deferred corrections methods*. Comm. Appl. Math. Comput. Sci., 2007. 1(2): p. 1-34.
- [58] Levy, D., Shu, C.-W., and Yan, J., *Local discontinuous Galerkin methods for nonlinear dispersive equations*. Journal of Computational Physics, 2004. 196: p. 751-772.
- [59] Liu, H. and Yan, J., *The direct discontinuous Galerkin (DDG) methods for diffusion problems*. SIAM Journal on Numerical Analysis, 2009. 47: 475-698.
- [60] Maday, Y., Patera, A.T., and Ronquist, E.M., *An operator-integration-factor splitting method for time-dependent problems: application to incompressible fluid flow*. Journal of Scientific Computing, 1990. 5: 263-292.
- [61] Minion, M.L., *Semi-implicit spectral deferred correction methods for ordinary differential equations*. Commun. Math. Sci., 2003. 1(3): p. 471-500.
- [62] Moler, C., and Van Loan, C., *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*. SIAM Review, 2003. 45: p. 3-49.
- [63] Moret, I., and Novati, P., *RD-rational approximations of the matrix exponential*. BIT Numerical Mathematics, 2004. 44(3): p. 595-615.
- [64] Najm, H.N., Wyckoff, P.S., and Knio, O.M., *A semi-implicit numerical scheme for reacting flow. I. Stiff chemistry*. J. Comput. Phys., 1998. 143: p. 381-402.
- [65] Nauts, A., and Wyatt, R.E., *New approach to many-state quantum dynamics: the recursive-residue-generation method*. Phys. Rev. Lett., 1983. 51: p. 2238-2241.
- [66] Nie, Q., Zhang, Y.-T., and Zhao, R., *Efficient semi-implicit schemes for stiff systems*. Journal of Computational Physics, 2006. 214: p. 521-537.
- [67] Nie, Q., Wan, F., Zhang, Y.-T., and Liu, X.-F., *Compact integration factor methods in high spatial dimensions*. Journal of Computational Physics, 2008. 227: p. 5238-5255.

- [68] Oden, J.T., Babuska, I., and Baumann, C.E., *A discontinuous hp finite element method for diffusion problems*. Journal of Computational Physics, 1998. 146: p. 491-519.
- [69] Park, T.J., and Light, J.C., *Unitary quantum time evolution by iterative Lanczos reduction*. J. Chem. Phys., 1986. 85: p. 5870-5876.
- [70] Reed, W.H. and Hill, T.R., *Triangular mesh methods for neutron transport equation*. Tech. Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.
- [71] Riviere, B., Wheeler, M.F., and Girault, V., *A priori error estimates for finite element methods based on discontinuous approximation spaces for elliptic problems*. SIAM J. Numer. Anal., 2001. 39(3): p. 902-931.
- [72] Romkes, A., Prudhomme, S., and Oden, J.T., *A posteriori error estimation for a new stabilized discontinuous Galerkin method*. Appl. Math. Lett., 2003. 16(4): p. 447-452.
- [73] Ruuth, S.J., *Implicit-explicit methods for reaction-diffusion problems in pattern formation*. Journal of Mathematical Biology, 1995. 34: p. 148-176.
- [74] Saad, Y., *Analysis of some Krylov subspace approximations to the matrix exponential operator*. SIAM J. Numer. Anal., 1992. 29(1): p. 209-228.
- [75] Saad, Y., *Iterative methods for sparse linear systems*. SIAM, 2003.
- [76] Shu, C.-W., *TVD time discretizations*. SIAM J. Sci. Stat. Comput., 1988. 9: p. 1073-1084.
- [77] Shu, C.-W., and Osher, S., *Efficient implementation of essentially non-oscillatory shock-capturing schemes*. J. Comput. Phys., 1988. 77: p. 439-471.
- [78] Stewart, D.E., and Leyk, T.S., *Error estimates for Krylov subspace approximations of matrix exponentials*. J. Comput. Appl. Math., 1996. 72: p. 359-369.
- [79] Tal-Ezer, H., *Spectral methods in time for parabolic problems*. SIAM J. Numer. Anal., 1989. 26: p. 1-11.
- [80] Trefethen, L.N., *Spectral Methods in Matlab*. SIAM, 2000.
- [81] Trefethen, L.N., and Bau, D., *Numerical Linear Algebra*, SIAM, 1997.
- [82] van Den Eshof, J., and Hochbruck, M., *Preconditioning Lanczos approximations to the matrix exponential*. SIAM J. Sci. Comp., 2006. 27(4): p. 1438-1457.
- [83] van Leer, B. and Nomura, S. *Discontinuous Galerkin for diffusion*. in *17th AIAA Computational Fluid Dynamics Conference*. June 6-9, 2005, AIAA paper 2005-5108.
- [84] Verwer, J.G., Sommeijer, B.P., and Hundsdorfer, W., *RKC time-stepping for advection-diffusion-reaction problems*. J. Comput. Phys., 2004. 201: p. 61-79.
- [85] Xia, Y., Xu, Y., and Shu, C.-W., *Efficient time discretization for local discontinuous Galerkin methods*. Discrete and Continuous Dynamical Systems - Series B, 2007. 8: p. 677-693.
- [86] Xu, Y. and Shu, C.-W., *Local discontinuous Galerkin methods for three classes of nonlinear wave equations*. Journal of Computational Mathematics, 2004. 22: p. 250-274.

- [87] Xu, Y. and Shu, C.-W., *Local discontinuous Galerkin methods for nonlinear Schrodinger equations*. Journal of Computational Physics, 2005. 205: p. 72-97.
- [88] Xu, Y. and Shu, C.-W., *Local discontinuous Galerkin methods for two classes of two dimensional nonlinear wave equations*. Physica D, 2005. 208: p. 21-58.
- [89] Xu, Y. and Shu, C.-W., *Local discontinuous Galerkin methods for the Kuramoto-Sivashinsky equations and the Ito-type coupled KdV equations*. Computer Methods in Applied Mechanics and Engineering, 2006. 195: p. 3430-3447.
- [90] Yan, J. and Shu, C.-W., *A local discontinuous Galerkin method for KdV type equations*. SIAM Journal on Numerical Analysis, 2002. 40: p. 769-791.
- [91] Yan, J. and Shu, C.-W., *Local discontinuous Galerkin methods for partial differential equations with higher order derivatives*. Journal of Scientific Computing, 2002. 17: p. 27-47.
- [92] Zhang, Y.-T., Lander, A., and Nie, Q., *Computational analysis of BMP gradients in dorsal-ventral patterning of the zebrafish embryo*. J. Theor. Biol., 2007. 248(4): p. 579-589.
- [93] Zhong, X., *Additive semi-implicit Runge-Kutta methods for computing high-speed nonequilibrium reactive flows*. Journal of Computational Physics, 1996. 128: p. 19-31.
- [94] Zhu, J., Zhang, Y.-T., Newman, S.A., and Alber, M., *Application of Discontinuous Galerkin Methods for reaction-diffusion systems in developmental biology*. Journal of Scientific Computing, 2009. 40: p.391-418.

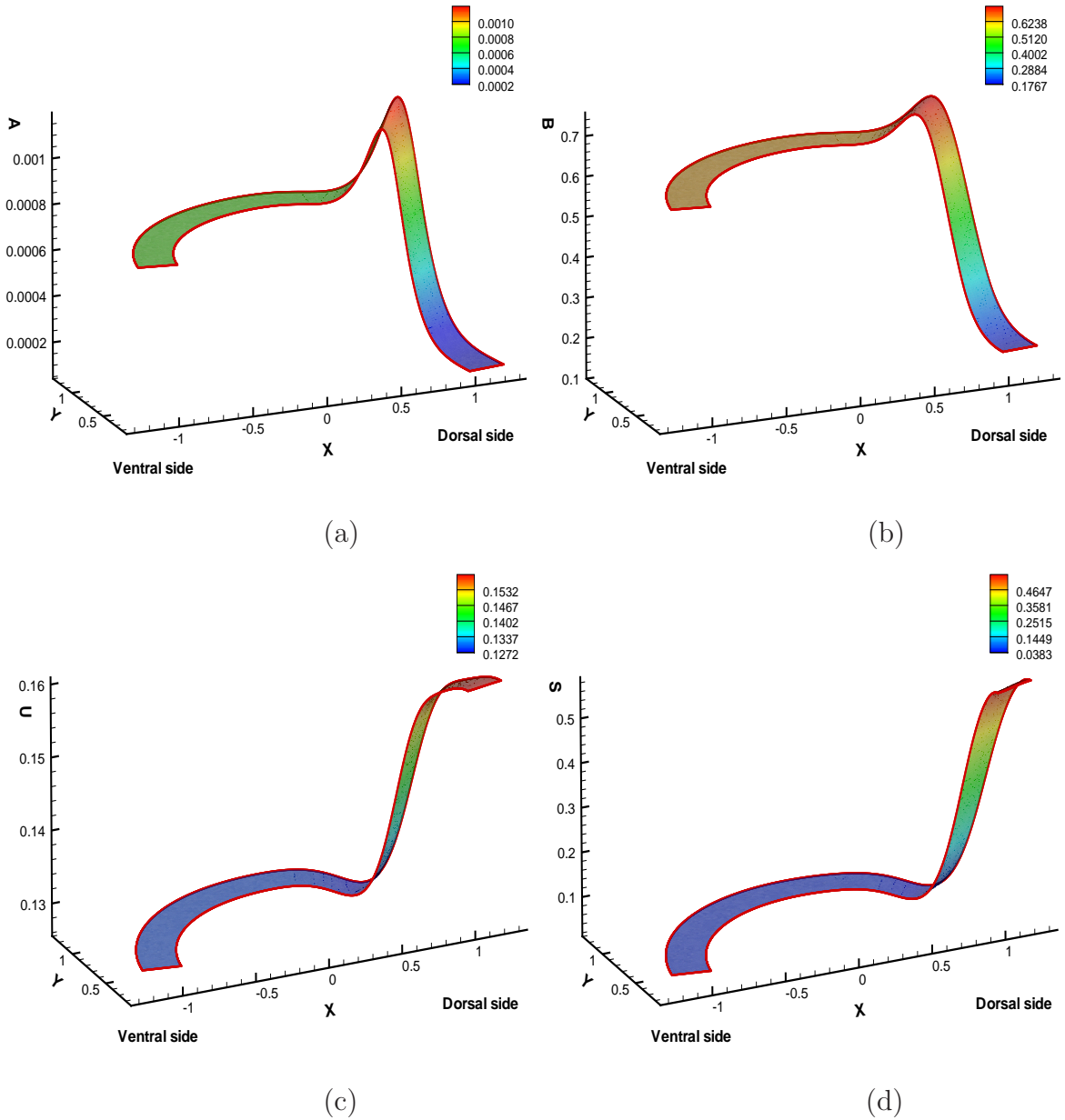


Figure 2: Simulation results of the normalized concentrations of morphogens for the zebrafish system. Normalized final time $T = 10$. The spatial triangular mesh has 154560 elements. $\Delta t = 0.5h_{min} = 6.416 \times 10^{-4}$. (a) A: the normalized concentration of molecule BMP; (b) B: the normalized concentration of BMP-receptor complexes; (c) U: the normalized concentration of BMP-Chordin complexes; (d) S: the normalized concentration of molecule Chordin.