

KRYLOV IMPLICIT INTEGRATION FACTOR METHOD FOR A CLASS OF STIFF REACTION-DIFFUSION SYSTEMS WITH MOVING BOUNDARIES

SHUANG LIU AND XINFENG LIU*

Department of Mathematics, University of South Carolina
Columbia, SC 29208, USA

(Communicated by Qing Nie)

ABSTRACT. The systems of reaction-diffusion equations coupled with moving boundaries defined by Stefan condition have been widely used to describe the dynamics of spreading population. There are several numerical difficulties to efficiently handle such systems. Firstly extremely small time steps are usually demanded due to the stiffness of the system. Secondly it is always difficult to efficiently and accurately handle the moving boundaries. To overcome these difficulties, we first transform the one-dimensional problem with a moving boundary into a system with a fixed computational domain, and then introduce four different temporal schemes: Runge-Kutta, Crank-Nicolson, implicit integration factor (IIF) and Krylov IIF for handling such stiff systems. Numerical examples are examined to illustrate the efficiency, accuracy and consistency for different approaches, and it can be shown that Krylov IIF is superior to other three approaches in terms of stability and efficiency by direct comparison.

1. Introduction. A moving boundary problem is characterized by the fact that the boundary of the domain is not known in advance but it has to be determined as a part of the solution. These problems are often called Stefan problems due to the Stefan condition that links the behavior of the boundary with the unknown solution [10, 39, 40].

According to the seminal paper by Du and Lin [12], where a Stefan condition is introduced and managing a moving boundary problem of parabolic type to describe the spreading of species population, the reaction-diffusion system for the density of population of the invasive species $U(t, x)$ depending on time t and spatial variable x in one dimension can be described as follows:

$$\frac{\partial U}{\partial t} - D \frac{\partial^2 U}{\partial x^2} = f(U), \quad t > 0, \quad 0 < x < H(t), \quad (1)$$

together with the boundary conditions

$$\frac{\partial U}{\partial x}(t, 0) = 0, \quad U(t, H(t)) = 0, \quad t > 0, \quad (2)$$

2010 *Mathematics Subject Classification.* Primary: 65N06, 65N40; Secondary: 92D25.

Key words and phrases. Implicit integration factor methods, Krylov subspace, moving boundaries, stiffness, Stefan problem.

* Corresponding author: Xinfeng Liu.

where the Stefan condition

$$H'(t) = -\mu \frac{\partial U}{\partial x}(t, H(t)), \quad t > 0, \quad (3)$$

and the initial conditions

$$H(0) = H_0, \quad U(0, x) = U_0(x), \quad 0 \leq x \leq H_0. \quad (4)$$

The initial function $U_0(x)$ satisfies the following properties:

$$U_0(x) \in C^2([0, H_0]), \quad U_0'(0) = U_0(H_0) = 0, \quad U_0(x) > 0, \quad 0 \leq x < H_0. \quad (5)$$

Here $H(t)$ is the unknown moving boundary such that the population is distributed in the interval $[0, H(t)]$, and $D > 0$ is the dispersal rate. The parameter $\mu > 0$ involved in the Stefan condition (3) is the proportionally constant between the population gradient at the front and the speed of the moving boundary.

The nonlinear function $f(U)$ is assumed to be a C^1 function satisfying $f(0) = 0$, and in the literature it is often taken to be the logistic function $f(U) = U(a - bU)$ with a, b positive constants. The positive parameters a and b are the intrinsic growth rate and the intra-specific competition, respectively. In the rest of this paper, we will take this logistic function as an example to demonstrate the numerical methods which can be easily extended to more general functions $f(U)$.

The moving boundary is generally called the “free boundary”. If $f(U) \equiv 0$, then this problem reduces to the classical Stefan problem, which has been extensively studied theoretically [4] and numerically [7, 15, 17, 5, 6, 37, 35] and the references therein. Other theoretical studies of related free boundary problems can be found in [2] and the references therein.

In contrast, very few numerical methods have been developed to solve such free boundary problems (1)-(5). When solving such system numerically, difficulties arise from the stiffness along with moving boundaries. Firstly it is always difficult to efficiently and accurately handle the moving boundaries. To efficiently handle the moving boundaries, level set methods [15, 36, 42, 43, 49, 50] and front tracking methods [27, 38, 48, 51] are two popular numerical approaches. One distinct feature of front tracking [9, 18, 21, 23, 28, 47] is using a pure Lagrangian approach to explicitly track locations of interfaces, but it is difficult to handle topological bifurcations in high dimensions, while the level set method can efficiently overcome such difficulties. In [29], a front-tracking framework and a front-fixing framework are introduced to solve the system (1)-(5) for 2D model with radial symmetry, and a level set approach is employed for general 2D model.

On the other hand, extremely small time steps are required due to the stiffness of the system. When the explicit schemes are applied to solve such a system, due to stability constraints, an extremely small time step should be used and it might take a long time to finish one single simulation. However, while applying an implicit scheme [3, 19, 31] may be able to remove the stability constraints on the time step Δt , it usually requires solving a large global system of nonlinear equations for each time step, and the computational cost could be significant. To efficiently solve stiff reaction-diffusion equations, the authors in [34] have developed a class of implicit integration factor (IIF) methods that are computationally much cheaper than fully implicit schemes and can be unconditionally stable or have generous stability conditions. The flexibility of representation of IIF method allows either direct calculation of the exponentials of matrices, or the use of Krylov subspace [8, 22, 24, 25, 30, 41] for non-constant diffusion coefficients or/with moving boundaries

to compute their exponential matrix-vector multiplications for further saving in storage and computational cost.

While very little existing work accounts for solving a stiff system with moving boundaries, this paper aims to develop and compare different numerical schemes to solve the system (1)-(5) in one dimension, and it can be observed that Krylov IIF is advantageous to other approaches in terms of stability and efficiency by direct comparison through numerical examples.

The rest of paper is organized as follows. In section 2, we first use the Landau transformation [10, 26] to convert the problem (1)-(5) into a system with fixed computational domain, in which the moving domain is included as a new separate variable coupled in the system. In section 2.1, the Crank-Nicolson scheme is applied to preserve second order accuracy both in time and space for better stability. In section 2.2, the Runge-Kutta method is introduced, which keeps second order accuracy both in time and space. In section 2.3, we extend the implicit integration factor (IIF) method to a system with moving boundaries, which successfully removes the stability constraint associated with the diffusion and the stiff reaction terms thus allowing for large time step size. In section 2.4, to calculate the matrix exponentials more efficiently, Krylov subspace incorporated with IIF is employed to speed up the simulation while maintaining the similar accuracy and stability conditions. In section 3, numerical examples are performed to show the accuracy, stability and efficiency of four different proposed approaches. Finally a brief conclusion is drawn.

2. Numerical schemes for stiff systems with moving boundaries. As discussed in [39], the system (1)-(5) in one dimension can be transforming into a problem with a fixed computational domain $[0, 1]$. For instance, Let us consider using the Landau transformation [10, 26],

$$z(t, x) = \frac{x}{H(t)}, \quad W(t, z) = U(t, x). \quad (6)$$

Then the 1D diffusive logistic model (1) turns into the form,

$$G(t) \frac{\partial W}{\partial t} - G'(t) \frac{z}{2} \frac{\partial W}{\partial z} - D \frac{\partial^2 W}{\partial z^2} = G(t)W(a - bW), \quad t > 0, \quad 0 < z < 1, \quad (7)$$

where

$$G(t) = H^2(t), \quad t \geq 0. \quad (8)$$

The boundary conditions (2) together with Stefan condition (3) take the following form

$$\frac{\partial W}{\partial z}(t, 0) = 0, \quad W(t, 1) = 0, \quad t > 0, \quad (9)$$

and

$$G'(t) = -2\mu \frac{\partial W}{\partial z}(t, 1), \quad t > 0, \quad (10)$$

respectively, while the initial conditions of (4) become

$$G(0) = H_0^2, \quad W(0, z) = W_0(z) = U_0(zH_0), \quad 0 \leq z \leq 1. \quad (11)$$

The smoothness of conditions of (5) for the initial function $U_0(x)$ are translated to $W_0(z)$ as follows,

$$W_0(z) \in C^2([0, 1]), \quad W_0'(0) = W_0(1) = 0, \quad W_0(z) > 0, \quad 0 \leq z < 1. \quad (12)$$

After the transformation, the new problem lies in solving the nonlinear system (7) in the fixed computational domain $[0, 1]$ for the new variable z and unknown functions W, G . To numerically solve the system, let us consider the time step size

$k = \Delta t$, and spatial mesh size $h = \Delta z = 1/M$, where M is a positive integer denoting the total number of intervals in $[0,1]$. The mesh points (t^n, z_j) are denoted with $t^n = kn$, $n \geq 0$, $z_j = jh$, $0 \leq j \leq M$. Let us denote w_j^n as the numerical approximation of $W(t^n, z_j)$ at the mesh point (t^n, z_j) , i.e., $w_j^n \approx W(t^n, z_j)$, and let g^n be the numerical value of $G(t^n)$.

2.1. Crank-Nicolson scheme for implicit temporal discretization. In this section, Crank-Nicolson scheme is applied to update G and W for each time step with the transformed system (7)-(10), which includes the following three steps:

Step 1. Evaluate $G(t)$ at t^{n+1} .

The new transformed Stefan condition (10) is discretized using first order forward approximation for $G'(t)$, and second order approximation of $\frac{\partial W}{\partial z}(t, 1)$ with three points backward,

$$g^n = -\frac{\mu}{h}(3w_M^n - 4w_{M-1}^n + w_{M-2}^n), \quad n > 0. \quad (13)$$

So $G(t^{n+1})$ can be evaluated by the following,

$$g^{n+1} = g^n + kg'^n, \quad n > 0. \quad (14)$$

Specially, we use $g^1 = g^0 + kg'(0)$ to evaluate the starting value g^1 .

Step 2. Update $W(t, z)$ at t^{n+1} by using Crank-Nicolson scheme.

Let us consider the forward approximation of the time derivative of W ,

$$\frac{w_j^{n+1} - w_j^n}{k} \approx \frac{\partial W}{\partial t}(t^n, z_j), \quad (15)$$

and the second-order central approximation of each spatial derivative of W ,

$$\frac{w_{j+1}^n - w_{j-1}^n}{2h} \approx \frac{\partial W}{\partial z}(t^n, z_j), \quad \frac{w_{j-1}^n - 2w_j^n + w_{j+1}^n}{h^2} \approx \frac{\partial^2 W}{\partial z^2}(t^n, z_j). \quad (16)$$

Based on the transformed Stefan condition (10) and from (13)-(14), $G'(t)$ can be evaluated by the following,

$$g'^{n+1} \approx -\frac{\mu}{h}(3w_M^{n+1} - 4w_{M-1}^{n+1} + w_{M-2}^{n+1}), \quad (17)$$

Putting (15)-(17) all together into equation (7), we obtain

$$\begin{aligned} \frac{w_j^{n+1} - w_j^n}{k} &= \frac{1}{2} \left(\frac{z_j}{2} \frac{w_{j+1}^n - w_{j-1}^n}{2h} \frac{g'^n}{g^n} + D \frac{w_{j-1}^n - 2w_j^n + w_{j+1}^n}{g^n h^2} \right) \\ &+ \frac{1}{2} \left(\frac{z_j}{2} \frac{w_{j+1}^{n+1} - w_{j-1}^{n+1}}{2h} \frac{g'^{n+1}}{g^n + kg'^n} + D \frac{w_{j-1}^{n+1} - 2w_j^{n+1} + w_{j+1}^{n+1}}{g^{n+1} h^2} \right) \\ &+ \frac{1}{2} (w_j^n (a - bw_j^n) + w_j^{n+1} (a - bw_j^{n+1})). \end{aligned} \quad (18)$$

where $n \geq 0$, $0 \leq j \leq M-1$.

For $j = 0$, Equation (18) involves the fictitious value w_{-1}^n at the ghost point $(t^n, -h)$. The value w_{-1}^n can be evaluated from the discretization of the boundary condition (9),

$$\frac{w_1^n - w_{-1}^n}{2h} = 0.$$

Such a nonlinear system can be solved by Newton iteration or other solvers.

Step 3. Reevaluate $G(t)$ at t^{n+1} using *BDF2*.

$$g^{n+1} = \frac{4}{3}g^n - \frac{1}{3}g^{n-1} - \frac{2\mu k}{3h}(3w_M^{n+1} - 4w_{M-1}^{n+1} + w_{M-2}^{n+1}), \quad n \geq 1. \quad (19)$$

We use BDF2 scheme to reevaluate $G(t)$ at t^{n+1} for $n = 1, 2, 3, \dots$ in order to preserve the second order accuracy with $O(k^2) + O(h^2)$. When $n = 0$, we use g^1 as a starting value. The same process follows to select starting values.

Remark 1. Consistency and error analysis Let us consider the problems (6)-(12), and denote the Equations (7), (9) and (10) in the following form,

$$\mathcal{L}_1(W, G) = \frac{\partial W}{\partial t} - \frac{G'(t)z}{G(t)} \frac{\partial W}{\partial z} - \frac{D}{G(t)} \frac{\partial^2 W}{\partial z^2} - W(a - bW) = 0, \\ t > 0, \quad 0 < z < 1,$$

$$\mathcal{L}_2(W, G) = \frac{\partial W}{\partial z}(t, 0) = 0, \quad t > 0,$$

$$\mathcal{L}_3(W, G) = G'(t) + 2\mu \frac{\partial W}{\partial z}(t, 1) = 0, \quad t > 0,$$

and after discretization, the equations are approximated by

$$L_1(w, g) = \frac{w_j^{n+1} - w_j^n}{k} - \frac{1}{2} \left(\frac{z_j}{2} \frac{w_{j+1}^n - w_{j-1}^n}{2h} \frac{g^n}{g^n} + D \frac{w_{j-1}^n - 2w_j^n + w_{j+1}^n}{g^n h^2} \right) \\ - \frac{1}{2} \left(\frac{z_j}{2} \frac{w_{j+1}^{n+1} - w_{j-1}^{n+1}}{2h} \frac{g^{n+1}}{g^{n+1}} + D \frac{w_{j-1}^{n+1} - 2w_j^{n+1} + w_{j+1}^{n+1}}{g^{n+1} h^2} \right) \\ - \frac{1}{2} (w_j^n (a - bw_j^n) + w_j^{n+1} (a - bw_j^{n+1})) = 0, \quad n > 0, \quad 0 < j \leq M-1,$$

$$L_2(w, g) = \frac{w_1^n - w_{-1}^n}{2h} = 0, \quad n > 0,$$

$$L_3(w, g) = \frac{g^{n+1} - g^n}{k} + \frac{\mu}{h} (3w_M^n - 4w_{M-1}^n + w_{M-2}^n) = 0, \quad n > 0.$$

Denoting the local truncation error $T_j^n(W, G)$ as

$$T(1)_j^n(W, G) = L_1(W_j^n, G^n) - \mathcal{L}_1(W_j^n, G^n),$$

$$T(2)_j^n(W, G) = L_2(W_j^n, G^n) - \mathcal{L}_2(W_j^n, G^n),$$

$$T(3)_j^n(W, G) = L_3(W_j^n, G^n) - \mathcal{L}_3(W_j^n, G^n),$$

Suppose we are given exact solutions W and G of problems (6)-(12) at t^n , here $W_j^n = W(t^n, z_j)$ and $G^n = G(t^n)$. According to [44], if the local truncation error $T_j^n(W, G) = (T(1)_j^n, T(2)_j^n, T(3)_j^n)$ tend to zero as $k \rightarrow 0$, $h \rightarrow 0$, we say that the scheme $L(w, g) = (L_1(w, g), L_2(w, g), L_3(w, g))$ is consistent with problem $\mathcal{L}(W, G) = (\mathcal{L}_1(W, G), \mathcal{L}_2(W, G), \mathcal{L}_3(W, G))$. With some calculation, we can get

$$T(1)_j^n(W, G) = \frac{k}{2} \frac{\partial^2 W}{\partial t^2}(t^n, z_j) + \frac{k^2}{6} \frac{\partial^3 W}{\partial t^3}(\tau_1, z_j) \\ + \frac{1}{2} \frac{\partial W}{\partial t}(t^n, z_j) + J_1 - \frac{1}{2} \frac{\partial W}{\partial t}(t^{n+1}, z_j) + J_2, \\ = \frac{k}{2} \frac{\partial^2 W}{\partial t^2}(t^n, z_j) + \frac{k^2}{6} \frac{\partial^3 W}{\partial t^3}(\tau_1, z_j) + \frac{1}{2} \frac{\partial W}{\partial t}(t^n, z_j) + J_1$$

$$\begin{aligned}
& -\frac{1}{2}\left(\frac{\partial W}{\partial t}(t^n, z_j) + k\frac{\partial^2 W}{\partial t^2}(t^n, z_j) + \frac{k^2}{2}\frac{\partial^3 W}{\partial t^3}(\tau_2, z_j)\right) + J_2, \\
& = \frac{k^2}{6}\frac{\partial^3 W}{\partial t^3}(\tau_1, z_j) - \frac{k^2}{4}\frac{\partial^3 W}{\partial t^3}(\tau_2, z_j) + J_1 + J_2.
\end{aligned}$$

J_1 and J_2 are defined as

$$\begin{aligned}
J_1 &= -\frac{z_j}{4}\frac{h^2}{6}\frac{\partial^3 W}{\partial z^3}(t^n, \delta_1)\frac{G'(t^n)}{G^n} - \frac{D}{2G^n}\frac{h^2}{12}\frac{\partial^4 W}{\partial z^4}(t^n, \delta_2) \\
&\quad - \frac{z_j}{4}\frac{\partial W}{\partial z}(t^n, z_j)\frac{2\mu h^2}{3G^n}\frac{\partial^3 W}{\partial z^3}(t^n, \delta_1) + \mathcal{O}(h^4). \\
J_2 &= -\frac{z_j}{4}\frac{\partial W}{\partial z}(t^{n+1}, z_j)\frac{G'(t^{n+1})}{G(t^{n+1})}\frac{k^2 G''(\tau_3)}{2G(t^{n+1})} \\
&\quad - \frac{z_j}{4}\frac{\partial W}{\partial z}(t^{n+1}, z_j)\frac{2\mu h^2}{3G(t^{n+1})}\frac{\partial^3 W}{\partial z^3}(t^{n+1}, \delta_4) \\
&\quad - \frac{z_j}{4}\frac{\partial W}{\partial z}(t^{n+1}, z_j)\frac{2\mu h^2}{3G(t^{n+1})}\frac{\partial^3 W}{\partial z^3}(t^{n+1}, \delta_4)\frac{k^2 G''(\tau_3)}{2G(t^{n+1})} \\
&\quad - \frac{z_j}{4}\frac{h^2}{6}\frac{\partial^3 W}{\partial z^3}(t^{n+1}, \delta_3)\frac{G'(t^{n+1})}{G(t^{n+1})} \\
&\quad - \frac{z_j}{4}\frac{h^2}{6}\frac{\partial^3 W}{\partial z^3}(t^{n+1}, \delta_3)\frac{G'(t^{n+1})}{G(t^{n+1})}\frac{k^2 G''(\tau_3)}{2G(t^{n+1})} \\
&\quad - \frac{D}{2}\frac{\partial^2 W}{\partial z^2}(t^{n+1}, z_j)\frac{k^2 G''(\tau_3)}{2G(t^{n+1})} - \frac{D}{2}\frac{h^2}{12G(t^{n+1})}\frac{\partial^4 W}{\partial z^4}(t^{n+1}, \delta_5) \\
&\quad - \frac{D}{2}\frac{h^2}{12G(t^{n+1})}\frac{\partial^4 W}{\partial z^4}(t^{n+1}, \delta_5)\frac{k^2 G''(\tau_3)}{2G(t^{n+1})} + \mathcal{O}(h^4) + \mathcal{O}(k^4).
\end{aligned}$$

$$T(2)_j^n(W, G) = \frac{h^2}{2}\frac{\partial^3 W}{\partial z^3}(t^n, \delta_6).$$

$$T(3)_j^n(W, G) = -\frac{k^2}{3}\frac{d^3 G}{dt^3}(\tau_4, 1) - \frac{2\mu h^2}{3}\frac{\partial^3 W}{\partial z^3}(t^n, \delta_7).$$

where the introduced parameters are given by

$$\begin{aligned}
t^n &\leq \tau_1, \tau_2, \tau_3 \leq t^{n+1}, \\
t^{n-1} &\leq \tau_4 \leq t^{n+1}, \\
z_{j-1} &\leq \delta_1, \delta_2, \delta_3, \delta_5 \leq z_{j+1}, \\
z_{M-2} &\leq \delta_4, \delta_7 \leq 1. \\
z_{-1} &\leq \delta_6 \leq z_1.
\end{aligned}$$

By assuming that $W(t, z)$ is fourth-order partial differentiable with respect to z and third-order differentiable with respect to t , and that $G(t)$ is third-order differentiable, we can find that the local truncation error satisfies

$$T(i)_j^n(W, G) = \mathcal{O}(k^2) + \mathcal{O}(h^2), \quad i = 1, 2, 3.$$

2.2. Runge-Kutta method for explicit temporal discretization. In this section, we introduce the Runge-Kutta scheme to solve the transformed system (7)-(12). For this approach, the evaluation of W, G from t^n to t^{n+1} consists of the following five steps:

Step 1. Evaluate $G(t^{n+1/2})$.

The transformed Stefan condition (10) is discretized by using forward approximation in time for $G'(t)$ and three points backward approximation in space of $\frac{\partial W}{\partial z}(t, 1)$, i.e.,

$$g'^n = -\frac{\mu}{h}(3w_M^n - 4w_{M-1}^n + w_{M-2}^n), \quad n > 0. \quad (20)$$

Thus $G(t^{n+1/2})$ can be evaluated by the following,

$$g^{n+1/2} = g^n + \frac{k}{2}g'^n, \quad n > 0. \quad (21)$$

Step 2. Update $W(t, z)$ at $t^{n+1/2}$.

Let us consider the forward approximation of all time derivatives,

$$\frac{w_j^{n+1/2} - w_j^n}{k/2} \approx \frac{\partial W}{\partial t}(t^n, z_j), \quad \frac{g^{n+1/2} - g^n}{k/2} \approx G'(t^n), \quad (22)$$

and the second-order central approximation in space,

$$\frac{w_{j+1}^n - w_{j-1}^n}{2h} \approx \frac{\partial W}{\partial z}(t^n, z_j), \quad \frac{w_{j-1}^n - 2w_j^n + w_{j+1}^n}{h^2} \approx \frac{\partial^2 W}{\partial z^2}(t^n, z_j). \quad (23)$$

Combining (20)-(23) together, the equation (7) can be discretized by

$$\frac{w_j^{n+1/2} - w_j^n}{k/2} - \frac{z_j}{2} \frac{w_{j+1}^n - w_{j-1}^n}{2h} \frac{g'^n}{g^n} - D \frac{w_{j-1}^n - 2w_j^n + w_{j+1}^n}{g^n h^2} = w_j^n (a - bw_j^n).$$

where $n \geq 0$, $0 \leq j \leq M-1$.

Step 3. Repeat Step 1 to evaluate $G(t)$ at t^{n+1} using the updated W and G at $t^{n+1/2}$.

Step 4. Evaluate $W(t, z)$ at t^{n+1} using the Midpoint Method.

$$\begin{aligned} g'^{n+1/2} &= -\frac{\mu}{h}(3w_M^{n+1/2} - 4w_{M-1}^{n+1/2} + w_{M-2}^{n+1/2}), \quad n > 0. \\ \frac{w_j^{n+1} - w_j^n}{k} - \frac{z_j}{2} \frac{w_{j+1}^{n+1/2} - w_{j-1}^{n+1/2}}{2g^{n+1/2}h} g'^{n+1/2} - D \frac{w_{j-1}^{n+1/2} - 2w_j^{n+1/2} + w_{j+1}^{n+1/2}}{g^{n+1/2}h^2} \\ &= w_j^{n+1/2} (a - bw_j^{n+1/2}), \quad n \geq 0, \quad 0 \leq j \leq M-1. \end{aligned} \quad (24)$$

i.e.,

$$\begin{aligned} w_j^{n+1} &= w_j^n + \frac{k}{g^{n+1/2}} \left(\frac{z_j}{2} \frac{w_{j+1}^{n+1/2} - w_{j-1}^{n+1/2}}{2h} g'^{n+1/2} \right) \\ &\quad + kD \frac{w_{j-1}^{n+1/2} - 2w_j^{n+1/2} + w_{j+1}^{n+1/2}}{g^{n+1/2}h^2} + kw_j^{n+1/2} (a - bw_j^{n+1/2}). \end{aligned}$$

Step 5. Reevaluate $G(t)$ at t^{n+1} using backward differentiation formula (BDF2).

$$g^{n+1} = \frac{4}{3}g^n - \frac{1}{3}g^{n-1} - \frac{2\mu k}{3h}(3w_M^{n+1} - 4w_{M-1}^{n+1} + w_{M-2}^{n+1}), \quad n \geq 1. \quad (25)$$

Remark 2. Similar to Crank-Nicolson scheme, we have also performed consistency and error analysis to the Runge-Kutta scheme, which exhibits the second order accuracy in both space and time $O(k^2) + O(h^2)$. We omit the details of derivation here to keep the paper more focused and concise.

2.3. Implicit integration factor (IIF) method. In this section, the implicit integration factor (IIF) scheme is extended to solve the transformed Stefan problem (7)-(10). According to the equation (7),

$$\frac{\partial W}{\partial t}(t, z) = \frac{G'(t)}{G(t)} \frac{z}{2} \frac{\partial W}{\partial z}(t, z) + \frac{D}{G(t)} \frac{\partial^2 W}{\partial z^2}(t, z) + W(t, z)(a - bW(t, z)), \quad 0 \leq z < 1, \quad (26)$$

and notice that $W(t, 1) \equiv 0$ from the boundary condition.

First let us consider the central approximation of the spatial derivatives,

$$\begin{aligned} \frac{W(t, z_{j+1}) - W(t, z_{j-1})}{2h} &\approx \frac{\partial W}{\partial z}(t, z_j), \quad 0 \leq j \leq M-1, \\ \frac{W(t, z_{j-1}) - 2W(t, z_j) + W(t, z_{j+1}))}{h^2} &\approx \frac{\partial^2 W}{\partial z^2}(t, z_j), \quad 0 \leq j \leq M-1. \end{aligned}$$

Putting this into the right hand side (RHS) of (26), and we obtain a semi-discretized ODE system

$$\frac{d\vec{W}}{dt} = \mathcal{C}(t)\vec{W} + \mathcal{R}(\vec{W}). \quad (27)$$

Here $\vec{W} = (W(t, z_0); W(t, z_1); W(t, z_2); \dots; W(t, z_{M-1}))$, and $\mathcal{C}(t)$ is the approximation matrix for the spatial derivatives, $\mathcal{R}(\vec{W})$ is the nonlinear reaction term. For instance, $\mathcal{C}(t)$ at $t = t^n$ denoted by \mathcal{C}_n with size $M \times M$ is given by

$$\begin{bmatrix} \frac{-2D}{g^n h^2} & \frac{2D}{g^n h^2} & 0 & \dots & 0 & 0 \\ \frac{D}{g^n h^2} - \frac{z_1 g'^n}{4hg^n} & \frac{-2D}{g^n h^2} & \frac{D}{g^n h^2} + \frac{z_1 g'^n}{4hg^n} & \dots & \dots & 0 \\ 0 & \frac{D}{g^n h^2} - \frac{z_2 g'^n}{4hg^n} & \frac{-2D}{g^n h^2} & \frac{D}{g^n h^2} + \frac{z_2 g'^n}{4hg^n} & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & \dots & \dots & 0 & \frac{D}{g^n h^2} - \frac{z_{M-1} g'^n}{4hg^n} & \frac{-2D}{g^n h^2} \end{bmatrix}$$

Again based on the transformed Stefan condition (10), g'^n in the matrix \mathcal{C}_n can be evaluated by the second order approximation of $\frac{\partial W}{\partial z}(t^n, 1)$ with three points,

$$-\frac{\mu}{h}(3w_M^n - 4w_{M-1}^n + w_{M-2}^n) \approx g'^n, \quad (28)$$

Since $\mathcal{C}(t)$ is nonlinear, we can rewrite $\mathcal{C}(t)$ as

$$\mathcal{C}(t) = \mathcal{C}(t^n) + \mathcal{E}(t), \quad (29)$$

where $\mathcal{E}(t)$ is the correction term [24]. Thus we obtain

$$\frac{d\vec{W}}{dt} = \mathcal{C}_n \vec{W} + \mathcal{E}(t)\vec{W} + \mathcal{R}(\vec{W}). \quad (30)$$

Multiplying by the integration factor $e^{-\mathcal{C}_n t}$ on both sides, we get

$$e^{-\mathcal{C}_n t} \frac{d\vec{W}}{dt} = e^{-\mathcal{C}_n t} (\mathcal{C}_n \vec{W} + \mathcal{E}(t)\vec{W} + \mathcal{R}(\vec{W})). \quad (31)$$

Let \vec{W}_{n+i} denote as the numerical approximation for $\vec{W}(t^{n+i})$. After integrating it over one time step from t^n to $t^{n+1} \equiv t^n + \Delta t$, we have

$$\vec{W}_{n+1} = e^{C_n \Delta t} \vec{W}_n + e^{C_n \Delta t} \int_0^{\Delta t} e^{-C_n \tau} (\mathcal{E}(t^n + \tau) \vec{W}(t^n + \tau) + \mathcal{R}(\vec{W}(t^n + \tau))) d\tau. \quad (32)$$

Following the same ideas as proposed in [34], to evaluate the integral, we approximate the nonlinear term $e^{-C_n \tau} (\mathcal{E}(t^n + \tau) \vec{W}(t^n + \tau) + \mathcal{R}(\vec{W}(t^n + \tau)))$ by an $(r-1)$ th order Lagrange polynomial $p(\tau)$ with interpolation points at $t^{n+1}, t^n, \dots, t^{n+2-r}$ for r -th order scheme. If we denote $\mathcal{F}(\vec{W}(t^n + \tau)) = \mathcal{E}(t^n + \tau) \vec{W}(t^n + \tau) + \mathcal{R}(\vec{W}(t^n + \tau))$, we obtain

$$p(\tau) = \sum_{i=-1}^{r-2} e^{i C_n \Delta t} \mathcal{F}(\vec{W}_{n-i}) \prod_{j=-1, j \neq i}^{r-2} \frac{\tau + j \Delta t}{(j - i) \Delta t}, \quad 0 \leq \tau \leq \Delta t. \quad (33)$$

Then we can get

$$\int_0^{\Delta t} e^{-C_n \tau} \mathcal{F}(\vec{W}(t^n + \tau)) d\tau \approx \sum_{i=-1}^{r-2} e^{i C_n \Delta t} \mathcal{F}(\vec{W}_{n-i}) \int_0^{\Delta t} \prod_{j=-1, j \neq i}^{r-2} \frac{\tau + j \Delta t}{(j - i) \Delta t} d\tau.$$

Putting all these together, and finally the r -th order IIF scheme is given by

$$\vec{W}_{n+1} = e^{C_n \Delta t} \vec{W}_n + \Delta t (\alpha_{n+1} \mathcal{F}(\vec{W}_{n+1}) + \sum_{i=0}^{r-2} \alpha_{n-i} \mathcal{F}(\vec{W}_{n-i})) \quad (34)$$

with the coefficients

$$\alpha_{n-i} = \frac{e^{(i+1)C_n \Delta t}}{\Delta t} \int_0^{\Delta t} \prod_{j=-1, j \neq i}^{r-2} \frac{\tau + j \Delta t}{(j - i) \Delta t} d\tau, \quad -1 \leq i \leq r-2. \quad (35)$$

More specially, the second order scheme (IIF2) is of the following form

$$\vec{W}_{n+1} = e^{C_n \Delta t} \vec{W}_n + \Delta t (\alpha_{n+1} \mathcal{F}(\vec{W}_{n+1}) + \alpha_n \mathcal{F}(\vec{W}_n)), \quad (36)$$

where

$$\alpha_n = \frac{1}{2} e^{C_n \Delta t}, \quad \alpha_{n+1} = \frac{1}{2}.$$

After evaluating \vec{W} at t^{n+1} , we reevaluate $G(t)$ at t^{n+1} using *BDF2* to preserve the second order accuracy for the front G ,

$$g^{n+1} = \frac{4}{3} g^n - \frac{1}{3} g^{n-1} - \frac{2\mu k}{3h} (3w_M^{n+1} - 4w_{M-1}^{n+1} + w_{M-2}^{n+1}), \quad n \geq 1. \quad (37)$$

Remark 3. The major computational cost of IIF arises from the evaluation of exponential matrices and multiplication of matrix with vectors. Since C_n is not a constant matrix, its exponential $e^{C_n \Delta t}$ needs to be computed at every time step that is extremely expensive thus will significantly slow down the simulation. Hence computation of exponential matrices at each time step should be avoided. Otherwise, it will become a bottleneck of regular IIF method for solving such system.

2.4. Krylov subspace with IIF. As discussed before [8], it is extremely expensive to construct matrix \mathcal{C}_n and evaluate $e^{\mathcal{C}_n \Delta t} \vec{W}_n$ for each time step. Although the matrix \mathcal{C}_n is sparse for most cases, the exponential matrix $e^{\mathcal{C}_n \Delta t}$ is dense in general. To overcome this difficulty, applying Krylov subspace for evaluation of the type of $e^{\mathbf{A} \Delta t} v$ is an excellent option. For example, in [16, 41], 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. Furthermore, Krylov subspace has been successfully incorporated with IIF and sparse grid structure for solving systems of PDEs with non-constant coefficient and unstructured grid meshes among many other systems [8, 24, 25, 30].

Following the literature (e.g. [16, 32]), we next illustrate how to apply the Krylov subspace to evaluate $e^{\mathbf{A} \Delta t} v$ in general. First the large sparse matrix \mathbf{A} can be projected to the Krylov subspace

$$K_Q = \text{span}\{v, \mathbf{A}v, \mathbf{A}^2v, \dots, \mathbf{A}^{Q-1}v\}. \quad (38)$$

The dimension Q of the Krylov subspace is usually **much** smaller than the dimension of the large sparse matrix \mathbf{A} . For instance, we take $Q = 25$ for most of our test simulations in the next section. An orthonormal basis $V_Q = [v_1, v_2, v_3, \dots, v_Q]$ of the Krylov subspace K_Q is generated by the well-known Arnoldi algorithm [46] as the following,

1. Compute the initial vector: $v_1 = v / \|v\|_2$.
2. Perform iterations: Do $j = 1, 2, \dots, Q$:
 - (a) Compute the vector $w = \mathbf{A}v_j$.
 - (b) Do $i = 1, 2, \dots, j$:
 - (i) Compute the inner product $h_{i,j} = (w, v_i)$.
 - (ii) Compute the vector $w = w - h_{i,j}v_i$.
 - (c) Compute $h_{j+1,j} = \|w\|_2$.
 - (d) 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 $j < Q$, it means that the convergence has already occurred and the Krylov subspace is equal to $\text{span}\{v_1, v_2, \dots, v_j\}$. Hence the iteration can be stopped at this step j , and we assign the value of this j to Q . This algorithm will produce an orthonormal basis V_Q of the Krylov subspace K_Q . Denote the $Q \times Q$ upper Hessenberg matrix consisting of the coefficients $h_{i,j}$ by H_Q . Since the columns of V_Q are orthogonal, we have

$$H_Q = V_Q^T \mathbf{A} V_Q \quad (39)$$

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

$$e^{\mathbf{A} \Delta t} v \approx V_Q V_Q^T e^{\mathbf{A} \Delta t} v = \beta V_Q V_Q^T e^{\mathbf{A} \Delta t} v_1 = \beta V_Q V_Q^T e^{\mathbf{A} \Delta t} V_Q e_1 \quad (40)$$

where $\beta = \|v\|_2$, and e_1 denotes the first column of the $Q \times Q$ identity matrix I_Q . Noting from (39), we have the following approximation

$$e^{\mathbf{A} \Delta t} v \approx \beta V_Q e^{H_Q \Delta t} e_1 \quad (41)$$

Thus $e^{\mathbf{A}\Delta t}v$ for the matrix exponential with multiplication of a vector problem can be solved by a problem with much smaller size. The small matrix exponential $e^{H_Q\Delta t}$ will be computed using a scaling and squaring algorithm with a *Padé* approximation with computational cost in the order of $\mathcal{O}(Q^2)$, see [16, 20, 32].

Such ideas of Krylov subspace approximation can be easily applied to IIF scheme. For instance, the second order Krylov IIF scheme gives

$$\vec{W}_{n+1} = \frac{1}{2}\Delta t\mathcal{F}(\vec{W}_{n+1}) + \beta_n\vec{W}_{Q,n}e^{H_{Q,n}\Delta t}e_1 \tag{42}$$

where $\beta_n = \|\vec{W}_n + \frac{1}{2}\Delta t\mathcal{F}(\vec{W}_n)\|_2$, $\vec{W}_{Q,n}$ and $H_{Q,n}$ are orthonormal basis and upper Hessenberg matrix generated by the Arnoldi algorithm with the initial vector $\vec{W}_n + \frac{1}{2}\Delta t\mathcal{F}(\vec{W}_n)$.

3. Numerical experiments. In this section, we will investigate the accuracy, stability and efficiency of the proposed four approaches: Runge-Kutta, Crank-Nicolson, IIF2 and Krylov IIF2, through a number of test examples with the following from,

$$\begin{cases} \frac{\partial U}{\partial t} - D\frac{\partial^2 U}{\partial x^2} = U(a - bU), & t > 0, \quad 0 < x < H(t), \\ \frac{\partial U}{\partial x}(t, 0) = 0, \quad U(t, H(t)) = 0, & t > 0, \\ H'(t) = -\mu\frac{\partial U}{\partial x}(t, H(t)), & t > 0, \\ H(0) = H_0, \quad U(0, x) = U_0(x), & 0 \leq x \leq H_0. \end{cases} \tag{43}$$

First we test the convergence order for all four methods in both space and time, in which the second order is observed for all cases. Next we show that the three schemes: Crank-Nicolson, IIF2 and Krylov IIF2 are much more stable than Runge-Kutta method. Finally Krylov IIF2 is shown to be able to dramatically reduce computational cost with comparison to Crank-Nicolson and IIF2.

3.1. Accuracy test. Here we test the accuracy of the proposed Runge-Kutta scheme, Crank-Nicolson scheme, IIF2 scheme and Krylov IIF2 scheme in space and time by a simple case with $D = a = b = \mu = 1, H_0 = 0.2, U_0(x) = \cos(\frac{x\pi}{2H_0})$ in the system (43).

Since the exact solution is not known for this problem (43), for all our simulations, the numerical solution by using a very fine resolution will be considered as reference or “exact” solution. The differences between the numerical solutions and the “exact” solutions at the final time T are measured by both L_2 and L_∞ errors. Here we set the final time $T = 0.1$. We also compare all the “exact solution” from four different schemes to make sure that they are consistent with each other.

In Tables (1 - 4), we run the numerical experiments at six different spatial and temporal resolutions. As expected, we can clearly see the second-order convergence rate in both space and time for all four schemes: Runge-Kutta, Crank-Nicolson, IIF2 and Krylov IIF2.

3.2. Stability test. In this section, we test the stability of the proposed four methods: Runge-Kutta scheme, Crank-Nicolson scheme, IIF2 scheme and Krylov IIF2 scheme by the system (43) with parameters $D = 1, \mu = 0.5, a = 2, b = 1, H_0 = 1.2, U_0(x) = \cos(\frac{x\pi}{2H_0})$. For all simulations here, we set the final simulation time to $T_{\text{end}} = 0.2$, and the spatial grid $N_z = 801$ while varying time step size. The reference solution is calculated by choosing $N_z = 801$ and $\Delta t = 10^{-8}$, and the maximum error is measured between the numerical solution and reference solution.

In Figures (1) and (2), we can see that Runge-Kutta only converges for very small time step size, it quickly blows up as Δt increases, while all other three schemes:

$N_z \times N_t$	$L_\infty Error$	Order	$L_2 Error$	Order
Accuracy test of W				
$26 \times 5e4$	1.85e-04		1.32e-04	
$51 \times 1e5$	4.62e-05	2.00	3.28e-05	2.01
$101 \times 2e5$	1.16e-05	1.99	8.22e-06	2.00
$201 \times 4e5$	2.89e-06	2.01	2.04e-06	2.01
$401 \times 8e5$	6.39e-07	2.18	4.50e-07	2.18
$801 \times 16e5$	Reference			
Accuracy test of G				
$26 \times 5e4$	2.66e-04		6.09e-06	
$51 \times 1e5$	6.65e-05	2.00	1.52e-06	2.00
$101 \times 2e5$	1.68e-05	1.99	3.85e-07	1.98
$201 \times 4e5$	4.20e-06	2.00	9.65e-08	2.00
$401 \times 8e5$	9.38e-07	2.16	2.17e-08	2.15
$801 \times 16e5$	Reference			

TABLE 1. Convergence test of Runge-Kutta method.

$N_z \times N_t$	$L_\infty Error$	Order	$L_2 Error$	Order
Accuracy test of W				
$26 \times 5e4$	1.85e-04		2.68e-04	
$51 \times 1e5$	4.65e-05	2.00	3.30e-05	1.99
$101 \times 2e5$	1.18e-05	1.98	8.30e-06	1.99
$201 \times 4e5$	2.92e-06	2.01	2.06e-06	2.01
$401 \times 8e5$	6.30e-07	2.21	4.38e-07	2.23
$801 \times 16e5$	Reference			
Accuracy test of G				
$26 \times 5e4$	1.33e-04		6.13e-05	
$51 \times 1e5$	6.72e-05	2.01	1.54e-05	1.99
$101 \times 2e5$	1.71e-05	1.98	3.92e-06	1.97
$201 \times 4e5$	4.30e-06	1.99	9.90e-07	1.99
$401 \times 8e5$	9.30e-07	2.21	2.15e-07	2.20
$801 \times 16e5$	Reference			

TABLE 2. Convergence test of Crank-Nicolson method.

Crank-Nicolson, IIF and Krylov IIF allow for moderate large time step size, which exhibit extremely nice stability conditions.

In Table 5, we also further test the order of accuracy for Crank-Nicolson, IIF2 and Krylov IIF2 with fixed spatial resolution $N_z=801$ and six different time resolution. The spatial resolution $N_z=801$ is chosen fine enough such that the error is dominated by the time step. We list the maximum error between the numerical solution and the reference solution, and the order of accuracy. Clearly, the orders of accuracy in time for Crank-Nicolson, IIF2 and Krylov IIF2 are two.

3.3. Efficiency test for stable schemes. Since Crank-Nicolson, IIF2 and Krylov IIF2 exhibit similar stability conditions, we test the efficiency for these three schemes in this section. To do this, here we consider two cases: case 1: large diffusion system

$N_z \times N_t$	$L_\infty Error$	Order	$L_2 Error$	Order
Accuracy test of W				
$26 \times 5e4$	1.82e-04		1.31e-04	
$51 \times 1e5$	4.51e-05	2.02	3.20e-05	2.03
$101 \times 2e5$	1.11e-05	2.02	7.84e-06	2.03
$201 \times 4e5$	2.65e-06	2.07	1.86e-06	2.07
$401 \times 8e5$	5.34e-07	2.31	3.73e-07	2.32
$801 \times 16e5$	Reference			
Accuracy test of G				
$26 \times 5e4$	2.65e-04		6.07e-06	
$51 \times 1e5$	6.58e-05	2.01	1.51e-06	2.01
$101 \times 2e5$	1.64e-05	2.00	3.78e-07	2.00
$201 \times 4e5$	4.00e-06	2.04	9.26e-08	2.03
$401 \times 8e5$	8.35e-07	2.26	1.94e-08	2.26
$801 \times 16e5$	Reference			

TABLE 3. Convergence test of IIF2 method.

$N_z \times N_t$	$L_\infty Error$	Order	$L_2 Error$	Order
Accuracy test of W				
$26 \times 5e4$	1.82e-04		1.31e-04	
$51 \times 1e5$	4.51e-05	2.02	3.20e-05	2.03
$101 \times 2e5$	1.11e-05	2.02	7.84e-06	2.03
$201 \times 4e5$	2.65e-06	2.07	1.86e-06	2.07
$401 \times 8e5$	5.30e-07	2.32	3.72e-07	2.32
$801 \times 16e5$	Reference			
Accuracy test of G				
$26 \times 5e4$	2.65e-04		6.07e-06	
$51 \times 1e5$	6.58e-05	2.01	1.51e-06	2.01
$101 \times 2e5$	1.64e-05	2.00	3.78e-07	2.00
$201 \times 4e5$	4.00e-06	2.04	9.26e-08	2.03
$401 \times 8e5$	8.40e-07	2.25	1.94e-08	2.26
$801 \times 16e5$	Reference			

TABLE 4. Convergence test of Krylov IIF2 method.

with $D = 100, a = b = 1$; case 2: stiff system with $D = 1, a = b = 100$. For both cases, we set $\mu = 1, H_0 = 2$, the final simulation time $T_{\text{end}} = 0.1$, and the initial condition $U_0 = \cos(\frac{x\pi}{2H_0})$. For all the simulations here, we select three different spatial resolutions $N_z = 1001, N_z = 2001$ and $N_z = 4001$ with fixed time step size $\Delta t = 10^{-4}$.

From Figures 3 and 4, first we show that the solutions agree with each other for three different schemes. We can also see that when compared to the stiff system, the front H for the large diffusion system moves much faster initially and quickly reaches the steady state afterwards.

From Table (6), for the large diffusion system, it is clear that the regular IIF2 is more than 20 times slower for the refined grid than the other two due to the very expensive evaluation of exponential matrices at every time step, while Krylov IIF2 is slightly faster than Crank-Nicolson for all different grid resolution.

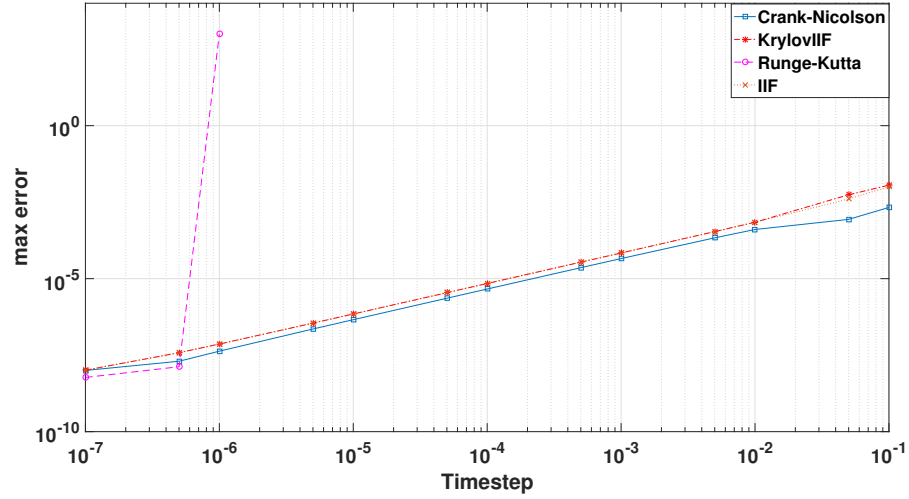


FIGURE 1. Error of U as a function of time step sizes

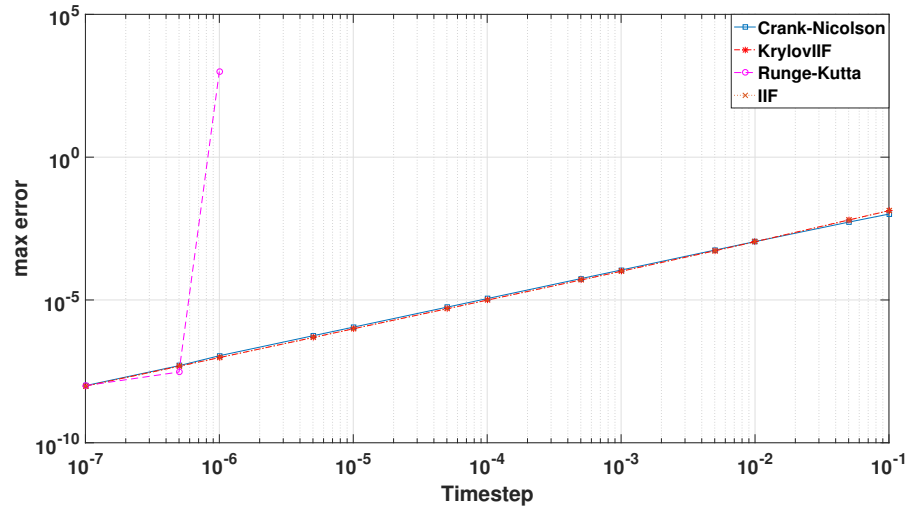


FIGURE 2. Error of H as a function of time step sizes

From Table (7), we can show that all the simulations for the stiff system are much slower than those of large diffusion system with the same grid size. Once again, the regular IIF2 is extremely slower than the other two, while Krylov IIF2 is more than 2 times faster than Crank-Nicolson and such trend is more obvious when the grid becomes finer. To our understanding, for the Crank-Nicolson scheme, it requires solving nonlinear system for each time step, thus demanding more CPU time for the stiff systems. However, Krylov IIF2 avoids solving the nonlinear system, instead it only handles nonlinear equation for each grid point. Thus Krylov IIF can reduce computational cost for the system with very stiff reaction terms. In brief,

Δt	Crank-Nicolson		IIF2		Krylov IIF2	
Accuracy test of W						
	L_∞ error	Order	L_∞ error	Order	L_∞ error	Order
8.0×10^{-5}	1.54e-8	-	6.50e-8	-	6.50e-8	-
4.0×10^{-5}	4.09e-9	1.91	2.23e-8	1.55	2.23e-8	1.55
2.0×10^{-5}	1.05e-9	1.97	6.28e-9	1.83	6.28e-9	1.83
1.0×10^{-5}	3.22e-10	1.70	1.30e-9	2.27	1.30e-9	2.27
5.0×10^{-6}	8.14e-11	1.98	2.85e-10	2.20	2.85e-10	2.20
2.5×10^{-6}	Reference		Reference		Reference	
Accuracy test of G						
	L_∞ error	Order	L_∞ error	Order	L_∞ error	Order
8.0×10^{-5}	4.16e-7	-	8.49e-7	-	8.49e-7	-
4.0×10^{-5}	1.44e-7	1.53	2.81e-7	1.59	2.81e-7	1.59
2.0×10^{-5}	4.86e-8	1.57	9.22e-8	1.61	9.22e-8	1.61
1.0×10^{-5}	1.54e-8	1.66	2.88e-8	1.68	2.88e-8	1.68
5.0×10^{-6}	3.92e-9	1.97	7.27e-9	1.99	7.27e-9	1.99
2.5×10^{-6}	Reference		Reference		Reference	

TABLE 5. Errors and order of accuracy in time for three stable schemes: Crank-Nicolson, IIF2 and Krylov IIF2

$\Delta t = 10^{-4}$	$N_z = 1001$	$N_z = 2001$	$N_z = 4001$
Crank-Nicolson	16.576	75.092	395.512
Krylov IIF2	14.595	59.566	295.958
IIF2	211.227	1099.695	9694.277

TABLE 6. Efficiency test for the large diffusion system

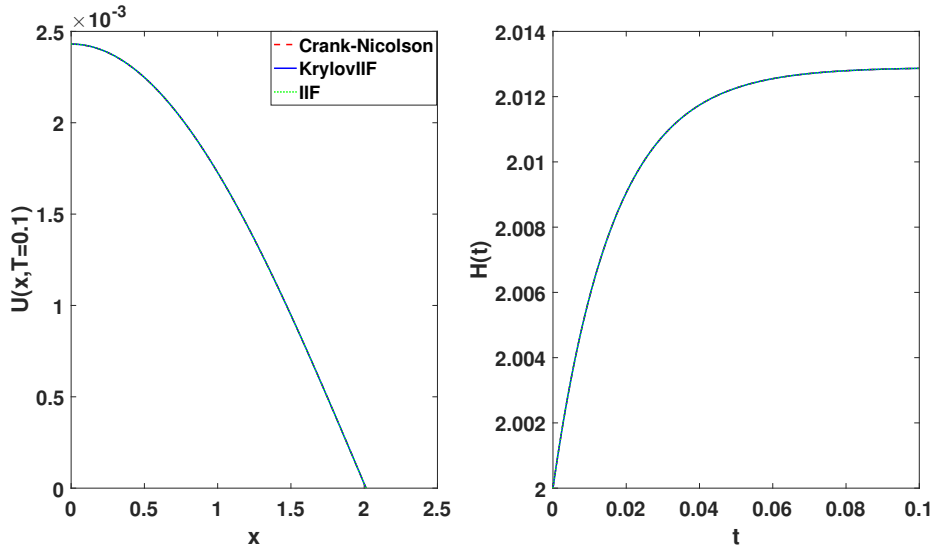
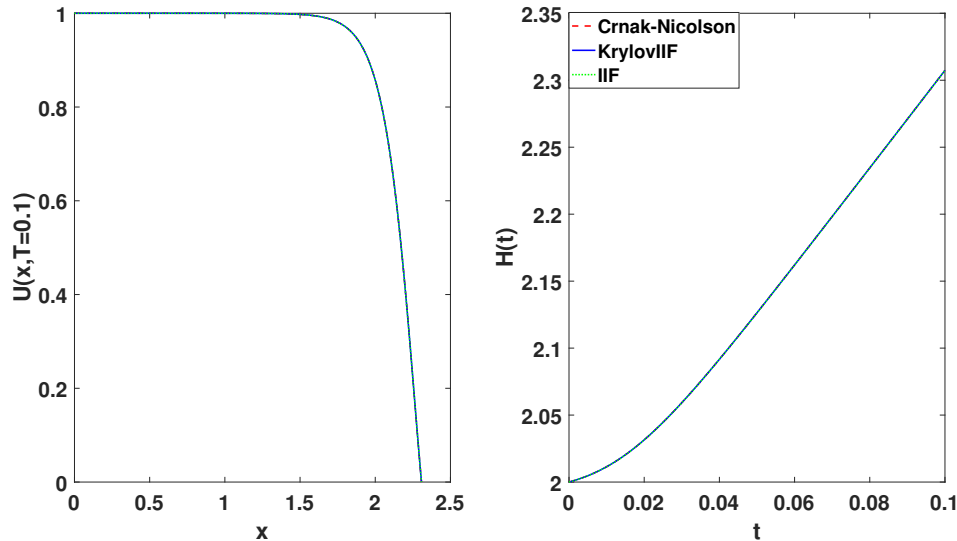


FIGURE 3. Solution U and H for the large diffusion system

$\Delta t = 10^{-4}$	$N_z = 1001$	$N_z = 2001$	$N_z = 4001$
Crank-Nicolson	30.149	141.601	760.832
Krylov IIF2	16.706	71.501	346.278
IIF2	389.514	1877.676	22626.109

TABLE 7. Efficiency test for the stiff system

FIGURE 4. Solution U and H for the stiff system

applying Krylov subspace method incorporated with IIF is able to dramatically further improve the efficiency for solving very stiff systems with moving boundaries.

4. Conclusion. In this paper, we introduce and derive four different numerical schemes: Runge-Kutta, Crank-Nicolson, IIF2 and Krylov IIF2, to systematically study reaction-diffusion systems with moving boundaries. Through numerical experiments, we first show that all four proposed schemes obey the second order accuracy in both space and time. Not surprisingly, we also show that the three schemes: Crank-Nicolson, IIF and Krylov IIF exhibit very nice stability property, while Runge-Kutta only admits for very small time steps with the fine mesh size. Finally through efficiency test, it reveals that the regular IIF is too expensive for solving such systems due to the evaluation of exponential matrices at every time step, while Krylov IIF is the most efficient approach, especially for the system with very stiff reaction terms. For example, Krylov IIF is more than 2 times faster than Crank-Nicolson and over 20 times faster than the regular IIF for a stiff system. Certainly the efficiency of the Krylov IIF method depends on the dimension size of the selected Q for the Krylov subspace approximation. In this paper, the dimension size Q is taken to be 25 for most of our numerical tests, in which the accuracy of the original IIF is still preserved.

Currently we are incorporating level set method with Krylov IIF for solving reaction-diffusion systems with moving boundaries in two dimensions, and some

nice preliminary results have been obtained that further demonstrates the great promise of the proposed approach. Compared to 1D problems, there are quite a few new numerical challenges when Krylov IIF is coupled with level set method for solving 2D problems. Firstly, due to the moving boundaries, the associated exponential matrices for discretized diffusion operator need to be assembled with cautious by an ordered vector for each time step. Secondly, the level set function should be carefully incorporated into IIF solver for reaction-diffusion equations with some special numerical treatment, especially for the points near moving boundaries. Finally, new ideas for extrapolation on the boundary points with different boundary conditions need to be explored to keep the higher order accuracy. Based on these observations, we would like to publish this work in a separate paper.

Acknowledgments. We would like to thank Professor Yihong Du for stimulating discussion and valuable comments to improve the manuscript. This work is partially supported by the NSF grant DMS 1853365.

REFERENCES

- [1] W. Bao, Y. Du, Z. Lin and H. Zhu, [Free boundary models for mosquito range movement driven by climate warming](#), *Journal of Mathematical Biology*, **76** (2018), 841–875.
- [2] G. Bunting, Y. Du and K. Krakowski, [Spreading speed revisited: analysis of a free boundary model](#), *Networks and Heterogeneous Media*, **7** (2012), 583–603.
- [3] K. Burrage and J. C. Butcher, [Stability criteria for implicit Runge-Kutta methods](#), *SIAM Journal on Numerical Analysis*, **16** (1979), 46–57.
- [4] L. A. Caffarelli and S. Salsa, *A Geometric Approach to Free Boundary Problems*, American Mathematical Soc., 2005.
- [5] Y. Cao, A. Faghri and W. S. Chang, [A numerical analysis of Stefan problems for generalized multi-dimensional phase-change structures using the enthalpy transforming model](#), *International Journal of Heat and Mass Transfer*, **32** (1989), 1289–1298.
- [6] H. Chen, C. Min and F. Gibou, [A numerical scheme for the Stefan problem on adaptive Cartesian grids with supralinear convergence rate](#), *Journal of Computational Physics*, **228** (2009), 5803–5818.
- [7] S. Chen, B. Merriman, S. Osher and P. Smereka, [A simple level set method for solving Stefan problems](#), *Journal of Computational Physics*, **135** (1997), 8–29.
- [8] S. Chen and Y. Zhang, [Krylov implicit integration factor methods for spatial discretization on high dimensional unstructured meshes: application to discontinuous Galerkin methods](#), *Journal of Computational Physics*, **230** (2011), 4336–4352.
- [9] I. L. Chern, J. Glimm, O. McBryan, B. Plohr and S. Yaniv, [Front tracking for gas dynamics](#), *Journal of Computational Physics*, **62** (1986), 83–110.
- [10] J. Crank, *Free and Moving Boundary Problems*, Clarendon Press, Oxford, 1984.
- [11] Y. Du and Z. Guo, [The Stefan problem for the Fisher-KPP equation](#), *Journal of Differential Equations*, **253** (2012), 996–1035.
- [12] Y. Du and Z. Lin, [Spreading-vanishing dichotomy in the diffusive logistic model with a free boundary](#), *SIAM Journal on Mathematical Analysis*, **42** (2010), 377–405.
- [13] Y. Du and B. Lou, [Spreading and vanishing in nonlinear diffusion problems with free boundaries](#), *Journal of the European Mathematical Society*, **17** (2015), 2673–2724.
- [14] Y. Du, H. Matano and K. Wang, [Regularity and asymptotic behavior of nonlinear Stefan problems](#), *Archive for Rational Mechanics and Analysis*, **212** (2014), 957–1010.
- [15] R. Fedkiw and S. Osher, *Level Set Methods and Dynamic Implicit Surfaces*, Applied Mathematical Sciences, 153. Springer-Verlag, New York, 2003.
- [16] E. Gallopoulos and Y. Saad, [Efficient solution of parabolic equations by Krylov approximation methods](#), *SIAM Journal on Scientific and Statistical Computing*, **13** (1992), 1236–1264.
- [17] F. Gibou and R. Fedkiw, [A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem](#), *Journal of Computational Physics*, **202** (2005), 577–601.
- [18] J. Glimm, X. L. Li, Y. Liu and N. Zhao, [Conservative front tracking and level set algorithms](#), *Proceedings of the National Academy of Sciences*, **98** (2001), 14198–14201.

- [19] E. Hairer and G. Wanner, [Stiff differential equations solved by Runge-Kutta methods](#), *Journal of Computational and Applied Mathematics*, **111** (1999), 93–111.
- [20] N. J. Higham, [The scaling and squaring method for the matrix exponential revisited](#), *SIAM Journal on Matrix Analysis and Applications*, **26** (2005), 1179–1193.
- [21] J. Hilditch and P. Colella, [A front tracking method for compressible flames in one dimension](#), *SIAM Journal on Scientific Computing*, **16** (1995), 755–772.
- [22] M. Hochbruck and C. Lubich, [On Krylov subspace approximations to the matrix exponential operator](#), *SIAM Journal on Numerical Analysis*, **34** (1997), 1911–1925.
- [23] J. Hua, J. F. Stene and P. Lin, [Numerical simulation of 3D bubbles rising in viscous liquids using a front tracking method](#), *Journal of Computational Physics*, **227** (2008), 3358–3382.
- [24] T. Jiang and Y. Zhang, [Krylov implicit integration factor WENO methods for semilinear and fully nonlinear advection-diffusion-reaction equations](#), *Journal of Computational Physics*, **253** (2013), 368–388.
- [25] T. Jiang and Y. Zhang, [Krylov single-step implicit integration factor WENO method for advection-diffusion-reaction equations](#), *Journal of Computational Physics*, **311** (2016), 22–44.
- [26] H. G. Landau, [Heat conduction in a melting solid](#), *Quarterly of Applied Mathematics*, **8** (1950), 81–94.
- [27] R. J. Leveque and Z. Li, [The immersed interface method for elliptic equations with discontinuous coefficients and singular sources](#), *SIAM Journal on Numerical Analysis*, **31** (1994), 1019–1044.
- [28] R. J. Leveque and K. M. Shyue, [Two-dimensional front tracking based on high resolution wave propagation methods](#), *Journal of Computational Physics*, **123** (1996), 354–368.
- [29] S. Liu and X. Liu, [Numerical methods for a two-species competition-diffusion model with free boundaries](#), *Mathematics*, **6** (2018), 72.
- [30] D. Lu and Y. Zhang, [Krylov integration factor method on sparse grids for high spatial dimension convection-diffusion equations](#), *Journal of Scientific Computing*, **69** (2016), 736–763.
- [31] M. M. Mac Low and R. S. Klessen, [Control of star formation by supersonic turbulence](#), *Reviews of Modern Physics*, **76** (2004), 125.
- [32] C. Moler and C. Van Loan, [Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later](#), *SIAM Review*, **45** (2003), 3–49.
- [33] Q. Nie, F. Y. Wan, Y. Zhang and X. Liu, [Compact integration factor methods in high spatial dimensions](#), *Journal of Computational Physics*, **277** (2008), 5238–5255.
- [34] Q. Nie, Y. Zhang and R. Zhao, [Efficient semi-implicit schemes for stiff systems](#), *Journal of Computational Physics*, **214** (2006), 521–537.
- [35] S. Osher and R. P. Fedkiw, [Level set methods: an overview and some recent results](#), *Journal of Computational Physics*, **169** (2001), 463–502.
- [36] S. Osher and J. A. Sethian, [Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations](#), *Journal of Computational Physics*, **79** (1988), 12–49.
- [37] D. Peng, B. Merriman, S. Osher, H. Zhao and M. Kang, [A PDE-based fast local level set method](#), *Journal of Computational Physics*, **155** (1999), 410–438.
- [38] C. S. Peskin, [The immersed boundary method](#), *Acta Numerica*, **11** (2002), 479–517.
- [39] M. A. Piqueras, R. Company and L. Jodar, [A front-fixing numerical method for a free boundary nonlinear diffusion logistic population model](#), *Journal of Computational and Applied Mathematics*, **309** (2017), 473–481.
- [40] L. I. Rubinstein, *The Stefan Problem*, Providence, RI: American Mathematical Society, 1971.
- [41] Y. Saad, [Analysis of some Krylov subspace approximations to the matrix exponential operator](#), *SIAM Journal on Numerical Analysis*, **29** (1992), 209–228.
- [42] J. A. Sethian, [A fast marching level set method for monotonically advancing fronts](#), *Proceedings of the National Academy of Sciences*, **93** (1996), 1591–1595.
- [43] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, 1999.
- [44] G. D. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, Oxford University Press, 1985.
- [45] M. Sussman, P. Smereka and S. Osher, [A level set approach for computing solutions to incompressible two-phase flow](#), *Journal of Computational Physics*, **114** (1994), 146–159.
- [46] L. N. Trefethen and D. Bau, *III Numerical Linear Algebra*, SIAM, 1997.

- [47] S. O. Unverdi and G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, *Journal of Computational Physics*, **100** (1992), 25–37.
- [48] A. Wiegmann and K. P. Bube, [The immersed interface method for nonlinear differential equations with discontinuous coefficients and singular sources](#), *SIAM Journal on Numerical Analysis*, **35** (1998), 177–200.
- [49] J. J. Xu, Z. Li, J. Lowengrub and H. Zhao, [A level-set method for interfacial flows with surfactant](#), *Journal of Computational Physics*, **212** (2006), 590–616.
- [50] H. K. Zhao, T. Chan, B. Merriman and S. Osher, [A variational level set approach to multi-phase motion](#), *Journal of Computational Physics*, **127** (1996), 179–195.
- [51] L. Zhu and C. S. Peskin, [Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method](#), *Journal of Computational Physics*, **179** (2002), 452–468.

Received November 2018; revised February 2019.

E-mail address: shuang@email.sc.edu

E-mail address: xfliu@math.sc.edu