Short-Time Approximation of a Prescribed Gaussian Curvature Equation: ADM vs Finite Difference Methods

Sura Mathboob Hull[1], Omar Al-Tameemi[2] and Faik Mayah[3]
[1]*Department of Mathematics, College of Education for Pure Sciences, Wasit University, 52001 Al-Kut, Iraq*
[2]*Department of Mathematics, College of Science, Al-Nahrain University, 10072 Baghdad, Iraq*
[3]*Department of Physics, College of Science, Wasit University, 52001 Al-Kut, Iraq*
suramathboob1999@gmail.com, omar.ismael@nahrainuniv.edu.iq, fmayah@uowasit.edu.iq

Keywords: Curvature-Driven Flow, Finite Difference Method, Numerical Simulation, Semi-Discretization.

Abstract: The short-time evolution of a two-dimensional curvature-driven flow is investigated using smooth initial data. The governing equation is formulated for a height function $f(x,y,t)$ over a rectangular domain and is evolved under its associated curvature operator. Two truncated Adomian Decomposition Method approximations, namely the first- and second-order schemes (ADM1 and ADM2), are derived and compared with a finite difference (FD) reference solution. To assess the accuracy and short-time behavior of the proposed methods, a smooth Gaussian initial condition is considered. The numerical results show that both ADM1 and ADM2 accurately capture the curvature-driven dynamics and agree closely with the FD solution within the short-time regime. The corresponding error distributions remain small and smooth over the computational domain, indicating stable and reliable performance of the Adomian-based approximations.Furthermore, the difference between the first- and second-order truncations is found to be negligible over the considered time interval, which suggests that even low-order Adomian expansions are sufficient to achieve reliable accuracy over short-time intervals. The results suggest that the Adomian Decomposition Method provides an effective way to approximate curvature-driven flows in short-time regimes.

1 INTRODUCTION

Curvature-driven geometric flows have attracted considerable interest in differential geometry and mathematical analysis. They also play a key role in partial differential equations, since they characterize the time evolution of curves and surfaces. Research on these flows can be traced back to 1957. Mullins [1] proposed a physical model to describe surface evolution at grain boundaries in polycrystalline materials. It was shown that surface motion is driven by mechanisms that aim to minimize surface energy. Later, the Gaussian curvature flow was rigorously studied by Andrews [2].

The existence and uniqueness of solutions were established. It was also shown that strictly convex surfaces converge smoothly to a spherical shape. This provided a theoretical foundation for curvature-driven flows. In 2006, Zhao and Xu [3] applied Gaussian curvature flow to triangular surface mesh fairing. Its effectiveness in smoothing convex meshes was demonstrated within a numerical framework.. In 2013, fast iterative algorithms for surface fairing were proposed by Brito-Loza and Chen [4]. These methods were based on curvature-related models, including Gaussian curvature. Special attention was given to computational efficiency. More recently, curvature-driven flows have been extended to nonlocal settings. In particular, Cinti et al. [5] investigated volume-preserving fractional mean curvature flows.

These flows preserve volume. Important analytical properties were also established for smooth convex solutions. Additionally, Julin and La Manna [6] proved that smooth solutions exist for fractional mean curvature flow over short time intervals under suitable conditions. Furthermore, Cesaroni and Novaga [7] studied fractional mean curvature flow defined on Lipschitz graphs, analyzing its long-time behavior and stability, and showing convergence to self-similar solutions under suitable conditions.

In 2024, Yuta Noma et al. [8] presented an efficient numerical algorithm based on curve flows for surface filling, formulated via the gradient flow of a simple energy function, with particular emphasis on computational efficiency, stability, and

controllability in engineering modeling and computer graphics applications.

In this context, a numerical representation based on a height function $f(x,y,t)$ is employed.In the short-time regime, such equations provide a suitable framework for studying the behavior of numerical solutions and for approximating them through low-order analytic–numerical approaches.ap. This regime is particularly suitable for assessing the accuracy of time-expansion approximations. It also enables comparison with reference solutions computed using finite difference methods. This study focuses on the following aspects:

- Using the Adomian Decomposition Method, low-order short-time approximations are derived and implemented. In particular, the first- and second-order schemes (ADM1 and ADM2) are applied to the prescribed Gaussian curvature equation;
- Construction of a finite difference (FD) scheme on a regular rectangular grid to serve as a reference numerical solution for comparison;
- Use of manufactured solutions to enable a quantitative and accurate evaluation of the performance of the numerical methods;
- Investigation of the relative error behavior of short-time approximations and the implementation of a consistency test for derivatives at the initial time $t = 0$.

This work does not aim to develop high-order numerical schemes, but rather to analyze the accuracy and efficiency of short-time approximations based on the Adomian method and to compare their performance with that of the finite difference method within a clear and systematic numerical framework.

2 MODEL PROBLEM AND CURVATURE OPERATOR

A surface is considered in the form of a graph defined by a height function:

$$u: \Omega \times [0,T] \to \mathbb{R},$$

defined on the rectangular domain:

$$\Omega = [-1,1] \times [-1,1].$$

Instead of solving the steady prescribed Gaussian curvature equation directly, we introduce a curvature-driven evolution problem and employ a manufactured solution framework for validation.

The evolution problem can be expressed as:

$$u_t(x,y,t) = \kappa\big(u(x,y,t)\big) - g(x,y,t), (x,y) \in \Omega, t > 0, \quad (1)$$

where $\kappa(u)$ denotes the Gaussian curvature of the surface. The Gaussian curvature is given by:

$$\kappa(u) = \frac{(u_{xx}u_{yy} - {u_{xy}}^2)}{\left(1 + {u_x}^2 + {u_y}^2\right)^2}.$$

In the numerical implementation, a small regularization term is introduced in the denominator to enhance stability and avoid numerical singularities.This explicit formulation will be used throughout the analysis.

The initial condition is chosen consistently with the manufactured solution:

$$u(x,y,0) = a(0)\,(x^2 + y^2),\ (x,y) \in \Omega. \quad (2)$$

The manufactured exact solution is defined as:

$$u(x,y,t) = a(t)\,(x^2 + y^2), \quad (3)$$

where

$$a(t) = 1 + 0.5 \sin(2\pi t).$$

Note that the manufactured solution does not exactly satisfy the homogeneous Neumann boundary conditions; however, the numerical scheme applies a reflection-based boundary treatment to approximate the Neumann condition and maintain numerical stability and consistency.

The forcing term $g(x,y,t)$ is introduced to ensure that the given function exactly satisfies the evolution equation. Substituting into the PDE gives:

$$g(x,y,t) = \kappa(u(x,y,t)) - u_t(x,y,t).$$

For the quadratic profile $u = a(t)(x^2 + y^2)$, we obtain:

$$u_t = a'(t)(x^2 + y^2),$$

where $a'(t) = \pi\cos(2\pi t)$ and the Gaussian curvature:

$$\kappa(u) = \frac{4a(t)^2}{(1 + 4a(t)^2(x^2 + y^2))^2}.$$

Hence:

$$g(x,y,t) = \frac{4a(t)^2}{(1 + 4a(t)^2(x^2 + y^2))^2} - a'(t)(x^2 + y^2). \quad (4)$$

Homogeneous Neumann boundary conditions are imposed as follows:

$$\frac{\partial u}{\partial n} = 0,\ (x,y) \in \partial\Omega,\ t \geq 0, \tag{5}$$

where $\partial u/\partial n$ denotes the normal derivative on the boundary.

2.1 Short-Time Expansion

To analyze the behavior of the solution for small time, a Taylor expansion about $t = 0$ is employed:

$$u(x,y,t) = u_0(x,y) + t\,u^{(1)}(x,y) + \frac{t^2}{2}\,u^{(2)}(x,y) + \mathcal{O}(t^3), \tag{6}$$

Where $u_0(x,y) = u(x,y,0)$. Differentiating the evolution equation $u_t = \kappa(u) - g(x,y,t)$ and evaluating at $t = 0$ yields:

$$u^{(1)}(x,y) = \kappa(u_0(x,y)) - g(x,y,0), \tag{7}$$

$$u^{(2)}(x,y) = D\kappa[u_0]\left[u^{(1)}\right](x,y) - g_t(x,y,0), \tag{8}$$

where $D\kappa[u_0]$ denotes the Fréchet derivative of the curvature operator at u_0. For computational purposes, it is approximated numerically using a symmetric directional difference. Therefore, ADM1 and ADM2 correspond to the first- and second-order truncations of the Taylor expansion around $t = 0$, respectively.

3 ADOMIAN DECOMPOSITION FOR CURVATURE FLOW

The Adomian Decomposition Method (ADM) is recognized as a powerful semi-analytical approach that was introduced by George Adomian in the early 1980s. It has been extensively utilized to obtain solutions for nonlinear ordinary as well as partial differential equations, It has also been applied to algebraic and integral equations. The method is based on representing the solution of a nonlinear equation as a series of functions, where the components of the series are generated using Adomian polynomials obtained from the power-series expansion of analytic functions. The ADM has shown strong efficiency in generating accurate and fast-converging approximate solutions without the need to linearize the governing equations or assume small perturbations. The convergence properties of the method were have been studied by Cherruault and Adomian [9], while Babolian and Biazar established the order of convergence [10]. Consequently, ADM is considered a powerful and efficient tool for the analysis of nonlinear systems within both mathematical and numerical frameworks [11], [12]. Moreover, recent studies have demonstrated its effectiveness in solving nonlinear ordinary and partial differential equations, including fractional models [13]-[15].

3.1 Adomian Decomposition Formulation

The curvature-driven evolution problem is considered as follows:

$$\frac{\partial u(x,y,t)}{\partial t} = \kappa(u(x,y,t)) - g(x,y,t) \tag{9}$$

where the (Gaussian) curvature operator acting on the graph $z = u(x,y,t)$ is given by:

$$\kappa(u) = \frac{u_{xx}u_{yy} - u_{xy}^2}{\left(1 + u_x^2 + u_y^2\right)^2} \tag{10}$$

Based on the operator framework adopted in ADM studies, the following definition is introduced:

$$\mathcal{L}u := u_t,\ \mathcal{N}(u) := -\kappa(u) + g(x,y,t) \tag{11}$$

so that (9) may be written as:

$$\mathcal{L}u = -\mathcal{N}(u) \tag{12}$$

The Volterra equation is obtained by integrating over the interval from 0 to t:

$$u(x,y,t) = u_0(x,y) - \int_0^t \mathcal{N}(u(x,y,s))\,ds \tag{13}$$

where $u_0(x,y) = u(x,y,0)$ encodes the initial condition. ADM seeks a series representation of the solution:

$$u(x,y,t) = \sum_{n=0}^{\infty} u_n(x,y,t), \tag{14}$$

and expands the nonlinearity in terms of Adomian polynomials:

$$\mathcal{N}(u(x,y,t)) = \sum_{n=0}^{\infty} A_n(x,y,t). \tag{15}$$

By substituting (14–15) into (13) and matching like terms, the following recurrence is obtained:

$$u_0(x,y,t) = u_0(x,y), \tag{16}$$

$$u_{n+1}(x,y,t) = -\int_0^t A_n(x,y,s)\,ds,\ n \geq 0. \tag{17}$$

3.1.1 ADM1 and ADM2 Truncations

In the present study we use low-order truncations:

$$ADM1: u^{ADM1}(x,y,t): = u_0(x,y) + u_1(x,y,t), \quad (18)$$

$$ADM2: u^{ADM2}(x,y,t): = u_0(x,y) + u_1(x,y,t) + u_2(x,y,t) \quad (19)$$

For the operator splitting $\mathcal{L}u := u_t$ and $\mathcal{N}(u) := -\kappa(u) + g(x,y,t)$,the first two Adomian polynomials are:

$$A_0(x,y,t) = \mathcal{N}(u_0(x,y)) = -\kappa(u_0(x,y)) + g(x,y,t), \quad (20)$$

$$A_1(x,y,t) = D\mathcal{N}[u_0](u_1(\cdot,\cdot,t)) = -D\kappa[u_0](u_1(\cdot,\cdot,t)), \quad (21)$$

where $D\kappa[u_0]$ represents the Fréchet derivative of the curvature operator κ evaluated at the initial surface u_0. From the recurrence relation:

$$u_{n+1}(x,y,t) = -\int_0^t A_n(x,y,s)\, ds, \qquad n \geq 0,$$

we obtain:

$$u_1(x,y,t) = -\int_0^t A_0(x,y,s)ds = t\kappa(u_0(x,y)) - \int_0^t g(x,y,s)\, ds, \quad (22)$$

$$u_2(x,y,t) = -\int_0^t A_1(x,y,s)\, ds = -\int_0^t D\kappa[u_0](u_1(\cdot,\cdot,s))\, ds. \quad (23)$$

In particular, for a short-time approximation where the forcing term is treated at $t = 0$, we have:

$$u_1(x,y,t) \approx t(\kappa(u_0(x,y)) - g(x,y,0)), \quad (24)$$

$$u_2(x,y,t) \approx \frac{-t^2}{2} D\kappa[u_0](\kappa(u_0) - g(\cdot,\cdot,0)) \quad (25)$$

so that ADM1 and ADM2 correspond to the first- and second-order truncations of the short-time expansion (6). In practice, all spatial derivatives entering $\kappa(u_0)$ and $D\kappa[u_0](\kappa(u_0))$ are evaluated by centered finite differences on the same uniform grid used by the finite difference scheme. This ensures that ADM and FD see the same discrete representation of curvature.

4 FINITE DIFFERENCE SEMI-DISCRETIZATION

The finite difference method (FDM) is broadly applied as a core numerical technique for the solution of partial differential equations. The continuous domain is discretized into a grid, in which derivatives are approximated through finite difference schemes. This method originates from the foundational contributions of Euler and Lagrange. Its significance has grown considerably with the development of modern computational technologies. Early studies primarily addressed stability and convergence properties [16]. In addition, computational performance and numerical error control were examined [17]. These developments led to the systematic construction of finite difference schemes. More recent research has further improved these methods[18]. Higher-order schemes have been developed. Enhanced stability and convergence properties for nonlinear differential equations have also been reported [19]-[21].

Let (x_i,y_j), $i = 1,\ldots,N_x$, $j = 1,\cdots,N_y$, denote a uniform grid on Ω with mesh widths h_x and h_y. We denote the grid function at time t_n by:

$$U_{i,j}^n \approx u(x_i,y_j,t_n).$$

4.1 Spatial Discretization

The first derivatives are approximated by:

$$u_x(x_i,y_j,t_n) \approx \frac{U_{i+1,j}^n - U_{i-1,j}^n}{2h_x}, \quad u_y(x_i,y_j,t_n) \approx \frac{U_{i,j+1}^n - U_{i,j-1}^n}{2h_y}. \quad (26)$$

The second derivatives are approximated by:

$$u_{xx}(x_i,y_j,t_n) \approx \frac{U_{i+1,j}^n - 2U_{i,j}^n + U_{i-1,j}^n}{h_x^2}, \quad u_{yy}(x_i,y_j,t_n) \approx \frac{U_{i,j+1}^n - 2U_{i,j}^n + U_{i,j-1}^n}{h_y^2}. \quad (27)$$

The mixed derivative is approximated by:

$$u_{xy}(x_i,y_j,t_n) \approx \frac{U_{i+1,j+1}^n - U_{i+1,j-1}^n}{4h_xh_y} \frac{-U_{i-1,j+1}^n + U_{i-1,j-1}^n}{4h_xh_y}. \quad (28)$$

Using (26-28), the discrete Gaussian curvature operator is defined by:

$$\kappa_h(U^n)_{i,j} := \frac{u_{xx}^h(x_i,y_j,t_n)u_{yy}^h(x_i,y_j,t_n) - \left(u_{xy}^h(x_i,y_j,t_n)\right)^2}{\left(1+\left(u_x^h(x_i,y_j,t_n)\right)^2+\left(u_y^h(x_i,y_j,t_n)\right)^2\right)^2} \quad (i,j)\in\mathcal{I}, \tag{29}$$

where $\mathcal{I}$ denotes the set of interior grid indices. Collecting all interior values into a vector $U^n \in \mathbb{R}^M$ (with $M = (N_x-2)(N_y-2)$), we obtain the semi-discrete system:

$$\frac{dU^n}{dt} = K(U^n) - S \tag{30}$$

where $K(\cdot)$ denotes the discrete curvature operator induced by (29) and S is the grid sampling of the prescribed curvature function.

4.2 Time Stepping (Explicit Euler)

To compute a steady solution of the prescribed curvature equation $\kappa(u) = S(x,y)$, we evolve the pseudo-time problem:

$$u_t = \kappa(u) - S(x,y), \tag{31}$$

until convergence. Using explicit Euler, we update the grid function by:

$$U_{i,j}^{n+1} = U_{i,j}^n + \Delta t\left(\kappa_h(U^n)_{i,j} - S_{i,j}\right),\ (i,j)\in\mathcal{I}. \tag{32}$$

The explicit Euler scheme is conditionally stable. Therefore, a sufficiently small time step is used in all computations. In particular, $\Delta t = 10^{-7}$ is adopted to ensure stable numerical evolution over the short-time interval. No numerical instabilities or blow-up were observed in the simulations. The finite difference solution is used as a reference for comparison with the ADM approximations.

4.3 Boundary Conditions and Source Term

In agreement with the implementation, we impose homogeneous Neumann boundary conditions using a reflection technique:

$$U_{1,j}^n = U_{2,j}^n,\quad U_{N_x,j}^n = U_{N_x-1,j}^n\ \ j = 1,\ \ldots,N_y,$$

$$U_{i,1}^n = U_{i,2}^n,\quad U_{i,N_y}^n = U_{i,N_y-1}^n\ \ i = 1,\ \ldots,N_x,$$

These conditions ensure a consistent discrete approximation of the homogeneous Neumann boundary condition. For all time steps n. The prescribed curvature used in the numerical experiments is:

$$S(x,y) = \exp\left(-(x^2+y^2)\right),\quad S_{i,j} = S(x_i,y_j). \tag{33}$$

5 INITIAL CONDITIONS AND CURVATURE STRUCTURE

We employ a manufactured initial condition consistent with the exact solution used for validation.

5.1 Manufactured Smooth Initial Condition

As shown in Figure 1, the initial condition exhibits a smooth and radially symmetric profile, leading to continuous derivatives required for the computation of the Gaussian curvature.

$$f_0(x,y) = a(0)\,(x^2+y^2). \tag{34}$$

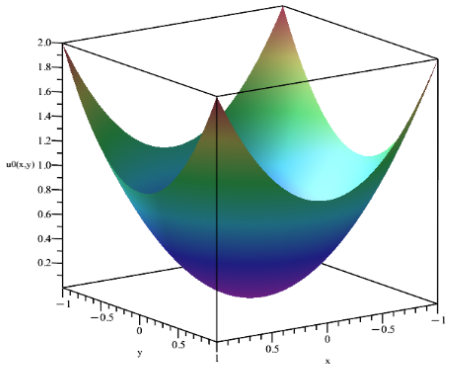


Figure 1: Manufactured smooth initial condition where $a(0) = 1$.

5.2 Boundary Conditions

Homogeneous Neumann boundary conditions are imposed on the boundary:

$$\frac{\partial u}{\partial n}(x,y,t) = 0,\qquad (x,y)\in\partial\Omega,\quad t\geq 0.$$

6 SHORT-TIME NUMERICAL SOLUTIONS

This section presents short-time numerical solutions for the proposed curvature-driven model. We compare the Adomian Decomposition Method of first and second order (ADM1 and ADM2) with a finite difference (FD) reference solution at a small time $T = 10^{-4}$. The numerical simulations are carried out on a uniform grid with $N_x = N_y = 81$ over the domain $\Omega = [-1,1] \times [-1,1]$. The corresponding spatial step sizes are $\Delta x = \Delta y = \frac{2}{(N_x-1)}$. For the time discretization, we use $\Delta t = 10^{-7}$, and all results are computed up to the short-time horizon $T = 10^{-4}$.

Figure 2 illustrates the evolution of the L^2 error norm as a function of time for the manufactured solution, comparing the first- and second-order Adomian decomposition methods (ADM1 and ADM2) with the finite difference (FD) method used as a reference. It is observed that the error grows smoothly in time on a logarithmic scale, consistent with the short-time expansion regime. Notably, both ADM approximations exhibit smaller error magnitudes and a more regular growth pattern compared to the FD method within the considered short-time interval. In particular, ADM 2 achieves the highest accuracy due to the inclusion of higher-order terms in the Adomian expansion, demonstrating the superiority of the Adomian-based approaches for short-time simulations.

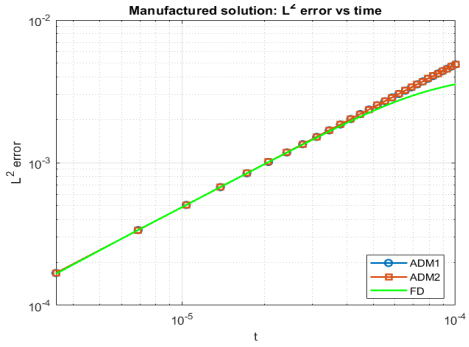


Figure 2: Comparison of the L^2 error versus time for the manufactured solution using ADM1, ADM 2, and the finite difference (FD) method.

The results, as shown in Table 1 and Table 2, indicate that both ADM1 and ADM2 follow a similar trend over time, with the error values increasing gradually. However, ADM2 demonstrates higher accuracy, as it consistently produces smaller errors than ADM1 at all considered time levels.

Table 1: L^2 error values corresponding to the Gaussian initial condition for several time levels using ADM1 and ADM2.

Time	ADM1(L^2)	$ADM2(L^2)$
2×10^{-4}	7.127×10^{-6}	7.380×10^{-7}
5×10^{-4}	4.092×10^{-5}	9.747×10^{-6}
1×10^{-3}	1.463×10^{-4}	6.323×10^{-5}

Table 2: L^∞ error values corresponding to the clipped initial condition for several time levels using ADM1 and ADM2.

Time	ADM1(L^∞)	$ADM2(L^\infty)$
1×10^{-4}	7.765×10^{-4}	1.322×10^{-4}
2×10^{-4}	2.839×10^{-3}	7.849×10^{-4}

We compare the short-time numerical solutions obtained using the first- and second-order Adomian Decomposition Methods (ADM1 and ADM2) at a short time $T = 10^{-4}$ for the Gaussian initial condition. Figure 3 shows the ADM1 and ADM2 solution surfaces. The two solutions are nearly indistinguishable visually, highlighting that for smooth initial data the curvature-driven flow is in the numerical implementation, these conditions are enforced using a reflection approach at the boundaries. Accurately captured by both Adomian approximations.

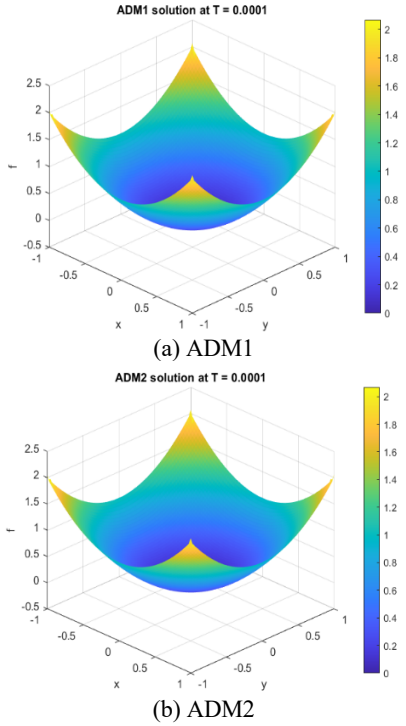
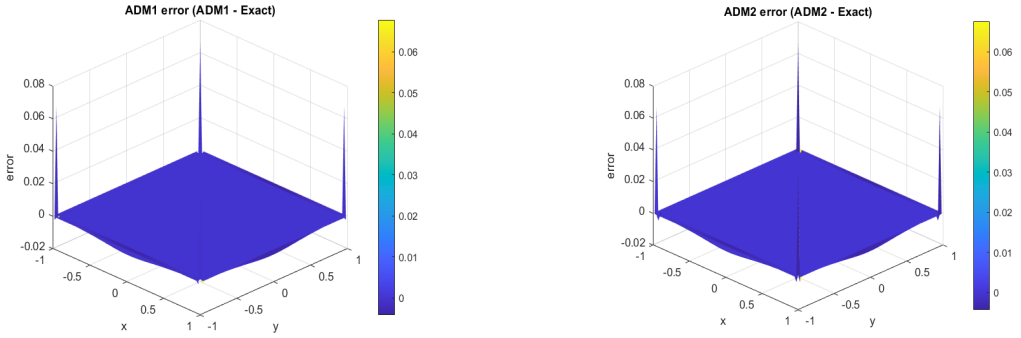


Figure 3: Short-time numerical solutions at $T = 10^{-4}$ for the Gaussian initial condition.

Figure 4 shows the spatial distribution of the numerical error at the short time $T = 10^{-4}$ using the Adomian Decomposition Method (ADM) with two truncation orders: the first-order scheme (ADM1) and the second-order scheme (ADM2). In both cases, the error remains small and smooth over the computational domain, and no significant difference is observed between the two surfaces. This indicates that both truncation orders provide comparable accuracy within the considered short-time regime for this smooth initial data.

Figure 5 illustrates that, at the short time $T = 10^{-4}$, the error fields corresponding to the Gaussian initial condition are small and exhibit smooth behavior. The left panel presents the error obtained using the finite difference (FD) method. This error is of low magnitude and follows a regular pattern without noticeable oscillations. The right panel shows the difference between the ADM2 and ADM1 solutions. The results are reported at $t = 10^{-3}$. These differences remain small and display a smooth pattern. This behavior suggests that both methods achieve similar levels of accuracy over the short-time regime

Figure 6 presents the exact solution along with the finite difference (FD) solution, as well as the error distributions associated with the first- and second-order Adomian Decomposition Method (ADM1 and ADM2). The errors remain small and smooth over the computational domain, indicating strong agreement between the numerical approaches and the exact solution over the short-time regime.



(a) ADM1 error (ADM1–exact) (b) ADM2 error (ADM2–exact)

Figure 4: Error surfaces at $T = 10^{-4}$ for the Gaussian initial condition.

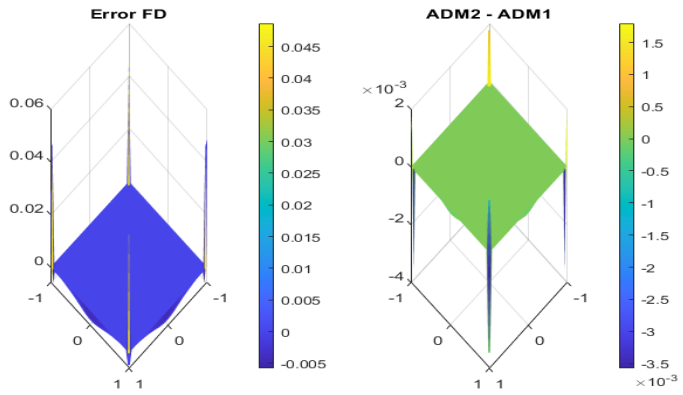


Figure 5: Difference fields at short time $T = 10^{-4}$ for the Gaussian initial condition. Left: FD error, Right: ADM2–ADM1 difference.

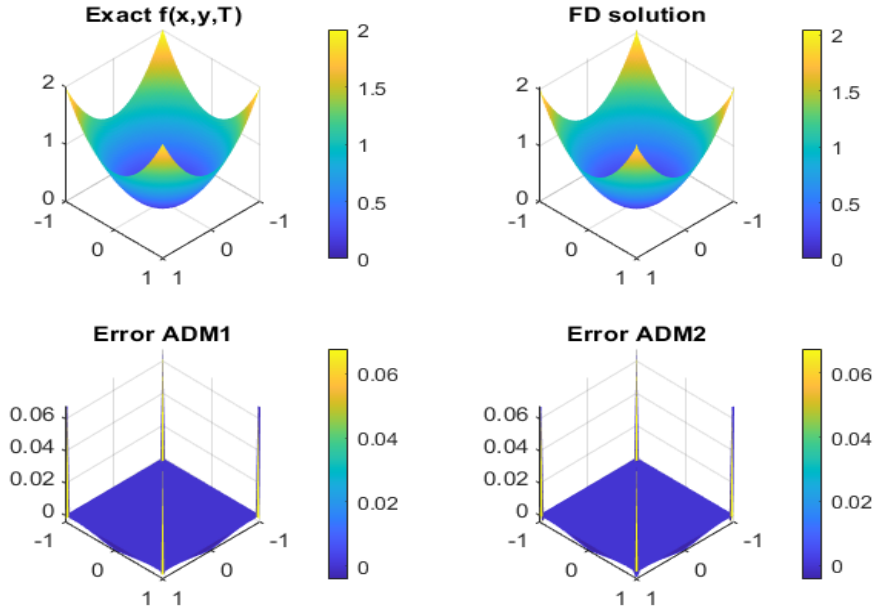


Figure 6: Exact and numerical solutions with corresponding error surfaces for the Gaussian initial condition at the short time $T = 10^{-4}$.

7 CONCLUSIONS

This study examines short-time numerical solutions of a two-dimensional curvature-driven flow using first- and second-order Adomian Decomposition schemes (ADM1 and ADM2), while the finite difference (FD) method is used as a reference at $T = 10^{-4}$.

The results show that both ADM1 and ADM2 provide excellent agreement with the FD solution for smooth initial data, particularly for the Gaussian initial condition. The numerical solution surfaces obtained by the two Adomian approximations are visually almost indistinguishable, indicating that the curvature-driven flow is accurately captured in the short-time regime.

Moreover, the corresponding error distributions remain small and smooth over the entire computational domain. No significant difference is Within the considered time interval, the difference between the first- and second-order truncations remains negligible, which reflects the similar behavior of ADM1 and ADM2. The results demonstrate that the Adomian Decomposition Method serves as a reliable and efficient approach for short-time simulations of curvature-driven flow. It delivers good accuracy while keeping the computational cost low.

REFERENCES

[1] W. W. Mullins, “Theory of thermal grooving,” J. Appl. Phys., vol. 28, no. 3, pp. 333-339, 1957, [Online]. Available: https://doi.org/10.1063/1.1722742.

[2] B. Andrews, “Gauss curvature flow: The fate of the rolling stones,” Math. Ann., vol. 138, pp. 151-161, 1999, [Online]. Available: https://doi.org/10.1007/s002229900004.

[3] H. Zhao and G. Xu, “Triangular surface mesh fairing via Gaussian curvature flow,” J. Comput. Appl. Math., vol. 195, no. 1-2, pp. 300-311, 2006, [Online]. Available: https://doi.org/10.1016/j.cam.2005.03.094.

[4] C. Brito-Loeza and K. Chen, “Fast iterative algorithms for solving the minimization of curvature-related functionals in surface fairing,” Int. J. Comput. Math., vol. 90, no. 1, pp. 92-108, 2013, [Online]. Available: https://doi.org/10.1080/00207160.2012.720370.

[5] E. Cinti, C. Sinestrari, and E. Valdinoci, “Convex sets evolving by volume-preserving fractional mean curvature flows,” Anal. PDE, vol. 13, no. 7, pp. 2149-2175, 2020, [Online]. Available: https://doi.org/10.2140/apde.2020.13.2149.

[6] V. Julin and D. La Manna, “Short time existence of the classical solution to the fractional mean curvature flow,” Ann. l’Institut Henri Poincaré C, Anal. Non Linéaire, vol. 37, no. 4, pp. 983-1016, 2020, [Online]. Available: https://doi.org/10.1016/j.anihpc.2020.02.007.

[7] A. Cesaroni and M. Novaga, “Fractional mean curvature flow of Lipschitz graphs,” Manuscripta Math., vol. 170, pp. 427-451, 2023, [Online]. Available: https://doi.org/10.1007/s00229-022-01371-5.

[8] Y. Noma, S. Sellán, N. Sharp, K. Singh, and A. Jacobson, "Surface-filling curve flows via implicit medial axes," ACM Trans. Graph., vol. 43, no. 4, pp. 1-12, 2024, [Online]. Available: https://doi.org/10.1145/3658158.

[9] Y. Cherruault and G. Adomian, "Decomposition methods: A new proof of convergence," Math. Comput. Model., vol. 18, no. 12, pp. 103-106, 1993, [Online]. Available: https://doi.org/10.1016/0895-7177(93)90233-O.

[10] E. Babolian and J. Biazar, "On the order of convergence of Adomian method," Appl. Math. Comput., vol. 130, no. 2-3, pp. 383-387, 2002, [Online]. Available: https://doi.org/10.1016/S0096-3003(01)00103-5.

[11] G. Adomian, "A new approach to nonlinear partial differential equations," J. Math. Anal. Appl., vol. 102, no. 2, pp. 420-434, 1984, [Online]. Available: https://doi.org/10.1016/0022-247X(84)90182-3.

[12] G. Adomian, "Explicit solutions of nonlinear partial differential equations," Appl. Math. Comput., vol. 88, no. 2-3, pp. 117-126, 1997, [Online]. Available: https://doi.org/10.1016/S0096-3003(96)00141-5.

[13] A. Jimoh and A. Oyedeji, "On Adomian decomposition method for solving nonlinear ordinary differential equations of variable coefficients," Open J. Math. Sci., vol. 4, pp. 476-484, 2020, [Online]. Available: https://doi.org/10.30538/oms2020.0138.

[14] D. A. Maturi and H. M. Malaikah, "The Adomian decomposition method for solving nonlinear partial differential equation using Maple," Adv. Pure Math., vol. 11, no. 6, pp. 595-603, 2021, [Online]. Available: https://doi.org/10.4236/apm.2021.116038.

[15] A. Mohammadein, M. El-Amin, E. Abo-Eldahab, and H. Ali, "Adomian decomposition method for time fractional diffusion-reaction equation," Delta J. Sci., vol. 51, no. 1, pp. 130-140, 2025.

[16] R. D. Richtmyer and E. H. Dill, "Difference methods for initial-value problems," Phys. Today, vol. 12, no. 4, pp. 50-50, 1959, [Online]. Available: https://doi.org/10.1063/1.3060778.

[17] R. J. LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equations, Cambridge, U.K.: Cambridge Univ. Press, 2007, [Online]. Available: https://doi.org/10.1137/1.9780898717839.

[18] J. W. Thomas, Numerical Partial Differential Equations: Finite Difference Methods, vol. 22, New York, NY, USA: Springer Science & Business Media, 2013, [Online]. Available: https://doi.org/10.1007/978-1-4899-7278-1.

[19] S. Çavuşoğlu, F. S. Muxtarov, and O. S. Mukhtarov, "A new modification of the finite difference method for solving transmission problems for two-interval differential equations," Carpath. Math. Publ., vol. 16, no. 2, pp. 575-586, 2024, [Online]. Available: https://doi.org/10.15330/cmp.16.2.575-586.

[20] N. Sahoo and R. Singh, "A stable higher-order numerical method for solving a system of third-order singular Emden-Fowler type equations," J. Appl. Math. Comput., vol. 71, pp. 387-414, 2025, [Online]. Available: https://doi.org/10.1007/s12190-024-02233-x.

[21] N. Sahoo and R. Singh, "Compact finite difference schemes and error estimation for third-order Emden-Fowler equations," Numer. Algor., vol. 101, pp. 847-882, 2026, [Online]. Available: https://doi.org/10.1007/s11075-025-02025-9.