Computational Mathematics Today
eISSN: 3122-7007
https://cmt.cultechpub.com/index.php/cmt

Computational Methods for Differential Equations: Numerical Solutions and Stability Analysis

Louise Hugh Zoe*

Department of Applied Mathematics, The University of Adelaide, Adelaide, Australia

*Corresponding author: Louise Hugh Zoe, louise_zoe@student.adelaide.edu.au

Abstract

The numerical solution of differential equations is fundamental to simulating complex phenomena across science and engineering. While analytical solutions are limited, robust computational methods are essential for tackling nonlinear, high-dimensional problems defined on complex geometries. This article provides a comprehensive review and comparative analysis of core computational methods for differential equations, with a focused emphasis on the critical role of stability as the cornerstone of reliable simulation. We systematically dissect key discretization techniques-Finite Difference Methods (FDMs), Finite Element Methods (FEMs), and Spectral Methods-detailing their implementation, inherent properties, and suitability for different problem classes. The core of our discussion establishes the theoretical framework of consistency, stability, and convergence via the Lax Equivalence Theorem, and then delves into practical stability analysis tools such as von Neumann (Fourier) analysis, the Courant-Friedrichs-Lewy (CFL) condition, and energy methods. Through illustrative examples and comparative tables, we quantitatively analyze the stability regions, accuracy order, and computational cost of various schemes for model parabolic and hyperbolic equations. The results unequivocally demonstrate that stability dictates practical algorithm selection and parameter tuning (e.g., time-step ΔtΔt). Finally, we comment on current challenges and emerging trends, including adaptive mesh refinement, the handling of stiff and high-dimensional problems, and the promising integration of machine learning techniques for developing novel, stable discretizations. This work serves as both a tutorial on foundational concepts and a reference for practitioners in choosing the appropriate, stability-guaranteed tool for their specific computational problem.

Keywords

Differential Equations, Numerical Solutions, Stability Analysis, Finite Difference Method, Finite Element Method, CFL Condition, Von Neumann Analysis

1. Introduction

Differential equations are the language of change, modeling everything from celestial mechanics and fluid dynamics to financial markets and epidemiological spread. While analytical solutions offer profound insight, they are only attainable for a limited class of linear or simplified problems. The vast majority of physically and scientifically relevant differential equations are nonlinear, high-dimensional, or defined on complex geometries, necessitating robust numerical approaches [1].

The advent of high-performance computing has transformed numerical simulation from a specialist tool into a ubiquitous methodology. However, the path from a continuous differential equation to a discrete, computable algorithm is fraught with potential pitfalls. A naive discretization can lead to solutions that diverge explosively, oscillate wildly, or converge to an incorrect result, despite seeming conceptually sound [2]. This underscores the paramount importance of *stability* and *convergence* analysis in computational mathematics.

This article aims to bridge the gap between the formulation of a numerical scheme and its successful implementation. We will survey essential methods for time-dependent (parabolic, hyperbolic) and stationary (elliptic) problems. Our central thesis is that *stability is the sine qua non for effective numerical simulation*. A method must not only approximate the equation locally (consistency) but must also control the amplification of errors over many computational steps [3]. We will explore how tools like the von Neumann analysis and the CFL condition provide vital criteria for selecting appropriate time steps and spatial discretizations.

The structure is as follows: Section 2 introduces core discretization frameworks. Section 3 delves into the fundamentals of consistency, stability, and the pivotal Lax Equivalence Theorem. Section 4 is dedicated to stability analysis techniques for various problem types. Section 5 presents practical implementations and comparative results via tables. The historical evolution of numerical methods for differential equations is itself a testament to the interplay between mathematical insight and computational advancement. Early work by Euler and Runge laid the groundwork for time-stepping methods, while the mid-20th century saw a explosion of activity with the development of stability theory, finite element methods, and the seminal work on stiffness. The late 20th and early 21st centuries have been characterized by the quest for higher-order accuracy, robustness for nonlinear problems, and algorithms tailored for massively parallel architectures. This article not only reviews these foundational pillars but also seeks to contextualize them within the modern computational landscape, where the choice of method is increasingly dictated by a triad of constraints: theoretical guarantee, implementation complexity, and scalable performance on heterogeneous hardware. Understanding stability is the common thread that binds these considerations, preventing computationally expensive yet futile simulations. We conclude in Section 6 with perspectives on future directions [4].

2. Core Numerical Discretization Methods

2.1 Finite Difference Methods (FDMs)

FDMs are the most intuitive approach for problems on simple domains. Derivatives in the PDE are replaced by differences of values on a structured grid. For a function $u(x,t)u(x,t)$, common approximations include:

•Forward Euler (explicit): $ut{\approx}(uin{+}1{-}uin)/\Delta tut \quad {\approx}(uin{+}1 \quad {-}uin \quad)/\Delta t$

•Backward Euler (implicit): $ut{\approx}(uin{-}uin{-}1)/\Delta tut \quad {\approx}(uin \quad {-}uin{-}1 \quad)/\Delta t$

•Central Difference: $uxx{\approx}(ui{+}1n{-}2uin{+}ui{-}1n)/(\Delta x)2uxx \quad {\approx}(ui{+}1n \quad {-}2uin \quad {+}ui{-}1n \quad)/(\Delta x)2$

While explicit methods are simple and computationally cheap per step, they impose severe stability restrictions. Implicit methods, which require solving a linear system each step, are typically unconditionally stable for parabolic problems but are more complex to implement [5].

Beyond the basic explicit/implicit dichotomy, modern FDM variants address specific challenges. High-Order Compact Schemes obtain superior accuracy on smaller stencils by solving a linear system for derivatives at each node, improving resolution for wave propagation problems. For conservation laws, Finite Volume Methods (FVMs) are the dominant FDM-derived approach [6]. They discretize the integral form of the PDE over control volumes, ensuring discrete conservation of quantities like mass, momentum, and energy-a crucial property for shock-capturing in fluid dynamics. The Method of Lines (MOL) is another powerful paradigm where spatial derivatives are discretized first (via FDM, FEM, or spectral methods), converting the PDE into a large system of ODEs in time, which can then be tackled with sophisticated ODE integrators.

2.2 Finite Element Methods (FEMs)

FEMs are the method of choice for complex geometries and irregular boundaries. The solution is approximated as a linear combination of basis functions (e.g., piecewise polynomials) defined on a mesh of elements. The method operates on the weak (integral) form of the PDE, making it naturally suited for problems with conservation laws or heterogeneous material properties. The flexibility of FEM comes at the cost of greater algorithmic and implementation complexity compared to FDMs. The evolution of FEM has branched into several specialized directions. Discontinuous Galerkin (DG) Methods combine ideas from FEM and FVMs, using piecewise polynomial approximations that are discontinuous across element boundaries [7]. This grants them exceptional flexibility for handling hyperbolic problems, adaptive hp-refinement, and achieving high-order accuracy on complex geometries. Isogeometric Analysis (IGA) seeks to bridge the gap between CAD geometry and simulation by using the same basis functions (e.g., NURBS) for both geometric representation and numerical analysis, eliminating meshing errors and streamlining the design-to-analysis workflow. These advancements highlight FEM's ongoing adaptation to the demands of high-fidelity, integrated computational engineering.

2.3 Spectral Methods

Spectral methods use global basis functions (e.g., Fourier series for periodic problems, Chebyshev polynomials for non-periodic ones) to achieve exponential ("spectral") convergence for smooth solutions. They are highly accurate for problems with simple boundaries and smooth data but can be susceptible to stability issues for problems with discontinuities or sharp gradients, often requiring specialized filtering or shock-capturing techniques [8]. The main challenge for spectral methods-handling non-smooth solutions-has spurred significant innovation. Spectral Element Methods (SEMs) marry the geometric flexibility of finite elements with the high accuracy of spectral methods within each element. Filtering and Post-Processing techniques, such as the application of a weak exponential filter in Fourier space or Gegenbauer reconstruction for piecewise smooth functions, are often essential to stabilize computations and recover accuracy near discontinuities. Furthermore, Penalty Methods and Collocation Methods on non-uniform grids (like Chebyshev-Gauss-Lobatto points) provide robust frameworks for imposing boundary conditions in spectral approximations, extending their applicability to a broader class of problems [9].

3. Foundations: Consistency, Stability, and Convergence

The celebrated Lax Equivalence Theorem provides the foundational framework: *For a consistent linear approximation, stability is the necessary and sufficient condition for convergence.*

•Consistency: The local truncation error of the discretization tends to zero as the grid is refined ($\Delta t,\Delta x\to 0\Delta t,\Delta x\to 0$). It measures how well the discrete scheme approximates the continuous PDE at a single step.

•Stability: The numerical scheme does not amplify errors (from initial data, rounding, etc.) unboundedly as the computation proceeds. It is a global property of the algorithm.

•Convergence: The numerical solution tends to the true solution of the PDE as the grid is refined.

Thus, the primary task in analyzing a linear scheme is to prove stability; convergence then follows from consistency. For nonlinear problems, stability analysis remains crucial but is often more heuristic, relying on linearized analyses or energy principles.

4. Stability Analysis in Practice

4.1 Von Neumann (Fourier) Analysis

This is the most common tool for analyzing the stability of linear FDMs on periodic domains. The error is decomposed into Fourier modes. By inserting a trial solution of the form $ujn=\xi neikj\Delta xujn\ =\xi neikj\Delta x$ into the *difference* equation, one solves for the amplification factor $G(k,\Delta t,\Delta x)G(k,\Delta t,\Delta x)$. The stability condition is $|G|\leq 1|G|\leq 1$ for all relevant wavenumbers kk. For example, applying this to the explicit FTCS scheme for the 1D heat equation yields the stability restriction $\mu=\alpha\Delta t/(\Delta x)2\leq 1/2\mu=\alpha\Delta t/(\Delta x)2\leq 1/2$.

4.2 Courant-Friedrichs-Lewy (CFL) Condition

For hyperbolic equations (e.g., wave, advection), the CFL condition is a necessary condition for stability. It states that the numerical domain of dependence must contain the physical domain of dependence. For a wave speed cc, the explicit 1D scheme requires $C=|c|\Delta t/\Delta x\leq 1C=|c|\Delta t/\Delta x\leq 1$. This number C$C$ is the famous Courant number. Violating the CFL condition makes it impossible for the scheme to access the information needed to construct the correct solution [10].

4.3 Energy Methods

For problems with a natural energy functional (common in elliptic and some parabolic equations), one can analyze stability by showing that a discrete analog of this energy is non-increasing in time. This method is powerful for analyzing implicit schemes and FEM formulations, often proving unconditional stability.

4.4 Stiff Equations and A-Stability

Systems of ODEs or PDEs where different components evolve on vastly different timescales are termed "stiff." Explicit methods fail catastrophically here, requiring prohibitively small time steps to maintain stability. The concept of A-stability is key: an A-stable method has a stability region that contains the entire left half of the complex plane. The implicit Euler and Crank-Nicolson methods are A-stable, making them indispensable for stiff problems [11].

4.5 Nonlinear Stability and Contractivity

The analyses in Sections 4.1-4.4 primarily address linear or linearized stability. For genuinely nonlinear problems, more nuanced concepts are required. Total Variation Diminishing (TVD) schemes are a cornerstone for nonlinear hyperbolic conservation laws. A TVD scheme ensures that the total variation of the numerical solution does not increase in time, which rigorously prevents the creation of new spurious oscillations and allows convergence to the physically relevant entropy solution.

A related concept for ODEs and semi-discrete PDE systems is contractivity or B-stability. A numerical method is B-stable if, when applied to a dissipative system (one satisfying a one-sided Lipschitz condition), the distance between any two numerical solutions is non-increasing. This is a strong, nonlinear stability property. While implicit Runge-Kutta methods like Gauss-Legendre can be B-stable, explicit methods generally are not [12]. The search for methods that are both high-order accurate and possess strong nonlinear stability properties (e.g., Strong Stability Preserving (SSP) Runge-Kutta methods) is an active area of research, crucial for simulating problems with sharp fronts and discontinuities without introducing non-physical oscillations [13].

5. Comparative Analysis and Numerical Examples

To illustrate the practical implications, we analyze two model problems: the 1D heat equation (parabolic) and the 1D linear advection equation (hyperbolic).

Table 1. Stability and properties of common schemes for model PDEs.

Scheme	PDE Type	Formulation	Stability Condition	Key Properties
FTCS	Parabolic	Explicit	*μ*≤0.5*μ*≤0.5	Simple, conditionally stable, prone to oscillations.
Crank-Nicolson	Parabolic	Implicit	Unconditionally stable	2nd-order accurate in time & space, G*G* often has unit magnitude leading to non-dissipative errors.
Implicit Euler	Parabolic	Implicit	Unconditionally stable	1st-order accurate, strongly dissipative (L-stable), excellent for stiff problems.
Upwind (1st order)	Hyperbolic	Explicit	*C*≤1*C*≤1 (*CFL*)	Stable, introduces numerical diffusion, 1st-order accurate.
Lax-Wendroff	Hyperbolic	Explicit	*C*≤1*C*≤1 (*CFL*)	2nd-order accurate, introduces numerical dispersion (oscillations near shocks).
Leapfrog	Hyperbolic	Explicit	*C*≤1*C*≤1 (*CFL*)	2nd-order, non-dissipative, prone to decoupling of odd/even steps.

Table 1 provides a comparative summary of several commonly used numerical schemes for solving model partial differential equations (PDEs), focusing on their stability characteristics and key properties. Each scheme is categorized by the type of PDE it typically solves (parabolic or hyperbolic), whether its formulation is explicit or implicit, the stability conditions required for reliable results, and its main numerical behaviors.

For parabolic PDEs such as the heat equation, the FTCS method is shown to be explicit and only conditionally stable, meaning it works correctly only when the nondimensional parameter μ\muμ satisfies *μ*≤0.5*mu* *le* 0.5*μ*≤0.5. Although simple to implement, FTCS can produce oscillations when stability is marginal. In contrast, the Crank–Nicolson and implicit Euler methods are implicit schemes that are unconditionally stable, meaning they remain stable regardless of the time-step size. Crank–Nicolson offers second-order accuracy in both time and space but may produce non-dissipative errors, while implicit Euler is first-order but strongly dissipative, making it suitable for stiff problems [14].

The lower part of the table summarizes explicit schemes commonly used for hyperbolic PDEs such as advection equations. The first-order upwind method is stable under the Courant–Friedrichs–Lewy (CFL) condition *C*≤1*C* *le* 1*C*≤1 and introduces numerical diffusion, which smooths the solution. Lax–Wendroff and leapfrog are both second-order accurate but behave differently: Lax–Wendroff can introduce numerical dispersion, producing oscillatory behavior near sharp gradients or shocks, while leapfrog is non-dissipative but susceptible to instabilities related to odd–even decoupling [15].

Overall, the table highlights the trade-offs between stability, accuracy, and numerical artifacts across various schemes, helping to clarify why different methods are chosen for different classes of PDEs.

Table 2. Error and performance comparison for 1D heat equation (at t=0.5).

Method (ΔxΔ*x*)	ΔtΔ*t*	L2*L*2 Error	Order (in space)	CPU Time (s)	Stability Observed?
FTCS (0.02)	0.0002	4.21×10−44.21×10−4	~2.0	1.2	Yes (*μ*=0.5*μ*=0.5)
FTCS (0.02)	0.0005	Diverged	-	-	No (*μ*=1.25*μ*=1.25)
Crank-Nicolson (0.02)	0.001	1.05×10−51.05×10−5	~2.0	0.8	Yes
Implicit Euler (0.02)	0.005	8.76×10−48.76×10−4	~1.0	0.5	Yes

Table 2 compares the numerical accuracy, computational cost, and stability properties of several finite-difference methods applied to the 1-D heat equation at time t=0.5t = 0.5t=0.5. Each method is evaluated using the same spatial grid size (Δ*x*=0.02*Delta x* = 0.02Δ*x*=0.02), while different time steps (Δ*t**Delta t*Δ*t*) are tested to observe their effect on performance.

The columns show four major metrics. The L2L2 error measures how closely each numerical solution matches the analytical solution; smaller values indicate higher accuracy. The order (in space) indicates the expected spatial convergence rate of each method. CPU time reflects the computational efficiency, showing how long the simulation takes to run. Finally, the Stability Observed? column records whether the chosen method and time step satisfy the scheme's stability condition.

The results illustrate typical behaviors of explicit and implicit schemes. The FTCS method is stable when the CFL condition is satisfied (Δ*t* = 0.0002), producing a small error of about 4.2×10−44.2*times*10^{−4}4.2×10−4. However, when Δt increases to 0.0005-violating the stability condition-the method diverges. In contrast, the Crank–Nicolson and implicit Euler methods remain stable even with larger time steps. Crank–Nicolson achieves high accuracy and second-order spatial convergence, while implicit Euler converges more slowly but requires less computational time. Overall, the table highlights the trade-off among accuracy, stability, and computational cost across different numerical schemes.

Table 1 provides a theoretical summary, while Table 2 shows a practical computational experiment. The divergence of FTCS when the stability condition is violated is stark. Crank-Nicolson achieves higher accuracy with a larger time step, showcasing the benefit of an unconditionally stable method. Implicit Euler, while less accurate per time step, allows the largest step of all, demonstrating its robustness.

The comparison in Tables 1 and 2, while informative, primarily focuses on deterministic, low-dimensional model problems. In practice, the choice of method is heavily influenced by problem dimensionality and stochasticity. Table 3 summarizes this expanded perspective. For high-dimensional parabolic PDEs (e.g., arising in quantum chemistry or financial mathematics), traditional mesh-based methods suffer from the "curse of dimensionality." Here, Mesh-Free Methods (like Radial Basis Function methods) or probabilistic approaches (like Monte Carlo methods for linear parabolic equations) become viable. Conversely, for nonlinear hyperbolic systems in multiple dimensions (e.g., compressible Euler equations), the need for robustness leads to the dominance of High-Resolution Shock-Capturing Schemes (a class of TVD or WENO methods) implemented within a Finite Volume framework, despite their computational intensity per cell.

Table 3. Method suitability for advanced problem classes.

Problem Class	Exemplar Equation	Dominant Challenges	Recommended Method Class
High-Dimensional Parabolic	Black-Scholes (d>3), Fokker-Planck	Curse of dimensionality, boundary treatment	Probabilistic (Monte Carlo), Tensor Decomposition, Deep Neural Networks
Nonlinear Hyperbolic Systems	Euler Equations, Shallow Water	Shock formation, entropy violation, multi-dimensional waves	High-Resolution Finite Volume (TVD/WENO), Discontinuous Galerkin
Multiscale/Stiff PDEs	Reaction-Diffusion (e.g., Turing patterns), Chemical Kinetics	Widely separated time scales, pattern formation	Implicit-Explicit (IMEX) Methods, Operator Splitting, Exponential Integrators
PDEs on Complex Geometries	Stokes Flow, Structural Mechanics	Mesh generation, boundary conformity, preserving invariants	Finite Element Method (FEM), Isogeometric Analysis (IGA), Immersed Boundary Methods

Table 3 summarizes which numerical methods are best suited for different classes of advanced partial differential equation (PDE) problems, based on the typical equations involved, the major computational challenges they present, and the method families that are generally recommended.

The first row concerns high-dimensional parabolic PDEs, such as Black–Scholes equations in more than three dimensions or the Fokker–Planck equation. These problems suffer from the "curse of dimensionality," meaning computational cost grows exponentially with dimension. As a result, traditional grid-based solvers become inefficient, and methods based on probabilistic Monte Carlo sampling, tensor decomposition, or deep neural networks are preferred.

The second row addresses nonlinear hyperbolic systems, represented by the Euler equations and shallow-water models. Their main challenges include shock formation, potential violation of entropy conditions, and the need to correctly capture multidimensional wave structures. Because of these complexities, high-resolution finite-volume schemes such as TVD or WENO, as well as discontinuous Galerkin methods, are recommended.

The third row describes multiscale or stiff PDEs, such as reaction–diffusion equations (including Turing pattern models) and chemical kinetics. These equations involve widely separated time scales and may exhibit emergent spatial patterns. To handle stiffness efficiently, implicit-explicit (IMEX) schemes, operator-splitting techniques, and exponential integrators provide effective numerical strategies.

The final row focuses on PDEs defined on complex geometries, such as those arising in Stokes flow or structural mechanics. The difficulties here include generating high-quality meshes, maintaining geometric conformity, and preserving physical invariants. Consequently, geometric-flexible methods-such as the finite element method (FEM), isogeometric analysis (IGA), and immersed boundary methods-are most suitable.

6. Conclusion and Future Perspectives

Stability analysis remains the bedrock of reliable computational differential equations. As we have shown, understanding the stability properties of a numerical scheme is not an optional theoretical exercise but a critical step in algorithm selection and parameter tuning (like $\Delta t\Delta t$ and $\Delta x\Delta x$). The interplay between accuracy, stability, and computational cost, as summarized in our tables, guides practitioners in choosing the right tool for their specific problem.

The comparative analysis presented in Sections 5 and Table 3 underscores a paradigm shift: the "optimal" numerical method is inherently context-dependent. For the model analyst working on canonical problems, classical FDMs with clear stability bounds may suffice. For the computational engineer simulating turbulent flow over an aircraft, a sophisticated blend of FVMs, DES/LES turbulence models, and parallel computing is mandatory, where stability is enforced through a combination of nonlinear limiters and implicit time-marching. The computational mathematician developing solvers for emerging problems in biology or data science must now consider paradigms where stability is learned from data or enforced through novel geometric constraints. Thus, the practitioner's toolkit must contain not only a repertoire of methods but also a deep understanding of the stability principles that underpin them, enabling informed adaptation and hybrid approaches.

Future directions are vibrant. Adaptive mesh refinement (AMR) and adaptive time-stepping algorithms dynamically adjust resolution based on local error or stability estimators. For multiscale and stochastic PDEs, heterogeneous computing (GPUs, TPUs) is enabling previously intractable simulations. Perhaps most intriguing is the incursion of machine learning. Neural networks are being used to learn stable discretizations, accelerate iterative solvers for implicit schemes, and develop closure models for turbulent flows, opening a new chapter in the enduring quest to solve the equations that describe our world.

References

[1] Lax, P. D., & Richtmyer, R. D. (1956). Survey of the stability of linear finite difference equations. Communications on Pure and Applied Mathematics, 9(2), 267-293. https://doi.org/10.1002/cpa.3160090206

[2] Courant, R., Friedrichs, K., & Lewy, H. (1928). Über die partiellen Differenzengleichungen der mathematischen Physik. Mathematische Annalen, 100(1), 32-74. https://doi.org/10.1007/BF01448839

[3] Dahlquist, G. (1963). A special stability problem for linear multistep methods. BIT Numerical Mathematics, 3(1), 27-43. https://doi.org/10.1007/BF01963532

[4] LeVeque, R. J. (2007). Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems. Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9780898717839

[5] Hairer, E., & Wanner, G. (1996). Solving ordinary differential equations II: Stiff and differential-algebraic problems (2nd ed.). Springer-Verlag. https://doi.org/10.1007/978-3-662-09947-6

[6] Trefethen, L. N. (2000). Spectral methods in MATLAB. Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9780898719598

[7] Iserles, A. (2009). A first course in the numerical analysis of differential equations (2nd ed.). Cambridge University Press. https://doi.org/10.1017/CBO9780511995569

[8] Morton, K. W., & Mayers, D. F. (2005). Numerical solution of partial differential equations: an introduction (2nd ed.). Cambridge University Press. https://doi.org/10.1017/CBO9780511812248

[9] Strikwerda, J. C. (2004). Finite difference schemes and partial differential equations (2nd ed.). Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9780898717938

[10] Quarteroni, A. (2017). Numerical models for differential problems (3rd ed.). Springer. https://doi.org/10.1007/978-88-470-5522-3

[11] Zienkiewicz, O. C., Taylor, R. L., & Zhu, J. Z. (2013). The finite element method: its basis and fundamentals (7th ed.). Butterworth-Heinemann. https://doi.org/10.1016/B978-1-85617-633-0.00001-0

[12] Ascher, U. M., & Greif, C. (2011). A first course in numerical methods. Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9780898719987

[13] Butcher, J. C. (2016). Numerical methods for ordinary differential equations. John Wiley & Sons. https://doi.org/10.1002/9781119121534

[14] Durran, D. R. (2010). Numerical methods for fluid dynamics: with applications to geophysics (2nd ed.). Springer. https://doi.org/10.1007/978-1-4419-6412-0

[15] Gottlieb, S., & Shu, C. W. (1998). Total variation diminishing Runge-Kutta schemes. Mathematics of Computation, 67(221), 73-85. https://doi.org/10.1090/S0025-5718-98-00913-2