Comparative Analysis of RK2, ETD2RK, and GRK2 Methods for Stiff Systems:

Applications to Lotka–Volterra and van der Pol

Leo Kim (260914383) Prof. Sangyong Jeon

McGill University, Department of Physics

February 25, 2025

Abstract

We compare three numerical methods—Runge–Kutta 2 (RK2), the second-order variant of Exponential Time Differencing (ETD2RK), and Gaussian Runge–Kutta 2 (GRK2)—by applying them to two classic stiff problems: the Lotka–Volterra predator–prey model and the van der Pol oscillator. By adjusting the parameters in each system, we introduce mild, moderate, and strong stiffness regimes, then examine the performance of each method under various time-step sizes. Our results show that at lower stiffness levels with sufficiently small step sizes, all three methods accurately reproduce the expected oscillations without instability. However, in more challenging regimes or with larger time steps, either RK2 or ETD2RK may diverge, while GRK2 remains stable at the cost of additional amplitude and phase distortions. This trade-off indicates that RK2 is the most accurate when it avoids blow-up, ETD2RK offers partial stabilization but can fail if the stiff term varies rapidly, and GRK2 robustly handles significant changes in stiffness, albeit with some numerical damping. We conclude that no single solver is universally optimal, and the best choice depends on the degree of stiffness, the allowable step size, and the user's tolerance for small errors versus blow-ups.

Contents

1 Introduction

A system of differential equations is called *stiff* when it involves widely separated time scales. In these cases, explicit numerical methods—such as classical Runge–Kutta schemes—often become impractical because their stability requirements force extremely small time steps. Essentially, if a system has terms with large negative eigenvalues or very rapid transients, a naive explicit solver must use a sufficiently small Δt to keep all modes from blowing up, even if many of those modes quickly decay or are not of primary interest. Trying to take larger time steps under these conditions can make an explicit method diverge rapidly.

A common way to circumvent such severe step-size restrictions is to split the ODE into a "stiff" component, which is approximated, and a "non-stiff" or "source" component, which is treated explicitly. One well-known example of this approach is *Exponential Time Differencing* (ETD), introduced by Cox and Matthews. The basic idea is to rewrite an equation of the form

$$\frac{dx}{dt} = -p_n x + S, \tag{1}$$

where p_n is an approximation of the stiff coefficient over the interval, and S contains the remaining terms [1]. In the second-order variant, called ETD2RK, the stiff term $-p_n x$ is approximated as a constant, while the non-stiff term $\tilde{S}(t,x)$ is approximated via second-order Runge–Kutta interpolation. This follows the formulation of Cox and Matthews:

$$x(t_{n+1}) = x(t_n)e^{p_n h} + e^{p_n h}\int_0^h e^{-p_n \tau} S(x(t_n+\tau), t_n+\tau), d\tau, \tag{2}$$

where h is the time-step size, and the integral accounts for the contribution of the source term S [2]. The advantage of ETD2RK is that it allows somewhat larger h than standard explicit methods. However, if $p(t)$ varies significantly within a single step, treating the stiff term as constant can introduce amplitude and phase errors, or even lead to instability.

To address this, a newer scheme called *Gaussian Runge–Kutta 2 (GRK2)* has been proposed. Rather than assuming $p(t)$ remains constant over each step, GRK2 uses a linear interpolation,

$$p(\tau) = p_n + \frac{p_{n+1} - p_n}{h}(\tau - t_n), \tag{3}$$

where $p_n = p(t_n)$ is the value of the stiff coefficient at the start of the interval, and p_{n+1} may be taken as $p(t_{n+1})$ or obtained from a predictor step. By letting $p(t)$ vary linearly in $[t_n, t_{n+1}]$ GRK2 integrates this linear approximation of the stiff component exactly, leading to an error-function expression instead of a single exponential factor [1]. This approach enables the method to adapt more accurately to changes in stiffness over the interval. The evolution equation in GRK2 follows a modified ETD-like form:

$$x_{n+1} = e^{\frac{-p_n + p_{n+1} + 1}{2}} x_n + \int_0^{t_h} d\rho e^{-b\rho - a\rho^2} S(\rho), \tag{4}$$

which accounts for variations in $p(t)$ [1]. In this way, GRK2 provides a more stable alternative to ETD2RK in situations where the stiff coefficient changes rapidly.

To evaluate the performance of these three methods—RK2, ETD2RK, and GRK2—we apply them to two test problems made stiff by tuning their parameters: the Lotka–Volterra predator–prey model and the van der Pol oscillator. By observing whether the numerical solutions blow up, decay incorrectly, or shift their oscillation frequency, we highlight each method's trade-offs in accuracy, stability, and efficiency. The next section describes the equations, the relevant parameter variations, and the implementation of each solver.

2 Methods

Test Problems

To compare the performance of RK2, ETD2RK, and GRK2, we apply them to two standard differential equation systems that become stiff for certain parameter choices. The first is the *Lotka–Volterra* predator–prey system:

$$\begin{cases} x'(t) = \alpha x - \beta xy, \\ y'(t) = -\gamma y + \delta xy. \end{cases} \tag{5}$$

Here, $x(t)$ represents the prey population and $y(t)$ the predator population [3]. For small values of $\alpha, \beta, \gamma, \delta$, the system exhibits relatively smooth oscillations. However, increasing these parameters intensifies the interaction rates between the coupled equations, creating faster transients and making the system stiffer.

We also examine the *van der Pol* oscillator:

$$\begin{cases} x'(t) = y(t), \\ y'(t) = \mu(1 - x^2)y - x. \end{cases} \tag{6}$$

The parameter μ determines the strength of nonlinear damping [4]. When μ is small, the system behaves like a perturbed harmonic oscillator. As μ increases, it develops pronounced *relaxation oscillations*, alternating between slow and fast phases, making the problem increasingly stiff.

Parameter Ranges and Step Sizes

For each model, we define three stiffness levels: "less stiff," "normal," and "more stiff." In the Lotka–Volterra system, we raise $(\alpha, \beta, \gamma, \delta)$ step by step to induce more rapid predator–prey interactions. In the van der Pol model, we vary μ from 1 to 10 to see how the solvers handle progressively stronger stiffness. We integrate each system from $t = 0$ to a chosen final time, using fixed time steps $h = 0.1, 0.05, 0.01$. This setup lets us observe solver behaviour at relatively large steps (where instability is more likely) and at smaller steps (where all methods should, in principle, converge to a similar solution).

Solver Implementations

We implemented all three solvers in C, each advancing the solution by a single time step. RK2 (Heun's method) follows the two-stage predictor–corrector formula:

$$y_{n+1} = y_n + \frac{h}{2}\Big[f(t_n, y_n) + f(t_n + h, y_n + hf(t_n, y_n))\Big], \tag{7}$$

which remains stable only if h is sufficiently small for stiff problems [2].

ETD2RK, as formulated by Cox and Matthews, integrates the stiff component using an exponential factor while approximating the non-stiff term with a second-order scheme. Its update rule is:

$$x_{n+1} = a_n + \big(S(a_n, t_n + h) - S_n\big)\frac{e^{p_n h} - 1 - hc}{hp_n^2}, \tag{8}$$

where the correction accounts for the change in $\tilde{S}$ during the step [2]. Although this approach permits moderately larger h than explicit methods, it can introduce phase errors or instability when $p(t)$ varies substantially within a step.

GRK2 Formulation

GRK2 refines ETD2RK by allowing the stiff coefficient $p(t)$ to vary linearly over each time step instead of assuming it is constant. Concretely, on the interval $[t_n, t_{n+1}]$, we write:

$$p(\tau) = p_n + \frac{p_{n+1} - p_n}{h}(\tau - t_n). \tag{9}$$

Integrating the stiff term under this linear interpolation leads to an integral in terms of error functions instead of a single exponential factor. Meanwhile, the non-stiff portion $S(t, x)$ is handled with a second-order Runge–Kutta-like approach, similar to ETD2RK but accommodating the changing $p(\tau)$. In practice, we introduce temporary or predicted values $\tilde{p}_{n+1}$ and $\tilde{S}_{n+1}$ to indicate that these quantities are evaluated using a predictor estimate of x_{n+1}. Specifically, (1) $S_n = S(x_n, t_n)$ is the non-stiff term at the current step, (2) $\tilde{S}_{n+1} = S(\tilde{x}_{n+1}, t_{n+1})$ is the non-stiff term evaluated at a predicted solution $\tilde{x}_{n+1}$, (3) $p_n = p(x_n, t_n)$ is the stiff coefficient at the current step, and (4) $\tilde{p}_{n+1} = p(\tilde{x}_{n+1}, t_{n+1})$ is the stiff coefficient at the predictor's estimate $\tilde{x}_{n+1}$.

After obtaining $\tilde{p}_{n+1}$ and $\tilde{S}_{n+1}$ from this predictor step, the GRK2 method applies the following "corrector" update to compute x_{n+1}

$$x_{n+1} = e^{-\frac{p_n+\tilde{p}_{n+1}}{2}h} x_n + \tilde{S}_{n+1} I_0(p_n, \tilde{p}_{n+1}) + \frac{S_n - \tilde{S}_{n+1}}{h} I_1(p_n, \tilde{p}_{n+1}). \tag{10}$$

Here, the functions I_0 and I_1 are special integrals involving the error function erf, accounting for the variation of $p(t)$ over the step [1].

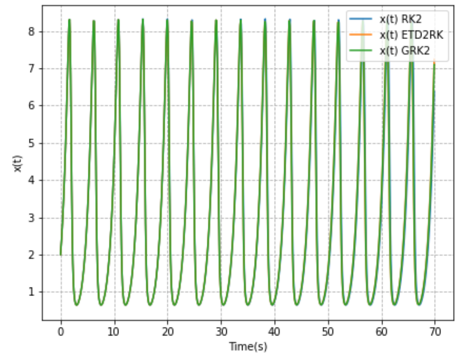
In principle, GRK2 is more stable than using a single exponential factor $e^{-p_n h}$ because it adjusts to how $p(t)$ evolves throughout the step. If $p(t)$ rises or falls sharply, treating it as constant can lead to inaccurate decay or growth, and potentially to divergence or excessive attenuation. GRK2, on the other hand, dynamically adapts to such variations, helping avoid the underestimation or runaway effects that often occur with simpler ETD methods.

In our implementation, we precompute the error-function expressions for I_0 and I_1 and evaluate them numerically using standard routines (e.g., `erf` in C), keeping the method explicit, avoiding large linear systems.

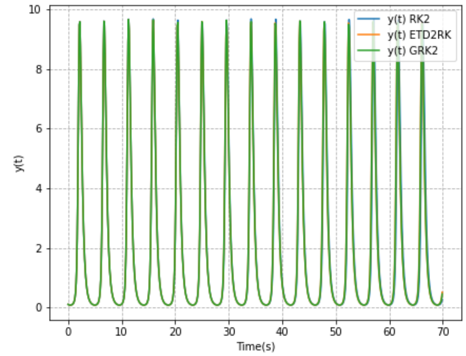
Observables and Performance Evaluation

We test RK2, ETD2RK, and GRK2 on both the Lotka–Volterra and van der Pol equations, focusing on stability—whether solutions remain finite or diverge—and on the accuracy of oscillation amplitude and frequency. For Lotka–Volterra, we track predator–prey cycles over time, while in van der Pol we examine the shape and periodicity of the relaxation oscillations. Although one could compute precise global errors by comparing with a small-step reference solution, we mainly analyze qualitative behaviour to highlight each solver's strengths and weaknesses when dealing with stiffness.

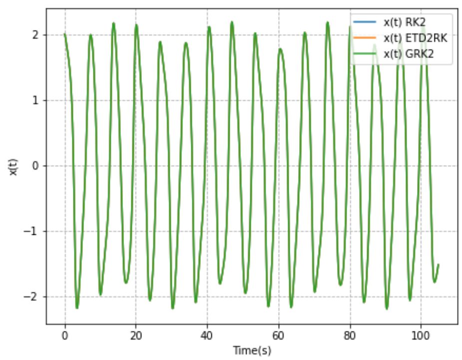
By running each solver at different stiffness levels and step sizes, we assess their performance under practical conditions. In the next section, we present numerical results, emphasizing differences in stability, accuracy, and computational behaviour.



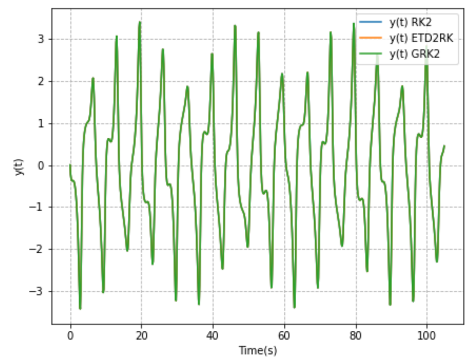
(a) Lotka–Volterra: $x(t)$



(b) Lotka–Volterra: $y(t)$



(c) van der Pol: $x(t)$



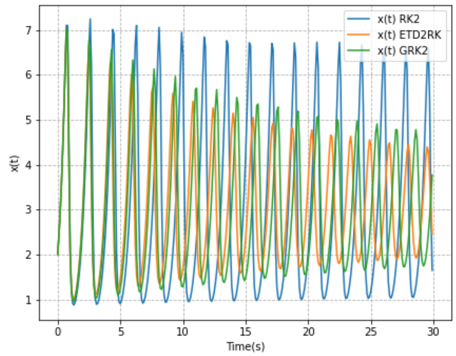
(d) van der Pol: $y(t)$

Figure 1: Less-stiff scenarios for Lotka–Volterra and van der Pol. Parameters for LV: $\alpha = 1, \beta = 0.5, \gamma = 3, \delta = 1$. Parameter for vdP: $\mu = 1$. Step size: $h = 0.01$. All three solvers produce virtually identical trajectories, so the solutions for RK2 and ETD2RK are overlaid by GRK2 (green).

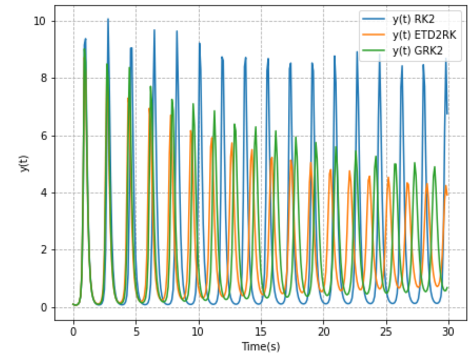
3 Results

We now present representative numerical results comparing RK2, ETD2RK, and GRK2 on the Lotka–Volterra LV (LV) and van der Pol (vdP) equations under three levels of stiffness. Unless otherwise specified, each figure includes all three methods. If a method diverges, it is omitted in subsequent figures to improve visual clarity of the other methods.

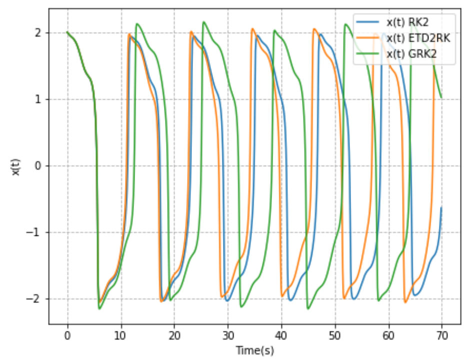
Less-Stiff Regime. Figure 1 illustrates the solutions for both LV and vdP in the "less-stiff" regime, where the parameters are kept small and the time step is $h = 0.01$. In both systems, all three solvers generate nearly identical trajectories, closely matching the expected oscillations. No solver diverges, and the amplitude and frequency remain consistent across the methods. These results confirm that when stiffness is low and the step size is relatively small, all three approaches preserve the oscillatory behaviour without significant damping or instability.



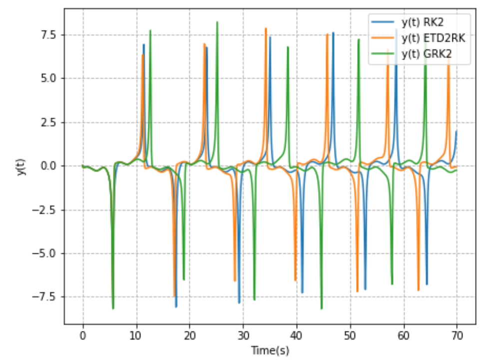
(a) Lotka–Volterra: $x(t)$



(b) Lotka–Volterra: $y(t)$



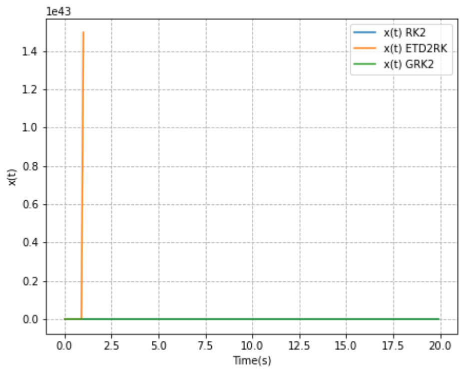
(c) van der Pol: $x(t)$



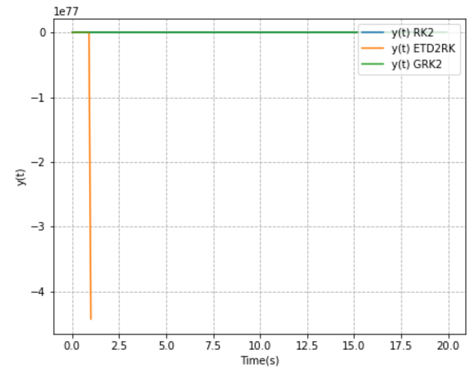
(d) van der Pol: $y(t)$

Figure 2: Normal-stiff scenarios for Lotka–Volterra and van der Pol. Parameters for LV: $\alpha = 2, \beta = 1, \gamma = 9, \delta = 3$. Parameter for vdP: $\mu = 5$. Step size: $h = 0.05$ for LV and $h = 0.1$ for vdP. ETD2RK and GRK2 exhibit amplitude and frequency discrepancies.

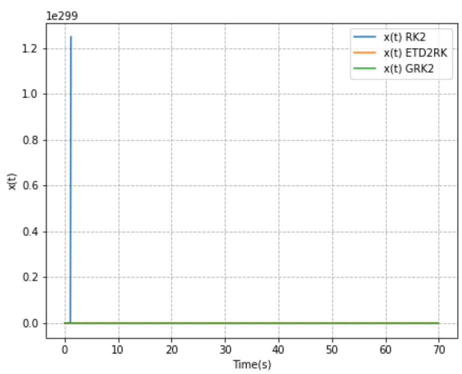
Normal-Stiff Regime. Figure 2 illustrates results under "normal" stiffness levels, using $h = 0.05$ for LV and $h = 0.1$ for vdP. In the LV case, RK2 remains closest to the reference solution, with minimal amplitude and phase errors. In contrast, ETD2RK and GRK2 show noticeable deviations: both experience amplitude decay over multiple cycles and advance more rapidly in phase (i.e., show a slightly higher frequency). Between these two ETD-based methods, GRK2 retains the amplitude and frequency somewhat more reliably. For the vdP oscillator, all three solvers produce comparable amplitudes, but GRK2 lags in phase by about half a cycle near $t = 80\,\text{s}$. Although these discrepancies become clear after several oscillations, none of the solvers diverges.



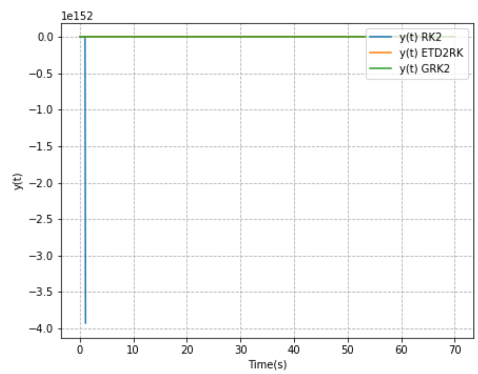
(a) Lotka–Volterra: $x(t)$



(b) Lotka–Volterra: $y(t)$

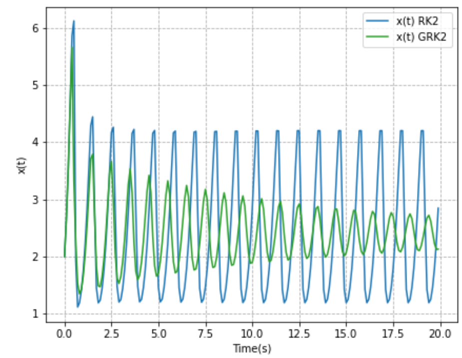


(c) van der Pol: $x(t)$

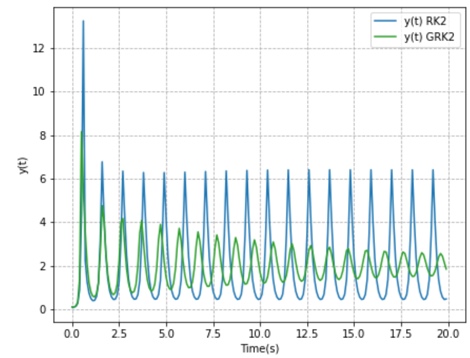


(d) van der Pol: $y(t)$

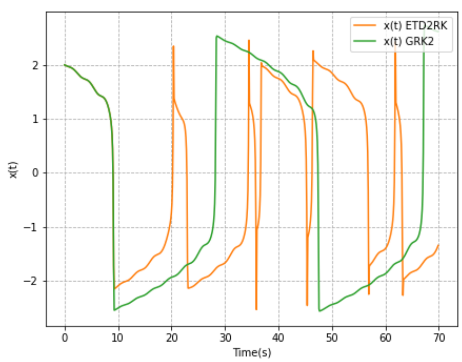
Figure 3: More-stiff scenarios for Lotka–Volterra and van der Pol. Parameters for LV: $\alpha = 3, \beta = 1.5, \gamma = -12, \delta = 5$. Parameter for vdP: $\mu = 10$. Step size: $h = 0.1$. ETD2RK diverges for LV, and RK2 diverges for vdP.



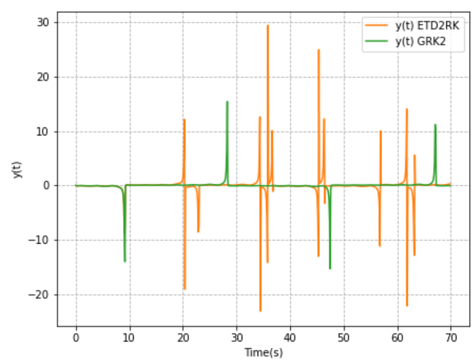
(a) Lotka–Volterra: $x(t)$



(b) Lotka–Volterra: $y(t)$



(c) van der Pol: $x(t)$



(d) van der Pol: $y(t)$

Figure 4: The same more-stiff scenarios for Lotka–Volterra and van der Pol, excluding the solvers that diverged (ETD2RK for LV and RK2 for vdP). Only GRK2 remains stable for both problems, though it exhibits larger amplitude and frequency discrepancies than in the normal-stiff scenarios.

More-Stiff Regime. To evaluate stability under stronger stiffness, we apply "more stiff" parameter sets with a step size of $h = 0.1$. For LV (Figure 3), ETD2RK diverges within a few steps, sending its trajectory to infinity and thus vanishing from the normal scale. Consequently, Figure 4 excludes ETD2RK and illustrates how RK2 and GRK2 both remain finite but differ in amplitude behaviour. RK2's amplitude decreases to about two-thirds of its initial value within half a cycle but then relatively constant afterward, whereas GRK2 continues to decay over time.

A similar pattern appears in the vdP system under the same conditions. In Figure 3, where all three methods are plotted, RK2 diverges to infinity after several oscillations, while ETD2RK and GRK2 remain stable. Therefore, Figure 4 omits RK2 and compares only ETD2RK and GRK2. Neither of these two schemes is particularly accurate in this regime, but GRK2 retains a shape closer to the less- and normal-stiff results, whereas ETD2RK fails to preserve the expected oscillatory structure.

Summary of Observations. Under low stiffness, all three methods yield similar solutions, accurately reproducing oscillatory behaviour in both LV and vdP. As stiffness increases, RK2 typically remains most accurate provided it does not diverge, whereas ETD2RK and GRK2 show visible amplitude/frequency errors. In the highest-stiffness tests at $h = 0.1$, ETD2RK may diverge for LV, and RK2 may diverge for vdP, yet GRK2 consistently stays finite for both problems–albeit with more pronounced amplitude and frequency discrepancies compared to the normal-stiff regime.

4 Discussion

Our numerical experiments show that RK2, ETD2RK, and GRK2 all handle mild stiffness effectively when the time step h is sufficiently small, as demonsstrated in the less-stiff scenarios (Figure 1). RK2 remains the most faithful to the exact trajectories provided it does not cross its stability boundary. In contrast, ETD2RK and GRK2 begin to exhibit amplitude decay and phase shifts–consistent with the broader notion that explicit Runge–Kutta methods can be very accurate if they stay within their stability limits, while ETD-type methods may introduce numerical damping or frequency drift due to their approximate treatment of stiff components.

Under stronger stiffness and a larger step size (Figure 3), at least one method diverges in each system, highlighting the shortcomings of each approach if h is not reduced. In the Lotka–Volterra example, ETD2RK diverges while RK2 and GRK2 remain finite, although GRK2 shows increased amplitude decay over time. In the van der Pol case at the same step-size, RK2 diverges instead, leaving ETD2RK and GRK2 with stable but not highly accurate solutions; GRK2 generally preserves the overall shape more effectively than ETD2RK, indicating an advantage when the stiff coefficient $p(t)$ varies rapidly.

These complementary blow-ups suggest that no single method is universally superior in all stiff regimes. RK2 can deliver excellent accuracy at sufficiently small step sizes but lacks the stabilization advantages that ETD-based methods offer for larger h. ETD2RK provides more stability than a straightforward explicit step, but it can fail if the assumption of a constant stiff coefficient is not valid. GRK2 refines this assumption by allowing $p(t)$ to vary linearly within each step, thereby reducing blow-ups when $p(t)$ changes significantly, though it often incurs additional numerical damping or phase errors compared to a stable RK2 run.

In practice, the solver choice depends on permissible step sizes, the rate at which $p(t)$ changes, and whether small amplitude or frequency inaccuracies are acceptable in exchange for greater stability. A more rigorous evaluation would involve applying these methods to larger PDE systems (e.g. relativistic hydrodynamics) and comparing their computational overhead: While GRK2 may require slightly more per-step computation due to error-function evaluations, but it can remain stable under conditions where simpler schemes diverge.

5 Conclusion

We have evaluated three numerical methods—RK2 (Heun's), ETD2RK, and GRK2—on two classic stiff problems (Lotka–Volterra and van der Pol) across different stiffness regimes. Our results indicate that RK2 can achieve impressive accuracy when stiffness is moderate and the step size is kept sufficiently small. Under more challenging conditions or with larger time steps, both RK2 and ETD2RK may diverge, whereas GRK2 proved more robust, never diverging in our tests. However, GRK2 attains this stability at the cost of introducing additional amplitude and phase errors compared to a stable RK2 result.

Overall, no single solver emerges as universally optimal: in cases where smaller steps are viable, RK2 generally outperforms the others in accuracy, but for larger steps in a strongly stiff system, GRK2 offers a more reliable alternative. Future work could involve evaluating these methods in more complex PDE settings (such as relativistic hydrodynamics) to determine whether GRK2's improved stability justifies its slightly higher computational cost and moderate phase or amplitude discrepancies.

References

[1] S. Jeon, "A new gaussian integrator for stiff differential equations in relativistic hydrodynamics," Unpublished notes.

[2] S. M. Cox and P. C. Matthews, "Exponential time differencing for stiff systems," *Journal of Computational Physics*, vol. 176, no. 2, pp. 430–455, 2002.

[3] LibreTexts, "The lotka–volterra predator–prey model," https://math.libretexts.org/Bookshelves/Applied_Mathematics/Mathematical_Biology_(Chasnov)/01%3A_Population_Dynamics/1.04%3A_The_Lotka-Volterra_Predator-Prey_Model, accessed: 2025-02-23.

[4] M. A. Elfouly and M. A. Sohaly, "Van der pol model in two-delay differential equation representation," *Scientific Reports*, vol. 12, no. 1, p. 2925, 2022.