APPLICATION OF ROSENBROCK METHODS TO TIGHTLY COUPLED MULTIPHYSICS SIMULATIONS IN NUCLEAR SCIENCE AND ENGINEERING

A Senior Scholars Thesis

by

JOSHUA EDMUND HANSEL

Submitted to the Office of Undergraduate Research
Texas A&M University
in partial fulfillment of the requirements for the designation as

UNDERGRADUATE RESEARCH SCHOLAR

April 2011

Major: Nuclear Engineering

APPLICATION OF ROSENBROCK METHODS TO TIGHTLY COUPLED MULTIPHYSICS SIMULATIONS IN NUCLEAR SCIENCE AND ENGINEERING

A Senior Scholars Thesis

by

JOSHUA EDMUND HANSEL

Submitted to the Office of Undergraduate Research
Texas A&M University
in partial fulfillment of the requirements for the designation as

UNDERGRADUATE RESEARCH SCHOLAR

Approved by:

Research Advisor,	Jean C. Ragusa
Director for Undergraduate Research,	Sumana Datta

April 2011

Major: Nuclear Engineering

ABSTRACT

Application of Rosenbrock Methods to Tightly Coupled Multiphysics Simulations in Nuclear Science and Engineering. (April 2011)

Joshua Edmund Hansel
Department of Nuclear Engineering
Texas A&M University

Research Advisor: Dr. Jean C. Ragusa
Department of Nuclear Engineering

Recently, researchers have investigated the implementation of accurate high order time discretization techniques in large-scale nonlinear multiphysics simulations using Implicit Runge-Kutta (IRK) methods. For a given time step, IRK methods require the iterative solution of a nonlinear system of equations using Newton's method. Rosenbrock methods, a variant of IRK methods, avoid this issue by linearizing this system of equations, so only one Newton iteration is required at each stage. Because Rosenbrock methods may achieve this without loss of accuracy order or stability, Rosenbrock methods have the potential to generate accurate solutions more efficiently. This research investigates these claims by applying Rosenbrock methods to two representative multiphysics problems found in nuclear science and engineering: (1) the Point Reactor Kinetics Equations (PRKE) with temperature-induced reactivity feedback, and (2) non-equilibrium radiation diffusion. To assess the merits of Rosenbrock methods, a measure of accuracy per computational cost was compared between Rosenbrock methods and IRK methods, and Rosenbrock methods were found to achieve a smaller computational cost for a given level of accuracy than IRK methods of the same convergence order.

ACKNOWLEDGMENTS

I would like to acknowledge the efforts of my research advisor, Dr. Jean C. Ragusa, who taught me everything I know about numerical methods and invested a large amount of his time to provide me with valuable guidance throughout the course of the research.

I would also like to acknowledge Vijay S. Mahadevan for his time and efforts answering the frequent questions I had regarding the KARMA software and numerical methods.

Lastly, I would like to acknowledge my family and friends for the love and support that makes my life so wonderful.

TABLE OF CONTENTS

Page

ABSTRACT iii

ACKNOWLEDGMENTS iv

TABLE OF CONTENTS v

LIST OF TABLES vii

LIST OF FIGURES viii

CHAPTER

I INTRODUCTION 1

Previous work 2
Background 3
Point reactor kinetics equations 3
Radiative transfer 8

II METHODS 11

Solution techniques 11
Linear solution techniques 11
Nonlinear solution techniques 12
Discretization techniques 13
Temporal discretization techniques 13
Spatial discretization of radiative transfer equations . . . 18
Verification techniques 20
Truncation error analysis 20
Adaptive time step control 21

III RESULTS 23

Linear results 24
Linear PRKE benchmarks convergence results 24
Nonlinear results 35

CHAPTER Page

Nonlinear PRKE convergence results 35
Nonlinear PRKE efficacy results 40
Nonlinear radiative transfer efficacy results 46

IV CONCLUSION 48

REFERENCES 49

CONTACT INFORMATION 51

LIST OF TABLES

TABLE		Page
1	Temporal Integration Methods Compared in This Thesis	24
2	Linear Benchmark Definitions	25
3	Reactivity Function Definitions	26
4	Reactor Data Groups	27
5	Nonlinear PRKE Cases Definitions	35

LIST OF FIGURES

FIGURE		Page
1	Convergence for Linear PRKE Benchmark 1	28
2	Convergence for Linear PRKE Benchmark 2	29
3	Convergence for Linear PRKE Benchmark 3	30
4	Convergence for Linear PRKE Benchmark 4	31
5	Convergence for Linear PRKE Benchmark 5	32
6	Convergence for Linear PRKE Benchmark 6	33
7	Convergence for Linear PRKE Benchmark 7	34
8	Convergence for Nonlinear PRKE Case 1	36
9	Convergence for Nonlinear PRKE Case 2	37
10	Convergence for Nonlinear PRKE Case 3	38
11	Convergence for Nonlinear PRKE Case 4	39
12	Convergence for Nonlinear PRKE Case 5	40
13	Efficacy Plot for Nonlinear PRKE Case 1	42
14	Efficacy Plot for Nonlinear PRKE Case 2	43
15	Efficacy Plot for Nonlinear PRKE Case 3	44
16	Efficacy Plot for Nonlinear PRKE Case 4	45
17	Efficacy Plot for Nonlinear PRKE Case 5	46

CHAPTER I

INTRODUCTION

Traditionally, multiphyics simulations in nuclear engineering have been divided into sub-problems, each to be solved by a mono-physics code, with ad-hoc data exchange through message-passing paradigms. This technique, referred to as Operator Splitting (OS), destroys the strong coupling between the physics, and thus may lead to a loss of accuracy and stability in time-dependent problems. Recently, researchers have investigated the implementation of accurate high order time discretization techniques in large-scale nonlinear multiphysics simulations using Implicit Runge-Kutta (IRK) methods. For many multiphysics problems, time implicitness is necessary for stability due to the great disparity of time scales. For a given time step, IRK methods require the iterative solution of a nonlinear system of equations using Newton's method. Rosenbrock methods, a variant of IRK methods, avoid this issue by linearizing this system of equations, so only one Newton iteration is required at each stage stage. Because Rosenbrock methods may achieve this without loss of accuracy order or stability, Rosenbrock methods have the potential to generate accurate solutions more efficiently. This research investigates these claims by applying Rosenbrock methods to two representative multiphysics problems found in nuclear science and engineering: (1) the Point Reactor Kinetics Equations (PRKE) with temperature-induced reactivity feedback, and (2) non-equilibrium radiation diffusion. The PRKE are useful for simulating nuclear reactor transients, which may be initiated by events such as control rod movements, pump flow changes, pipe breaks, etc. Radiative diffu-

---

This thesis follows the style of Journal of Computational Physics.

sion simulations, which couple radiative energy and material temperature, are used in several applications, including high temperature industrial ovens, glass cooling, and stellar atmospheres, and describe the interactions of energetic photons with matter. To assess the merits of Rosenbrock methods, a measure of accuracy per computational cost will be compared between Rosenbrock methods and IRK methods.

Previous work

Generally, the choice of the optimal time-integration method is very problem-specific. For simple cases, the behavior of time-integration methods can be predicted, but more complex systems require more analysis to determine the applicability of a time-integration method. Much work has been done on the development of Rosenbrock methods [1, 2, 3, 4], and they have been applied for several different problems, such as photochemical dispersion [5], electric circuits [6], and flexible multibody systems [7]. Rosenbrock methods have been applied to the PRKE [8, 9, 10], but this was to the standard linear PRKE. Recall that Rosenbrock methods are variants of IRK methods that linearize the system, so application of Rosenbrock methods to an already linear system fails to highlight the merits of the methods. This research applies the Rosenbrock methods to a nonlinear PRKE problem, which couples reactor fuel and moderator temperatures to reactor power. Some literature has included the application of Rosenbrock methods to the non-equilibrium radiation diffusion equations [11, 12, 13], and these studies suggest the possible merits of using Rosenbrock methods.

Background

Point reactor kinetics equations

The Point Reactor Kinetics Equations (PRKE) describe the relationship between reactor power P and delayed neutron precursor group concentrations C_j:

$$\frac{dP}{dt} = \frac{\rho - \beta}{\Lambda} P + \frac{1}{\Lambda} \sum_{j=1}^{N_p} \lambda_j \zeta_j, \tag{1.1}$$

and

$$\frac{d\zeta_j}{dt} = \beta_j P - \lambda_j \zeta_j, \tag{1.2}$$

with:

ρ = reactivity, equal to $(k-1)/k$, where k is the neutron multiplication factor.

β = total delayed neutron fraction.

Λ = is the mean generation time.

N_p = number of neutron precursor groups.

λ_j = is the decay constant for neutron precursor group j.

ζ_j = neutron precursor group concentration C_j multiplied by Λ.

β_j = delayed neutron fraction for neutron precursor group j.

These equations form a linear system for the PRKE when feedback is not included. This research mainly concerns nonlinear systems; however, this linear system is important in preliminary code verification, which includes verification of theoretical convergence orders and comparison to benchmark results in literature.

Because neutron cross sections depend on the relative speed between neutrons and target nuclei, cross sections must be averaged over the range of speeds resulting from the thermal motion of the target nuclei, which has the effect of smearing

resonances in the cross sections. As temperature is increased, the increased thermal motion increases the effect, a phenomenon known as Doppler broadening. This results in a decrease in resonance escape probability from capture; thus, less neutrons are able to thermalize and induce fission, decreasing reactivity. In addition, increasing the temperature of the coolant decreases its density, allowing neutrons to pass through coolant with decreased moderation, which causes increased resonance capture [14].

Two feedback models were considered in this research:

1. T_{fuel}, T_{cool}: Fuel Temperature and Coolant Temperature Feedback.
2. T_r: Reactor Temperature Feedback.

The first model explicitly defines fuel temperature T_{fuel} and coolant temperature T_{cool} to separately account for their effects on reactivity. The second model assumes a simplified relationship between reactivity and an average reactor temperature T_r.

Fuel temperature and coolant temperature feedback

This model was taken directly from [15]. The reactivity depends on fuel temperature T_{fuel} and coolant temperature T_{cool} as:

$$\rho = \rho_{ext} + \alpha_{Dop}\left(T_{fuel}^{d} - T_{fuel,0}^{d}\right) + \alpha_{cool}\left(T_{cool} - T_{cool,0}\right), \tag{1.3}$$

with:

ρ_{ext} = external reactivity contribution. This arises from events such as control rod movements, flow changes, pipe breaks, etc.

α_{Dop} = Doppler coefficient.

α_{cool} = coolant temperature coefficient of reactivity.

d = Doppler exponent.

$T_{fuel,0}$ = initial fuel temperature.

$T_{cool,0}$ = initial coolant temperature.

Energy balances in the fuel and coolant give, respectively:

$$\frac{dT_{fuel}}{dt} = \frac{\Omega_{pow}}{\rho_{fuel} c_p^{fuel}} \left[(1-\kappa) P + \sum_{k=1}^{N_{hg}} \lambda_k^{FP} \omega_k \right] - \frac{1}{\rho_{fuel} c_p^{fuel} R_{th}} \left(T_{fuel} - T_{cool} \right) \tag{1.4}$$

and

$$\frac{dT_{cool}}{dt} = -2 \frac{u}{H} \left(T_{cool} - T_{cool,in} \right) + \frac{A_{fuel}}{A_{flow}} \frac{1}{\rho_{cool} c_p^{cool} R_{th}} \left(T_{fuel} - T_{cool} \right), \tag{1.5}$$

with:

Ω_{pow} = conversion factor from normalized power to power density.

ρ_{fuel} = fuel density.

ρ_{cool} = coolant density.

c_p^{fuel} = fuel specific heat capacity.

c_p^{cool} = coolant specific heat capacity.

κ = decay heat power fraction of total power P.

N_{hg} = number of decay heat groups.

λ_k^{FP} = decay constant for decay heat group k.

ω_k = decay heat group k power.

R_{th} = thermal resistance between fuel pin and coolant.

u = inlet coolant speed.

H = reactor height.

$T_{cool,in}$ = inlet coolant temperature.

A_{fuel} = cross-sectional area of fuel pin, πR_{fuel}^2.

A_{flow} = average cross-sectional flow area around a single fuel pin.

The heat produced in a reactor is the sum of the heat generated from fission and heat generated by decay products, which occurs much more slowly. Decay heat is modeled with several groups of decay products based on decay rate. These heat components are ω_k, where k is the decay heat group. The rates of change of decay heat are

$$\frac{d\omega_k}{dt} = \kappa_k P - \lambda_k^{FP}\omega_k, \tag{1.6}$$

where κ_k is the decay heat group k fraction of total power P. The total decay heat fraction κ is the sum of all κ_k:

$$\kappa = \sum_{k=1}^{N_{hg}} \kappa_k. \tag{1.7}$$

Both Equations 1.4 and 1.5 depend on the thermal resistance R_{th} between the fuel pin and coolant. Here, the thermal resistance model contains components including convective heat transfer between the coolant and cladding, conduction through the cladding, conduction through the fuel-cladding gap, and conduction through the fuel:

$$R_{th} = A_{fuel}\left(\frac{1}{2\pi R_{gap}h_{gap}} + \frac{1}{2\pi k_{clad}}\ln\frac{R_{clad}}{R_{gap}} + \frac{1}{2\pi R_{clad}h_{cool}} + \frac{w}{4\pi k_{fuel}T_{fuel}}\right), \quad (1.8)$$

with:

R_{fuel} = radius of fuel pin.

R_{gap} = radius of fuel plus gas gap (also, inner radius of cladding).

R_{clad} = outer radius of cladding.

h_{gap} = convective heat transfer coefficient of gas gap.

h_{cool} = convective heat transfer coefficient of coolant.

k_{fuel} = conductivity of fuel pin.

k_{clad} = conductivity of cladding.

w = weighting factor used to compute the effective fuel temperature T_{fuel}.

The weighting factor relates the fuel pin centerline temperature T_{fuel}^{CL} and the surface temperature T_{fuel}^{s}:

$$T_{fuel} = wT_{fuel}^{CL} + (1-w)\,T_{fuel}^{s}. \quad (1.9)$$

Reactor temperature feedback

The reactivity ρ depends on an average reactor temperature T_r:

$$\rho = \rho_{ext} + \alpha\,(T - T_0)\,, \quad (1.10)$$

where α is the temperature reactivity coefficient. The rate of temperature change is proportional to the power:

$$\frac{dT}{dt} = KP, \tag{1.11}$$

where K is a constant.

This feedback model is a simplification of the fuel and coolant temperature feedback model, but it still makes the system nonlinear and thus is useful for evaluating the merit of Rosenbrock methods.

Radiative transfer

As photons move through a medium, they undergo absorption and black body reemission processes that depend on the radiative energy E (photon energy) and the material temperature T. Energy balances of E and T account for absorption, emission, and leakage out of the domain. Using the standard gray approximation, i.e., absorptivity α and emissivity ϵ are independent of the photon wavelength, and using the diffusion approximation, the absorptivity and emissivity become constants with $\alpha = \epsilon$ [16]. The diffusion approximation assumes that the current of a quantity is proportional to the gradient and in the opposite direction of the gradient; that is,

$$\vec{J_E} = -D\vec{\nabla}E. \tag{1.12}$$

With these approximations, the equations governing radiative transfer become

$$\frac{\partial E}{\partial t} - c\vec{\nabla} \bullet \left(D_r \vec{\nabla} E\right) = c\sigma_a \left(aT^4 - E\right), \tag{1.13}$$

and

$$\rho C_v \frac{\partial T}{\partial t} - \vec{\nabla} \bullet \left(D_T \vec{\nabla} T\right) = -c\sigma_a \left(aT^4 - E\right), \tag{1.14}$$

where

c = speed of light.

D_r = radiative energy diffusion coefficient.

σ_a = photon absorption cross section.

a = Stefan-Boltzmann constant.

ρ = material density.

C_v = material specific heat capacity.

D_T = material conductivity.

The photon absorption cross section is a function of temperature T:

$$\sigma_a(T) = \frac{z^3}{T^3}, \tag{1.15}$$

where z is the atomic number. The radiative energy diffusion coefficient is

$$D_r(T) = \frac{1}{3\sigma_a(T)}. \tag{1.16}$$

However, diffusion theory can fail in regions of strong gradients, resulting in energy moving faster than the speed of light. To prevent this, a technique known as flux limiting is employed [17]:

$$D_r(E,T) = \frac{1}{3\sigma_a(T) + \frac{|\vec{\nabla}E|}{E}}. \tag{1.17}$$

The conduction diffusion coefficient D_T is taken from [17]:

$$D_T(T) = kT^{5/2}, \tag{1.18}$$

where k is a constant.

To make these equations consistent with those in literature, the simplification $c = a = \rho = C_v = 1$ was made [17, 18, 19, 13, 20]. While this simplification does not give results in physical units, it preserves the behavior of the simulation and remains

a valuable tool to evaluate numerical methods.

CHAPTER II

METHODS

Solution techniques

The PRKE with feedback and the spatially discretized radiation transfer equations form nonlinear systems. Newton's method is a common solution technique for nonlinear systems that requires a linear solve at each iteration. This chapter will briefly review linear and nonlinear solution techniques. The spatial discretization of the radiative transfer equations will be reviewed as well as temporal discretization techniques such as those in the Runge-Kutta family and Rosenbrock family. This chapter will review truncation error analysis, which is used to verify that temporal integration methods converge correctly, and conclude with an introduction to adaptive time step integration, which is necessary for stiff systems that cannot be efficiently solved with constant time step sizes.

Linear solution techniques

Numerous techniques have been developed to solve linear systems and are generally well-known and thus will not be detailed in this thesis. Examples of techniques are LU decomposition for a direct solve, or iterative solvers such as the Jacobi method, the Gauss-Seidel method, the Successive Over-Relaxation method, the conjugate gradient method (for symmetric matrices), or the generalized minimum residual method (for unsymmetric matrices)[21].

Nonlinear solution techniques

Newton's method

Nonlinear equations may be solved by a variety of techniques, the most noteworthy being Newton's Method, which is the method employed by the software platform KARMA, which will be used to conduct this research. A system of nonlinear equations may be written in the form

$$\vec{g}(\vec{y}) = \vec{0}, \tag{2.1}$$

where $\vec{g}$ is the vector residual function. For each iteration ℓ, Newton's Method takes a step $\vec{\delta y}^{\,\ell}$ from the current iteration solution $\vec{y}^{\ell}$ to the next iteration solution $\vec{y}^{\ell+1}$:

$$\vec{y}^{\ell+1} = \vec{y}^{\ell} + \vec{\delta y}^{\,\ell}. \tag{2.2}$$

The objective of Newton's Method is to obtain a solution $\vec{y}^{\ell+1}$ such that Equation 2.1 is satisfied:

$$\vec{g}(\vec{y}^{\ell+1}) \approx \vec{0}. \tag{2.3}$$

Combining Equations 2.2 and 2.3 and taking a first-order Taylor series expansion about $\vec{y}^{\ell}$ gives

$$\vec{g}(\vec{y}^{\ell+1}) = \vec{g}(\vec{y}^{\ell} + \vec{\delta y}^{\,\ell}) = \vec{g}(\vec{y}^{\ell}) + \frac{\partial \vec{g}}{\partial \vec{y}}\left(\vec{y}^{\ell}\right)\vec{\delta y}^{\,\ell} = \vec{0}. \tag{2.4}$$

Thus the solution of the linear system

$$J\vec{\delta y}^{\,\ell} = -\vec{g}(\vec{y}^{\ell}) \tag{2.5}$$

is required at each iteration, where J is the Jacobian matrix $\frac{\partial \vec{g}}{\partial \vec{y}}$ evaluated at $\vec{y}^{\ell}$. The iterative process continues until the error between iterations ℓ and $\ell + 1$,

$$\epsilon^{\ell+1} = \left\| \vec{y}^{\ell+1} - \vec{y}^{\ell} \right\|, \tag{2.6}$$

satisfies the desired error tolerance τ. The $||$ brackets denote a norm of the argument. Several norms exist, including the L^∞ norm:

$$\epsilon_{L^\infty} = \max_i \left(\left| y_i^{\ell+1} - y_i^{\ell} \right| \right), \tag{2.7}$$

or the L^2 norm:

$$\epsilon_{L^2} = \sqrt{\sum_{i=1}^{N} \left(y_i^{\ell+1} - y_i^{\ell} \right)^2}. \tag{2.8}$$

Discretization techniques

Temporal discretization techniques

A problem may be expressed as a system of ordinary differential equations (ODEs), or a system of partial differential equations (PDEs), which after spatial discretization, becomes a system of ODEs:

$$\frac{d\vec{y}}{dt} = \vec{f}\left(t, \vec{y}\right), \tag{2.9}$$

which is solved by sequentially solving a time-discretized form of Equation 2.9. There are many choices of time-discretization techniques, such as theta discretization methods, Runge-Kutta methods, and Rosenbrock methods, which will all be described.

Theta discretization methods

For theta (θ) discretization methods, the derivative $\frac{d\vec{y}}{dt}$ is approximated as

$$\frac{\vec{y}_{n+1} - \vec{y}_n}{h} = \theta \left. \frac{d\vec{y}}{dt} \right|_{n+1} + (1-\theta) \left. \frac{d\vec{y}}{dt} \right|_{n}, \tag{2.10}$$

where h is the time step size taken from time t_n to time t_{n+1}. For $\theta = 0$, known as the Explicit Euler Method or the Forward Euler Method, solution for $\vec{y}_{n+1}$ becomes

explicit:

$$\vec{y}_{n+1} = \vec{y}_n + h \left.\frac{d\vec{y}}{dt}\right|_n . \tag{2.11}$$

For $\theta > 0$, the solution is implicit. For example, when $\theta = 0$, known as the Implicit Euler Method or the Backward Euler Method, the solution is:

$$\vec{y}_{n+1} - h \left.\frac{d\vec{y}}{dt}\right|_{n+1} = \vec{y}_n. \tag{2.12}$$

Another example is the Crank-Nicolson Method $\left(\theta = \frac{1}{2}\right)$:

$$\vec{y}_{n+1} - \frac{1}{2}h \left.\frac{d\vec{y}}{dt}\right|_{n+1} = \vec{y}_n + \frac{1}{2}h \left.\frac{d\vec{y}}{dt}\right|_n . \tag{2.13}$$

Runge-Kutta methods

For Runge-Kutta (RK) methods, the steady-state residual function $\vec{f}\,(t, \vec{y})$ is formed by a linear combination of function evaluations between times t^n and t^{n+1}:

$$\vec{y}^{n+1} = \vec{y}^n + h \sum_{i=1}^{s} b_i \vec{k}_i, \tag{2.14}$$

where $\vec{y}^n$ is the solution at time t^n, $\vec{y}^{n+1}$ is the solution at time t^{n+1}, h is the time step, s is the number of stages, the constants b_i are specific to the particular RK method used, and the vectors $\vec{k}_i$ are defined as

$$\vec{k}_i = \vec{f}\left(t^n + c_i h, \qquad \vec{y}^n + h \sum_{j=1}^{s} a_{i,j} \vec{k}_j\right). \tag{2.15}$$

The coefficients $a_{i,j}$, b_i, and c_i are specific to the particular RK method and are conventionally represented by a Butcher tableau:

$$\begin{array}{c|cccc} c_1 & a_{1,1} & a_{1,2} & \cdots & a_{1,s} \\ c_2 & a_{2,1} & a_{2,2} & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ c_s & a_{s,1} & \cdots & \cdots & a_{s,s} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

These coefficients are often referred to as "determining coefficients" because their values determine the accuracy order of the particular RK method. For a given accuracy order, there are degrees of freedom that allow multiple methods to be created of the same accuracy order but with varying stability properties such as absolute stability, L-stability, and magnitude of the error constant in the leader error term.

Explicit Runge-Kutta methods have a strictly-lower triangular coefficient matrix, i.e., only elements below the main diagonal are nonzero:

$$\begin{array}{c|cccc} c_1 & 0 & \cdots & \cdots & 0 \\ c_2 & a_{2,1} & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ c_s & a_{s,1} & \cdots & a_{s,s-1} & 0 \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

Explicit methods have the advantage that no linear solves are needed; each stage i may be computed directly because $\vec{k}_i$ only depends on $\vec{k}_j$ for $j < i$. However, explicit methods must be used with caution because they may be unstable for stiff systems, and thus they are not used in this research.

For *diagonally implicit* Runge-Kutta (DIRK) methods, the coefficients $a_{i,j}$ form

a lower triangular matrix, i.e., $a_{i,j} = 0$ for $j > i$:

$$\begin{array}{c|cccc} c_1 & a_{1,1} & 0 & \cdots & 0 \\ c_2 & a_{2,1} & a_{2,2} & \ddots & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ c_s & a_{s,1} & \cdots & \cdots & a_{s,s} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

Thus, the vector $\vec{k}_i$ may be reduced to

$$\vec{k}_i = \vec{f}\left(t^n + c_i h, \qquad \vec{y}^n + h\sum_{j=1}^{i} a_{i,j}\vec{k}_j\right). \tag{2.16}$$

Thus, each stage s requires a single nonlinear solve for $\vec{k}_i$ because the vectors $\vec{k}_j$ for $j < i$ have already been found in previous stages. *Fully implicit* Runge-Kutta methods, on the other hand, require the simultaneous solution of all $\vec{k}_j$ from $j = 1, \ldots, s$.

Singly diagonally implicit Runge-Kutta (SDIRK) methods are a subclass of DIRK methods that have the condition that the elements along the diagonal, i.e., $a_{i,i}$ for $i = 1, \ldots, s$ are equal:

$$\begin{array}{c|cccc} c_1 & \gamma & 0 & \cdots & 0 \\ c_2 & a_{2,1} & \gamma & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ c_s & a_{s,1} & \cdots & a_{s,s-1} & \gamma \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

For each stage i, the matrix $I - ha_{i,i}\frac{\partial \vec{f}}{\partial \vec{y}}$ must undergo LU decomposition, so the advantage of SDIRK methods is that they may reuse the LU decomposition from the first stage, since the diagonal elements $a_{i,i}$ are all equal.

Rosenbrock methods

Rosenbrock methods are a subclass of DIRK methods that linearize Equation 2.15, so Newton's method will converge after one iteration. The solution is still the same linear combination of s intermediate solutions:

$$\vec{y}^{n+1} = \vec{y}^{n} + h\sum_{i=1}^{s} b_i \vec{k}_i, \tag{2.17}$$

but the linearization makes $\vec{k}_i$:

$$\vec{k}_i = \vec{f}\left(t^n + \alpha_i h, \quad \vec{y}^{n} + \sum_{j=1}^{i-1} \alpha_{i,j}\vec{k}_j\right) + \gamma_i h \frac{\partial \vec{f}}{\partial t}\left(t^n, \vec{y}^{n}\right) + \frac{\partial \vec{f}}{\partial \vec{y}}\left(t^n, \vec{y}^{n}\right)\sum_{j=1}^{i} \gamma_{i,j}\vec{k}_j, \tag{2.18}$$

where the coefficients α_i is defined as

$$\alpha_i = \sum_{j=1}^{i-1} \alpha_{i,j}, \tag{2.19}$$

and the coefficients γ_i are defined as

$$\gamma_i = \sum_{j=1}^{i} \gamma_{i,j}. \tag{2.20}$$

Notice that there are now two sets of coefficients, $\alpha_{i,j}$ and $\gamma_{i,j}$. The coefficients $\alpha_{i,j}$ play the role of the coefficients $a_{i,j}$ from the nonlinearized Runge-Kutta methods. The coefficients $\gamma_{i,j}$ are the linear combination coefficients of the Jacobian $\frac{\partial \vec{f}}{\partial \vec{y}}$, i.e., the replacement

$$\alpha_{i,i}\vec{k}_i = \sum_{j=1}^{i} \gamma_{i,j}\vec{k}_j \tag{2.21}$$

was made, which increases the number of degrees of freedom [22]. Again, these determining coefficients are chosen to produce a method with a certain number of stages and accuracy order.

Spatial discretization of radiative transfer equations

To solve the radiative transfer equations, they must be discretized in space. This section shows how the finite volume method was used to create the nonlinear system of ODEs used to simulate the radiative transfer problem.

Finite volume, 1-D

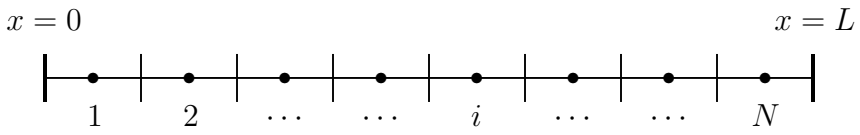
For a 1-D domain with the simplification $c = a = \rho = C_v = 1$, the radiative transfer equations become

$$\frac{\partial E}{\partial t} = \frac{\partial}{\partial x}\left(D_r \frac{\partial E}{\partial x}\right) + \sigma_a \left(T^4 - E\right), \tag{2.22}$$

and

$$\frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(D_T \frac{\partial T}{\partial x}\right) - \sigma_a \left(T^4 - E\right). \tag{2.23}$$

To discretize a system of equations using the Finite Volume method in 1-D, the problem domain spanning from $x = 0$ to $x = L$ is divided into N equally-sized cells of size h. The cells are numbered as $i = 1, 2, \ldots, N$, and energy and temperature are centered on each cell. The cell boundaries are numbered as $\frac{1}{2}, \frac{3}{2}, \ldots, N + \frac{1}{2}$.



Equations 2.22 and 2.23 are integrated over each cell, forming a nonlinear system of $2N$ unknowns: $E_1, E_2, \ldots, E_N, T_1, T_2, \ldots, T_N$. After integrating over each cell i and dividing by h, the following equations are obtained:

$$\frac{\partial E_i}{\partial t} = \frac{1}{h} D_{r,i+1/2} \left.\frac{\partial E}{\partial x}\right|_{i+1/2} - \frac{1}{h} D_{r,i-1/2} \left.\frac{\partial E}{\partial x}\right|_{i-1/2} + \frac{\sigma_{a,i}}{h}\left(T_i^4 - E_i\right), \tag{2.24}$$

and

$$\frac{\partial T_i}{\partial t} = \frac{1}{h} D_{T,i+1/2} \left.\frac{\partial T}{\partial x}\right|_{i+1/2} - \frac{1}{h} D_{T,i-1/2} \left.\frac{\partial T}{\partial x}\right|_{i-1/2} - \frac{\sigma_{a,i}}{h}\left(T_i^4 - E_i\right). \tag{2.25}$$

A finite difference approximation is made to the gradients $\frac{\partial E}{\partial x}$ and $\frac{\partial T}{\partial x}$ on the cell edges, using values from the adjacent cell volumes. For example,

$$\left.\frac{\partial E}{\partial x}\right|_{i+1/2} = \frac{E_{i+1} - E_i}{h}. \tag{2.26}$$

The radiative energy diffusion coefficient is evaluated at the boundary using an average energy and temperature between adjacent cells. For example,

$$D_{r,i+1/2} = D_r\left(E_{i+1/2}, T_{i+1/2}\right) = \frac{1}{3\sigma_{a,i+1/2} + \frac{|E_{i+1} - E_i|}{hE_{i+1/2}}}, \tag{2.27}$$

where

$$E_{i+1/2} = \frac{E_{i+1} + E_i}{2}, \tag{2.28}$$

$$T_{i+1/2} = \frac{T_{i+1} + T_i}{2}, \tag{2.29}$$

and

$$\sigma_{a,i+1/2} = \sigma_a\left(z_{i+1/2}, T_{i+1/2}\right) = \frac{z_{i+1/2}^3}{T_{i+1/2}^3}, \tag{2.30}$$

where

$$z_{i+1/2} = \frac{z_{i+1} + z_i}{2}. \tag{2.31}$$

The temperature diffusion coefficient D_T is evaluated at the cell edge by using an average temperature between adjacent cells:

$$D_T|_{i+1/2} = D_T\left(T_{i+1/2}\right) = k\left(\frac{T_{i+1} + T_i}{2}\right)^{5/2}. \tag{2.32}$$

At $x = 0$, there is a source boundary condition:

$$\frac{E}{4} - \frac{1}{6\sigma_a}\left.\frac{\partial E}{\partial x}\right|_{x=0} = 1, \tag{2.33}$$

and at $x = L$, there is a vacuum boundary condition:

$$\frac{E}{4} + \frac{1}{6\sigma_a} \left. \frac{\partial E}{\partial x} \right|_{x=L} = 0. \tag{2.34}$$

Reflective boundary conditions apply for temperature:

$$\left. \frac{\partial T}{\partial x} \right|_{x=0} = 0, \quad \left. \frac{\partial T}{\partial x} \right|_{x=L} = 0. \tag{2.35}$$

Verification techniques

Truncation error analysis

Truncation error is associated with any temporal integration technique and arises from the truncation of terms of the Taylor series expansion, which has the form

$$T_N \left(t_n + h \right) = \sum_{j=0}^{N} \frac{\vec{y}^{(j)} \left(t_n \right)}{j!} h^j. \tag{2.36}$$

The *local* truncation error is the truncation error accumulated from a single time step $t_n \rightarrow t_{n+1}$, and it can be computed by taking the difference between the solution $\vec{y}_{n+1}$ and the Taylor expansion taken about the previous time value:

$$\epsilon = \left| \vec{y}_{n+1} - T_\infty \left(t_n + h \right) \right|. \tag{2.37}$$

For the theta discretization technique, this difference looks like

$$\epsilon = \left(\frac{h^2}{2} - h^2 \theta \right) \left. \frac{\partial^2 \vec{y}}{\partial t^2} \right|_n + O\left(h^3 \right), \tag{2.38}$$

where the $O\left(h^3\right)$ notation denotes all terms proportional to h^3 and higher order (h^4,h^5, etc.). For $\theta = \frac{1}{2}$, the h^2 term is zero, and the local truncation error is proportional to h^3. For other θ values, the error is proportional to h^2.

The *global* truncation error is the truncation error accumulated from multiple

time steps, and its order is always one less than the local error order. For example, the Crank-Nicolson method has a local truncation error order of 3 and a global truncation error order of 2.

For a method with a global truncation error order of p, plotting $\log \epsilon$ as a function of $\log h$ yields a line with a slope of p:

$$\log \epsilon = p \log h + C. \tag{2.39}$$

Thus, one way to verify that a temporal integration method has been implemented correctly is to create these plots and compare slopes to theoretical global error orders.

Adaptive time step control

With the goal of achieving a solution of a desired accuracy, the use of a constant time step size is generally inefficient. Temporal integration methods impose restraints on the time step size for both stability and accuracy, but as a problem evolves the stability time step constraint may become less strict, enabling larger time steps to be taken without significant loss of accuracy. Stiff systems, i.e., systems with a large range of eigenvalues, are particularly notorious for the strict time step size constraint they impose. The solution to such systems is composed of a number of solution modes with a large range of lifetimes; some last throughout the time domain of interest, while others may quickly approach zero. After the quickly dying components become negligible, the time step size constraint becomes less strict, so it is economical to increase the time step size at that point. Time step adaptation is a technique that uses local truncation error estimates to determine an appropriate time step size to take. For a proposed time step $t_n \to t_{n+1}$ (time step size $h_n = t_{n+1} - t_n$), the local truncation error may be estimated by taking one step with h_n and two steps of $h_n/2$, giving solutions of $\vec{y}^{*}_{n+1}$ and $\vec{y}^{**}_{n+1}$, respectively. The local error may then be estimated

as

$$l_n \approx \left\| c_n \left(\frac{h_n}{2} \right)^{p+1} \right\| \approx \frac{\left\| \vec{y}^{*}_{n+1} - \vec{y}^{**}_{n+1} \right\|}{2^p - 1}, \tag{2.40}$$

where p is the global truncation error of the temporal integration method, and c_n is a constant [23]. The goal of time step adaptation is to keep this error below a certain tolerance. Typically, the tolerance τ is expressed as the local truncation error per unit step:

$$\frac{l_n}{h_n} \leq \tau. \tag{2.41}$$

If this inequality is false, then the proposed time step size is rejected, and a smaller time step size is chosen. For the next step, the proposed time step size is chosen such that the local truncation error per unit step estimate from the previous step exactly meets the tolerance, thereby increasing the time step size as stability constraints allow.

CHAPTER III

RESULTS

The results in this chapter are a comparison of results for a number of temporal integration methods, which are summarized in Table 1, where N_s represents the number of stages taken in a single time step, p is the theoretical convergence order, and the "Contains Explicit Stage" column lists whether the method has at least one explicit stage. Of primary interest is the comparison between the Rosenbrock family and the SDIRK family, as this will highlight the main advantages and disadvantages of using the linearization approach that Rosenbrock methods make.

Table 1. Temporal Integration Methods Compared in This Thesis

Abbreviation	**Family**	N_s	p	**Contains Explicit Stage**
BE	Theta	1	1	no
CN	Theta	1	2	no
GRK4A	Rosenbrock	4	4	no
GRK4Sn	Rosenbrock	4	4	no
GRK4T	Rosenbrock	4	4	no
RODASP	Rosenbrock	6	4	no
ROS22	Rosenbrock	2	2	no
ROS3P	Rosenbrock	3	3	no
SDIRK22	SDIRK	2	2	no
SDIRK23	SDIRK	2	3	yes
SDIRK33	SDIRK	3	3	no
SDIRK332	SDIRK	3	3	no
SDIRK34	SDIRK	3	4	yes
SDIRK45	SDIRK	4	5	yes
SDIRK543	SDIRK	5	4	no
SDIRK643	SDIRK	6	4	no

Linear results

Linear PRKE benchmarks convergence results

A total of seven linear PRKE benchmarks were chosen to verify theoretical convergence orders (truncation error orders) for each temporal integration method. These benchmarks were taken from [10] and are given in Table 2. These benchmarks vary in

the reactor data used (precursor group decay constants, precursor group fractions, and mean neutron lifetime), the number of precursor groups, and the reactivity function.

Table 2. Linear Benchmark Definitions

Benchmark	N_p	**Data Set**	**Reactivity Function**
1	6	fast	+\$0.5 step
2	6	thermal	-\$0.5 step
3	6	thermal	+\$1.0 step
4	6	thermal	+\$0.1/s ramp
5	6	fast	+\$1.0/s ramp
6	6	thermal	zigzag
7	1	1-group	sinusoidal

Table 3 gives the definitions for the reactivity functions. All of these reactivity functions are time-dependent, with the exception of "step". The time-dependent reactivity functions make the system non-autonomous (an explicit function of the independent variable, t), which is an important distinction for the evaluation of Rosenbrock methods because the linearization includes the partial derivative $\partial \vec{f}/\partial t$.

Table 3. Reactivity Function Definitions

Reactivity Function	$\rho(t)$	
step	ρ_0	
ramp	$mt + \rho_0$	
zigzag	mt	$0 \leq t < 0.5$
	$-m\,(t - 0.5) + C$	$0.5 \leq t < 1$
	$m\,(t - 1)$	$1 \leq t < 1.5$
	C	$1.5 \leq t < 10$
sinusoidal	$\rho_{max} \sin(At)$	

The reactor data sets "fast" and "thermal" correspond to data typical of fast spectrum reactors and thermal spectrum reactors. Fast spectrum reactors have a very short mean neutron lifetime, which causes rapid changes in neutron population and power. This short neutron lifetime stiffens the system because of the disparity of time scales in the problem. The data values are given in Table 4.

Table 4. Reactor Data Groups

	thermal		**fast**		**1-group**	
Group	λ_j [1/s]	β_j	λ_j [1/s]	β_j	λ_j [1/s]	β_j
1	0.0127	0.000285	0.0129	0.0001672	0.077	0.0079
2	0.0317	0.0015975	0.0311	0.001232	N/A	N/A
3	0.115	0.00141	0.134	0.0009504	N/A	N/A
4	0.311	0.0030525	0.331	0.001443	N/A	N/A
5	1.40	0.00096	1.26	0.0004534	N/A	N/A
6	3.87	0.000195	3.21	0.000154	N/A	N/A
β	0.0075		0.0044		0.0079	
Λ [s]	0.0005		10^{-7}		10^{-8}	

Figures 1 through 7 show the convergence results for power for the seven benchmarks. For Benchmarks 1, 4, 5, and 7, the SDIRK methods with explicit stages were unstable, i.e., they produced unbounded solutions. For this reason, they were omitted from Figures 1, 4, 5, and 7. Explicit methods have much stricter time step size constraints than implicit methods because even after the rapidly dying modes of the solution have gone to nearly zero, the modes exist at the roundoff error level and can be resurrected by an amplification factor $G = y^{n+1}/y^n$ that is greater than or equal to one. Taking a large number of time steps N gives a total amplification of this roundoff error equal to G^N. For example, with roundoff error at 1e-15, an amplification factor of 1.2, and $N = 1000$, the nearly-zero solution mode grows to 1.518e64, destroying the numerical solution.

The convergence plots show that the methods converge correctly but with some exceptions. CN and GRK4A in Benchmark 1 only reach theoretical convergence

behavior at fine time steps. ROS3P in Benchmark 4 deviates from theoretical behavior for coarse time steps. Benchmark 5 shows poor convergence behavior for all methods. The error is very large, going up to 10^4, which suggests instability in the system. Benchmark 6 shows that ROS3P and the GRK methods show regions of varying stability. Benchmark 7 shows very poor convergence results; the only methods that showed the correct convergence order were BE, ROS22, and SDIRK22. The inability of temporal integration methods to approach a level of error lower than 1e-14 is expected due to roundoff and truncation error in the reference solution, which was computed using a high order temporal integration method with fine time steps.

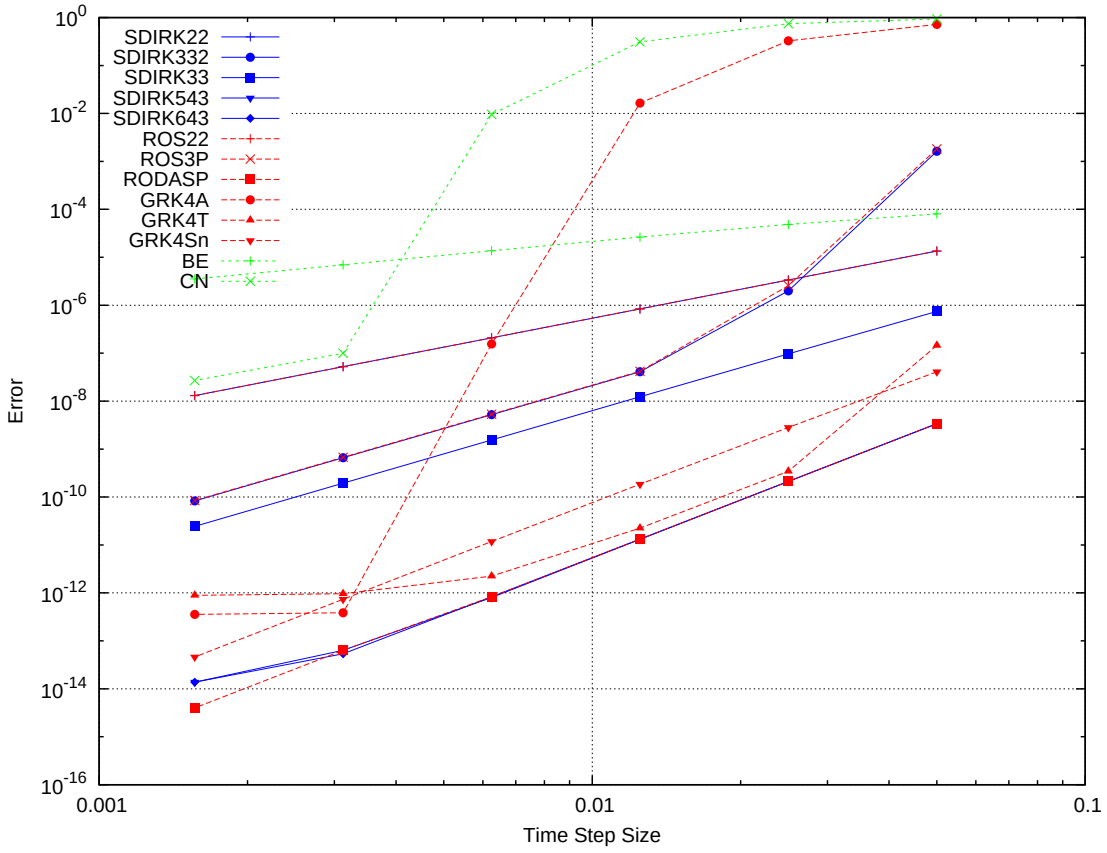


Fig. 1. Convergence for Linear PRKE Benchmark 1

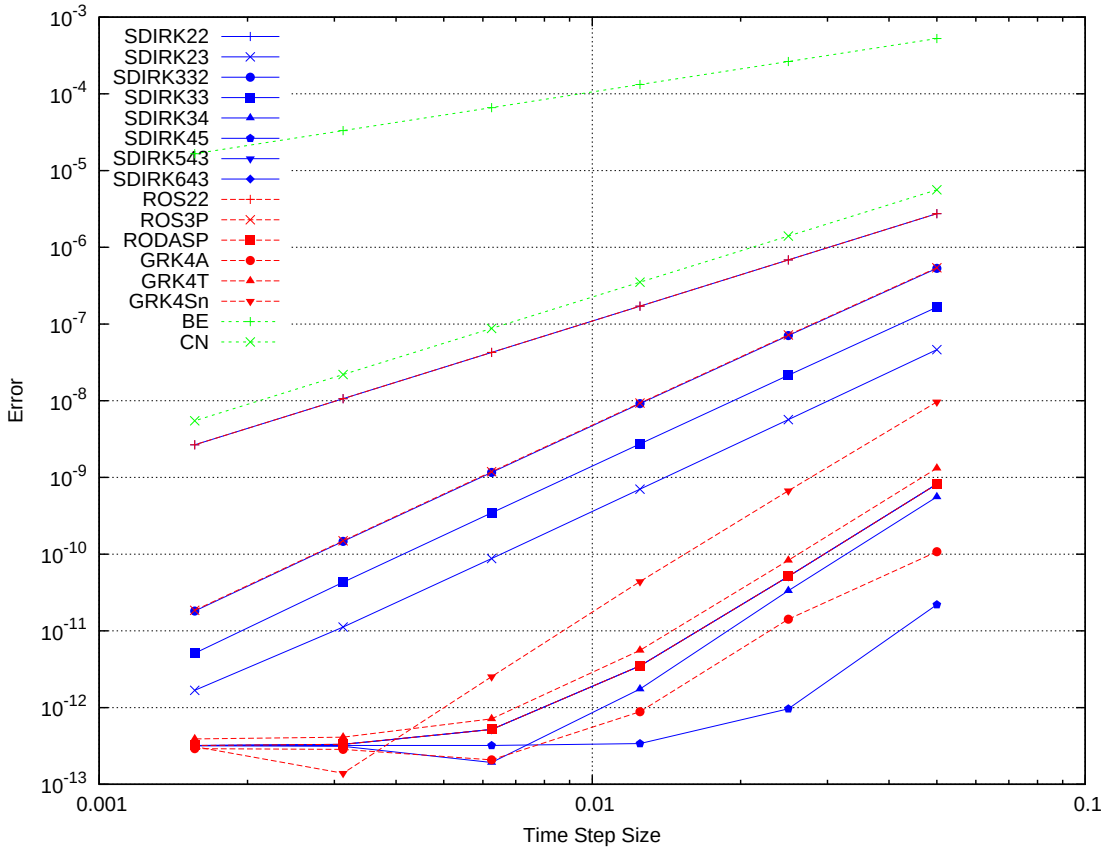


Fig. 2. Convergence for Linear PRKE Benchmark 2

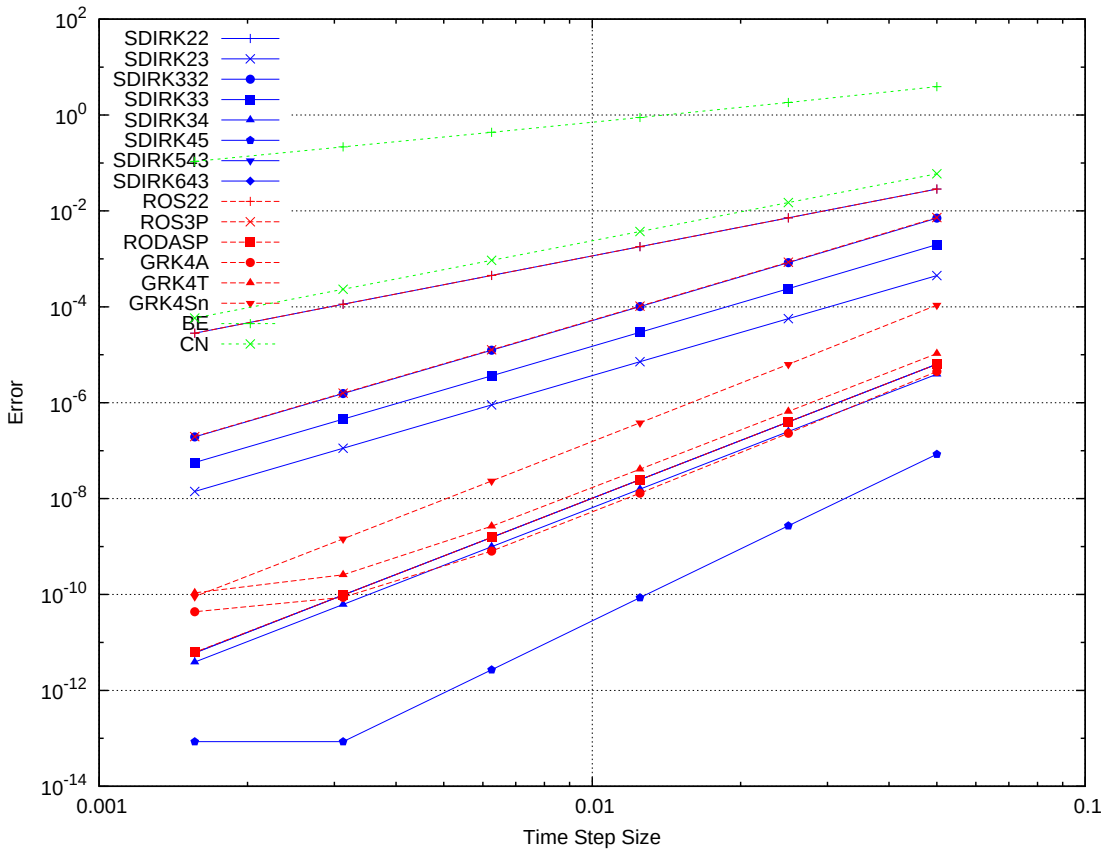


Fig. 3. Convergence for Linear PRKE Benchmark 3

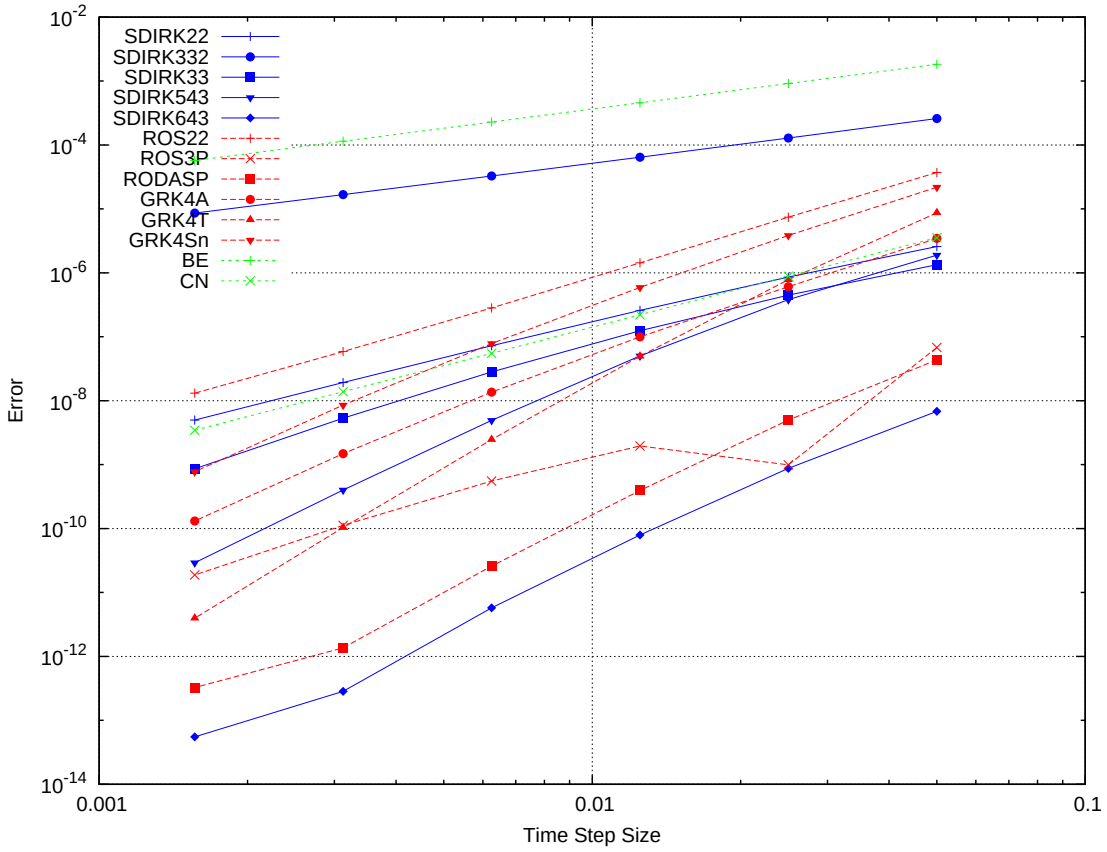


Fig. 4. Convergence for Linear PRKE Benchmark 4

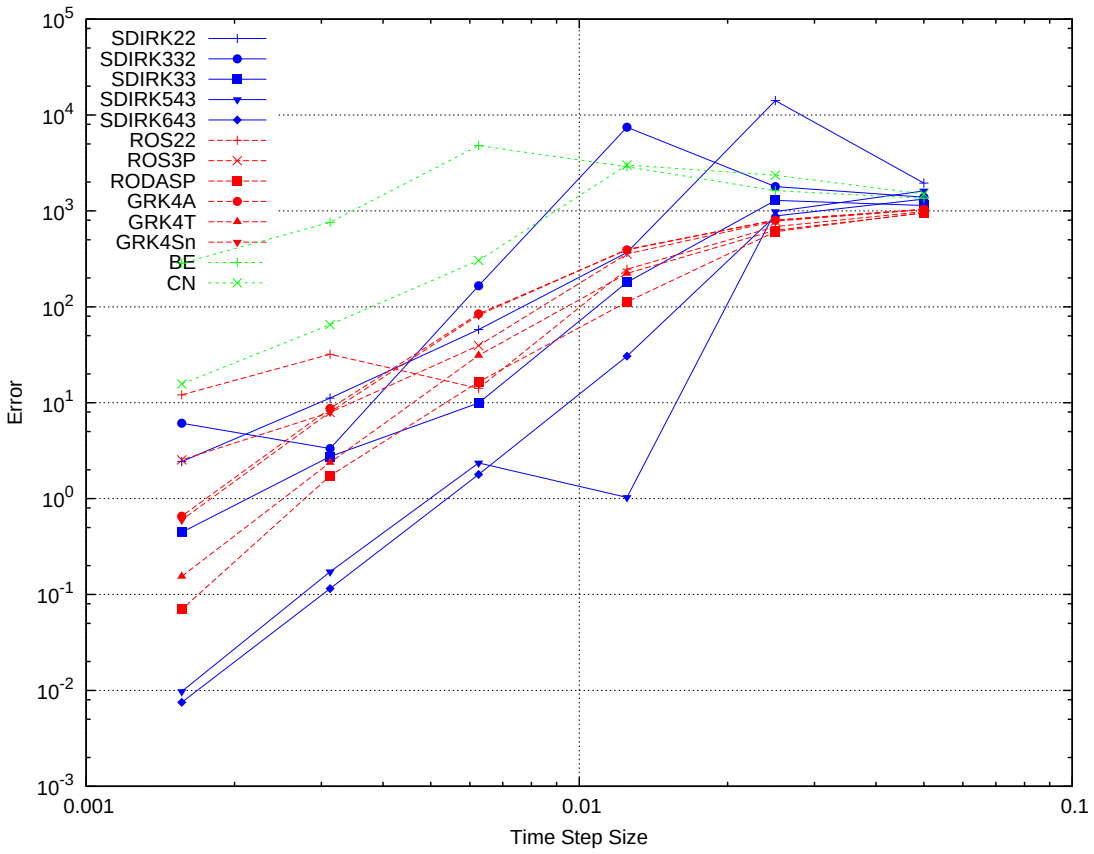


Fig. 5. Convergence for Linear PRKE Benchmark 5

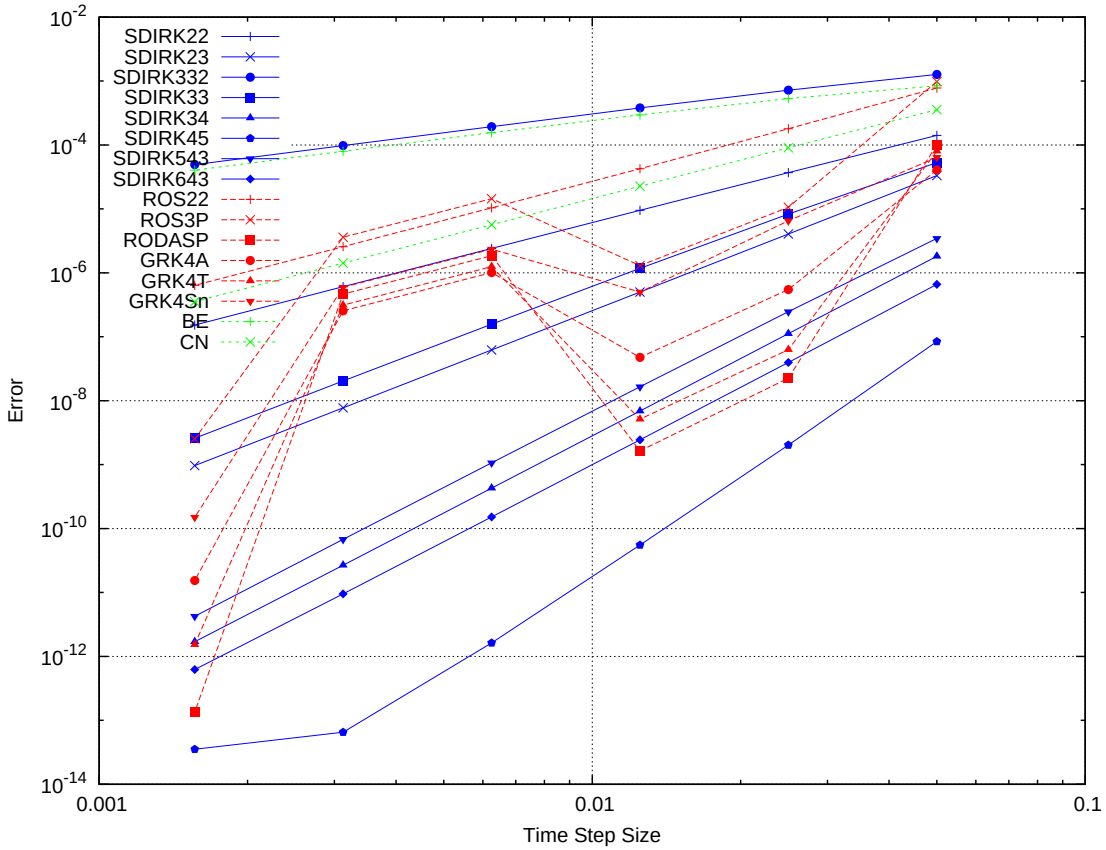


Fig. 6. Convergence for Linear PRKE Benchmark 6

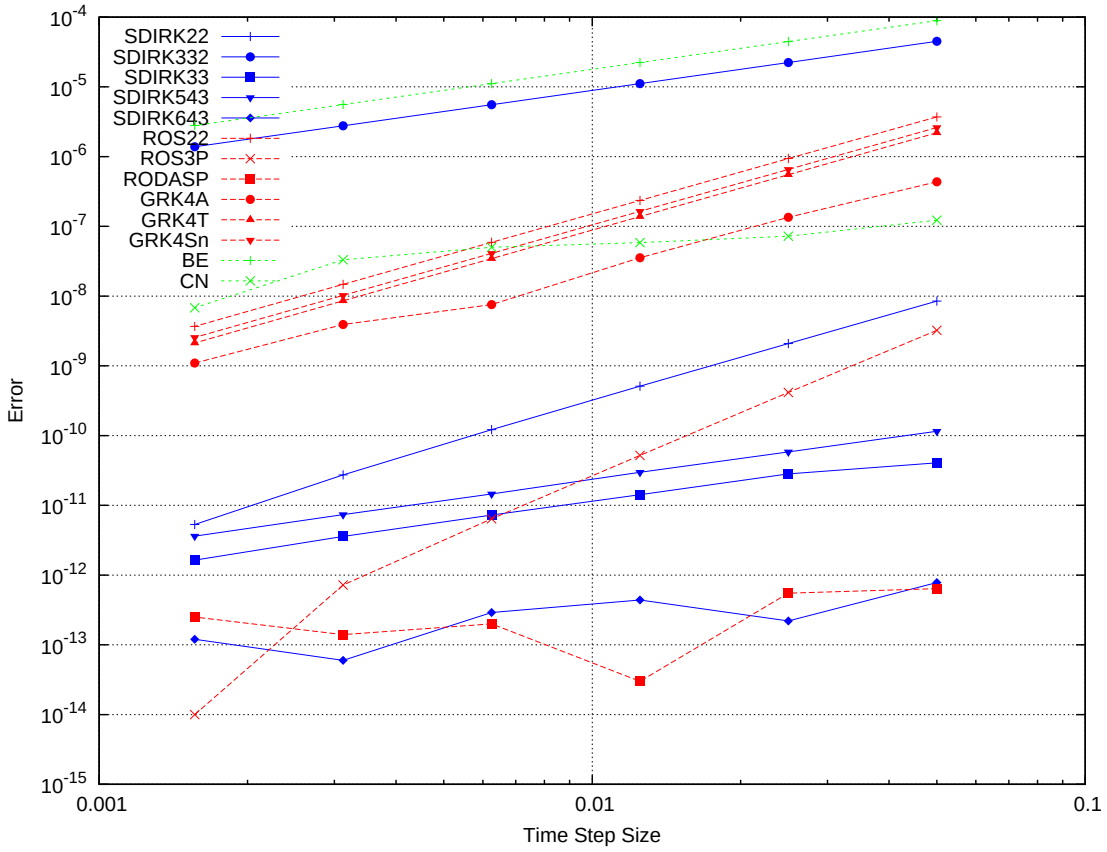


Fig. 7. Convergence for Linear PRKE Benchmark 7

While some linear PRKE benchmarks gave unstable numerical solutions for some time step sizes, the benchmarks proved that each method could produce theoretical convergence behavior for at least some of the systems examined, implying that the temporal integration methods were implemented correctly. Rosenbrock methods were verified to converge correctly for at least some of the non-autonomous systems, which is necessary because Rosenbrock methods require inclusion of $\partial \vec{f}/\partial t$.

Nonlinear results

Nonlinear PRKE convergence results

Table 5 shows the nonlinear PRKE cases run. Two competing feedback models were used:

1. T_{fuel}, T_{cool}: Fuel Temperature and Coolant Temperature Feedback.
2. T_r: Reactor Temperature Feedback.

For the T_{fuel}, T_{cool} model, two competing sets of property functions were used, referred to as "complicated" and "simplified".

Table 5. Nonlinear PRKE Cases Definitions

Case	**Feedback Model**	**Reactivity Function**	**Data Set**
1	T_{fuel}, T_{cool}	+\$0.5 step	complicated
2	T_{fuel}, T_{cool}	+\$0.5 step	simplified
3	T_{fuel}, T_{cool}	+\$1.0/s ramp	simplified
4	T_r	+\$0.5 step	N/A
5	T_r	+\$1.35/s ramp for 0.05 s	N/A

Figures 8 through 12 show the convergence results for the nonlinear PRKE cases. It should be noted that for Case 5, the run with the coarsest time step size, 0.05 s, did not converge for methods BE and ROS3P, so these were omitted. Otherwise, all methods and time step sizes converged, including methods containing an explicit stage.

For Case 3, SDIRK332 gave first order instead of third order. Case 5 showed that GRK4T, SDIRK643, and SDIRK45 gave a sharp increase in error for the finest

time step size. All of the Rosenbrock methods showed behavior suggesting varying regions of stability.

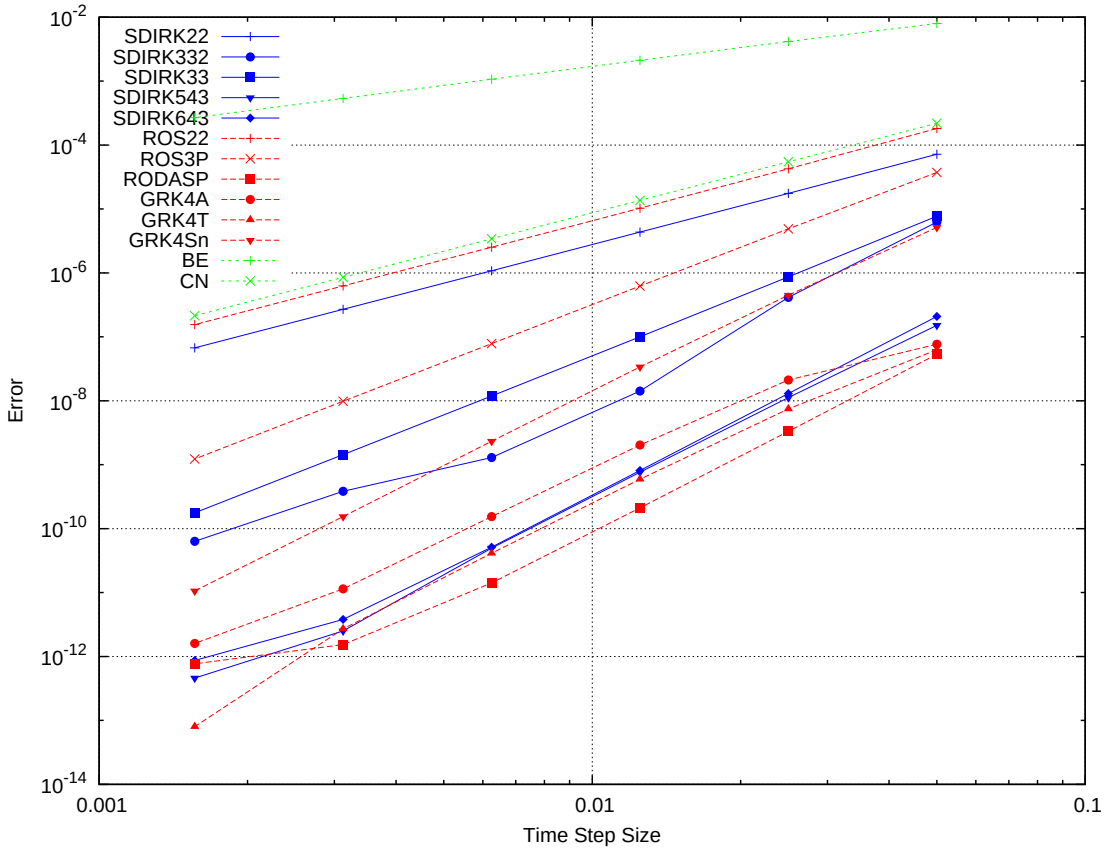


Fig. 8. Convergence for Nonlinear PRKE Case 1

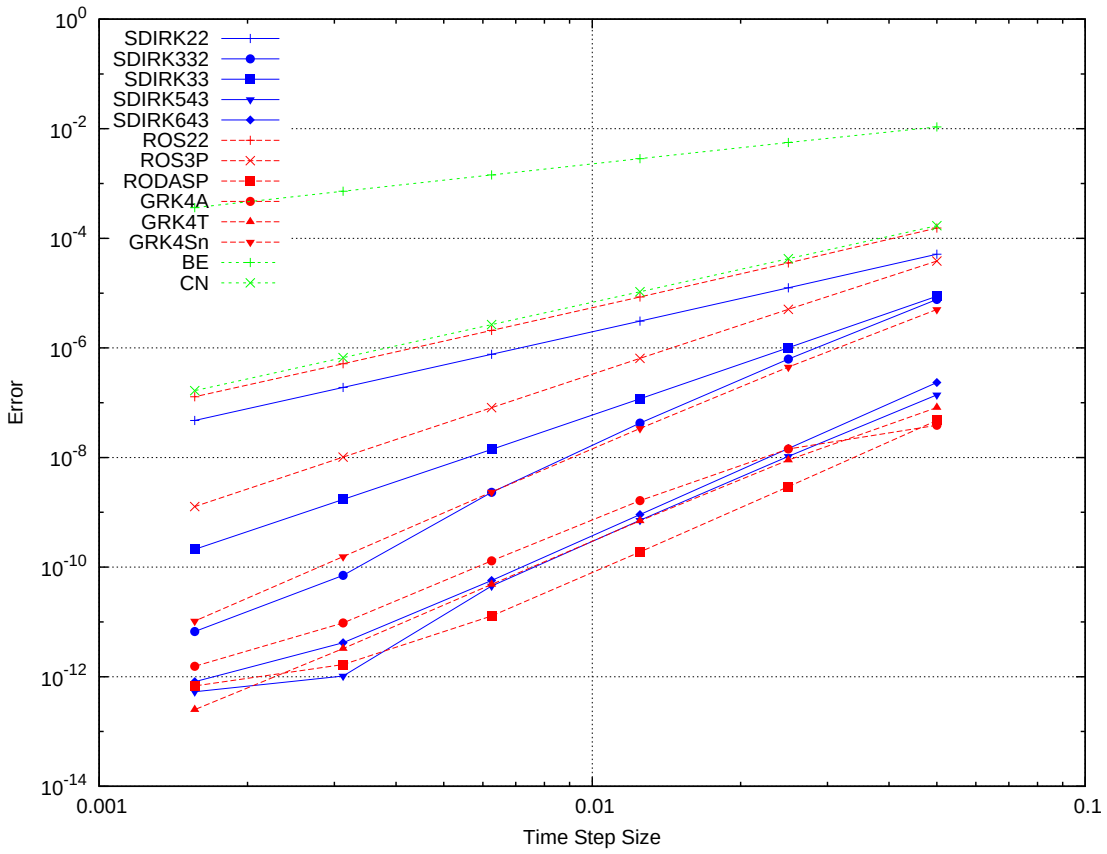


Fig. 9. Convergence for Nonlinear PRKE Case 2

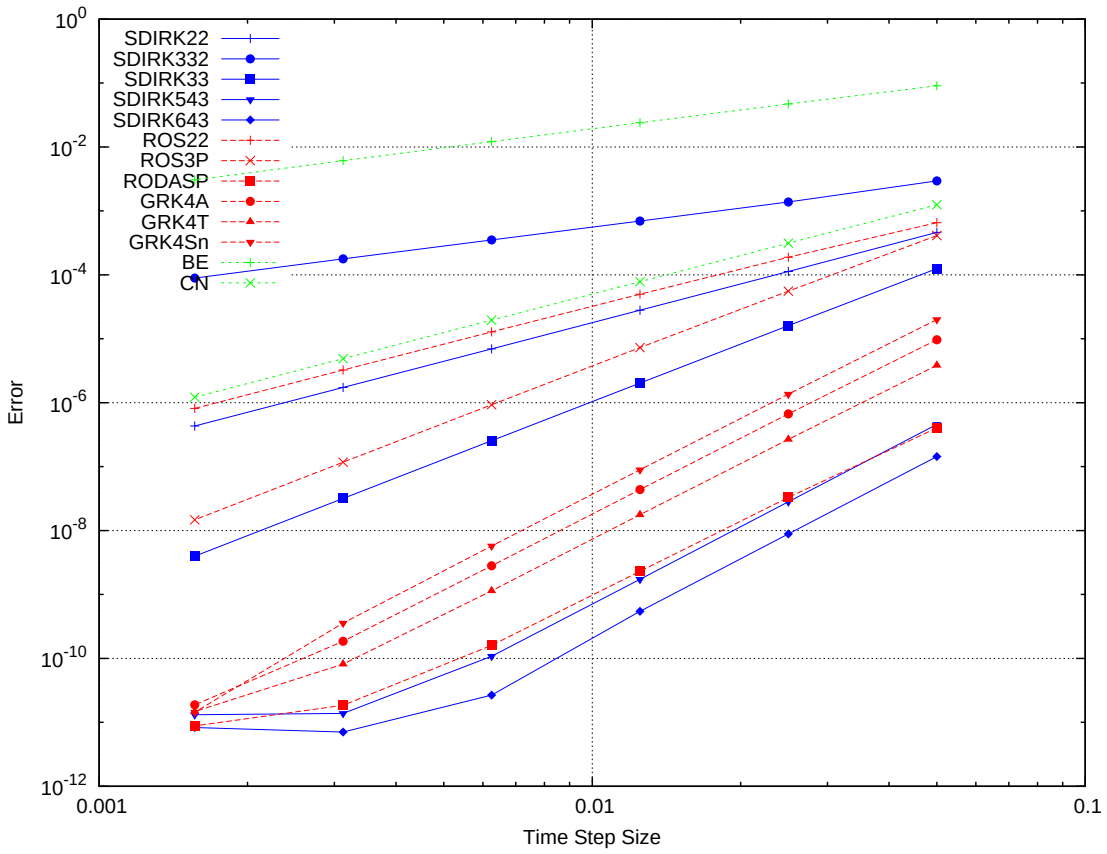


Fig. 10. Convergence for Nonlinear PRKE Case 3

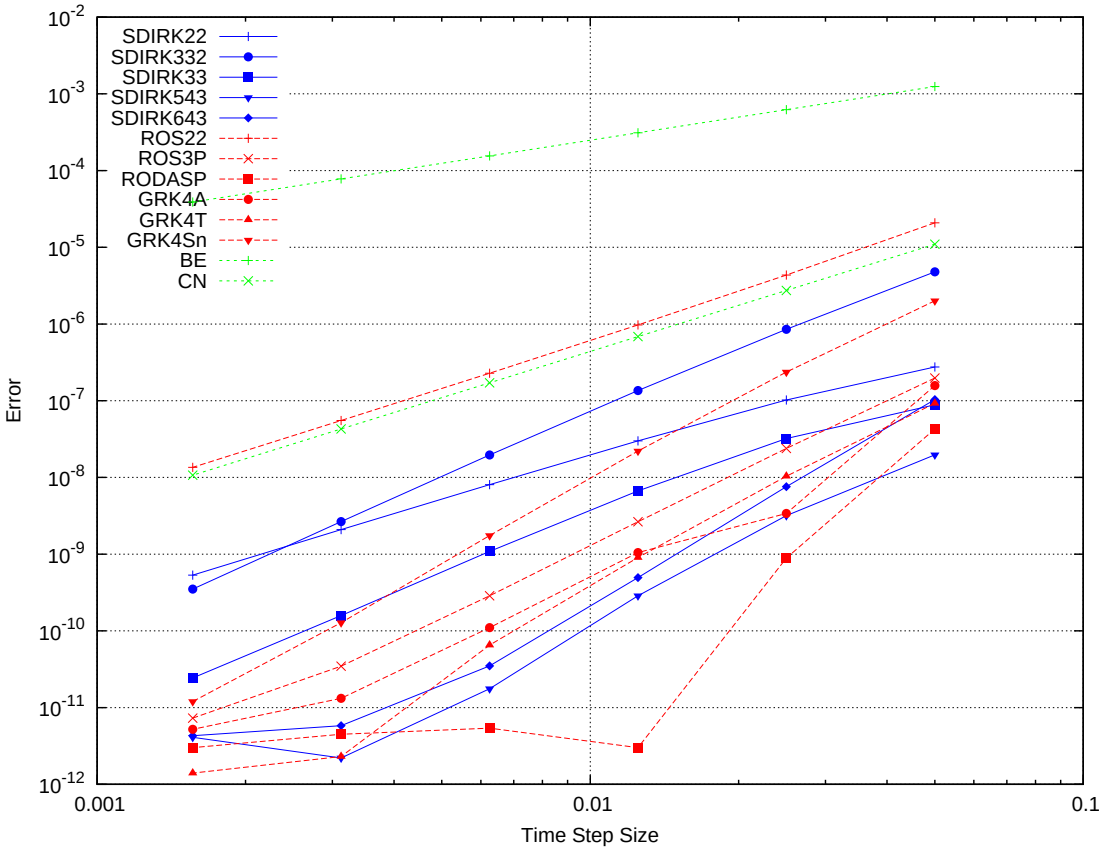


Fig. 11. Convergence for Nonlinear PRKE Case 4

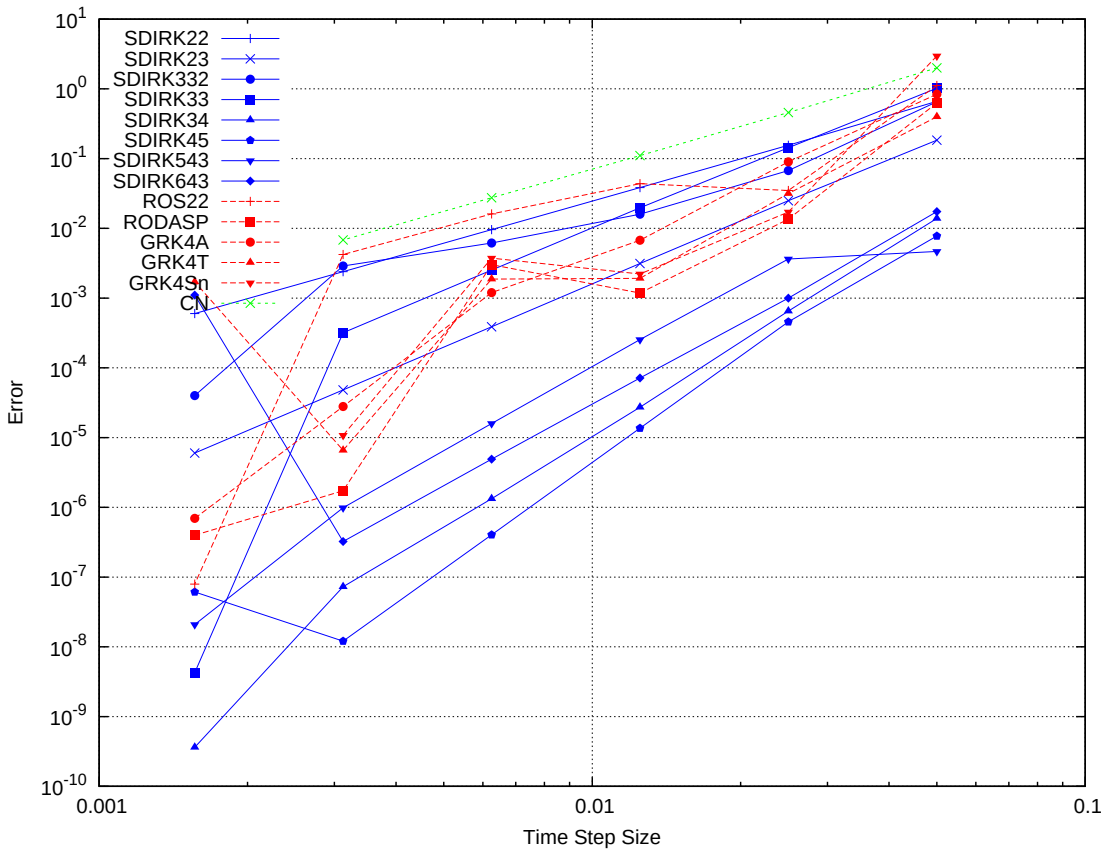


Fig. 12. Convergence for Nonlinear PRKE Case 5

Nonlinear PRKE efficacy results

To evaluate the merits of the temporal integration methods, efficacy plots were created for each nonlinear case, showing the solution error versus the computational effort (measured in number of linear solves) for all methods. For a desired level of accuracy (measured in error), the "superior" method is that which obtains that error with the least computational effort. For these plots, one can imagine a horizontal line at the desired accuracy level, and going from left to right, find the most "superior" method to the most "inferior" method for the given simulation case. These plots are shown in Figures 13 through 17.

Cases 1 through 3 give the expected results; for a given convergence order, Rosenbrock methods are superior to SDIRK methods with the exception of SDIRK methods containing an explicit stage. ROS22 outperforms SDIRK22, ROS3P outperforms SDIRK33 and SDIRK332, and GRK4A, GRK4T, GRK4Sn, and RODASP outperform SDIRK543 and SDIRK643. While the methods containing an explicit stage, SDIRK23 and SDIRK34, outperform Rosenbrock methods, these methods are unfavorable for many systems due to their instability. Case 4 shows that SDIRK22 actually outperforms ROS22 because the Rosenbrock linearization decreased the accuracy so much. Case 5 shows that the same general results are obtained as in Cases 1 through 3, but there are regions for which Rosenbrock methods achieve significantly less accuracy.

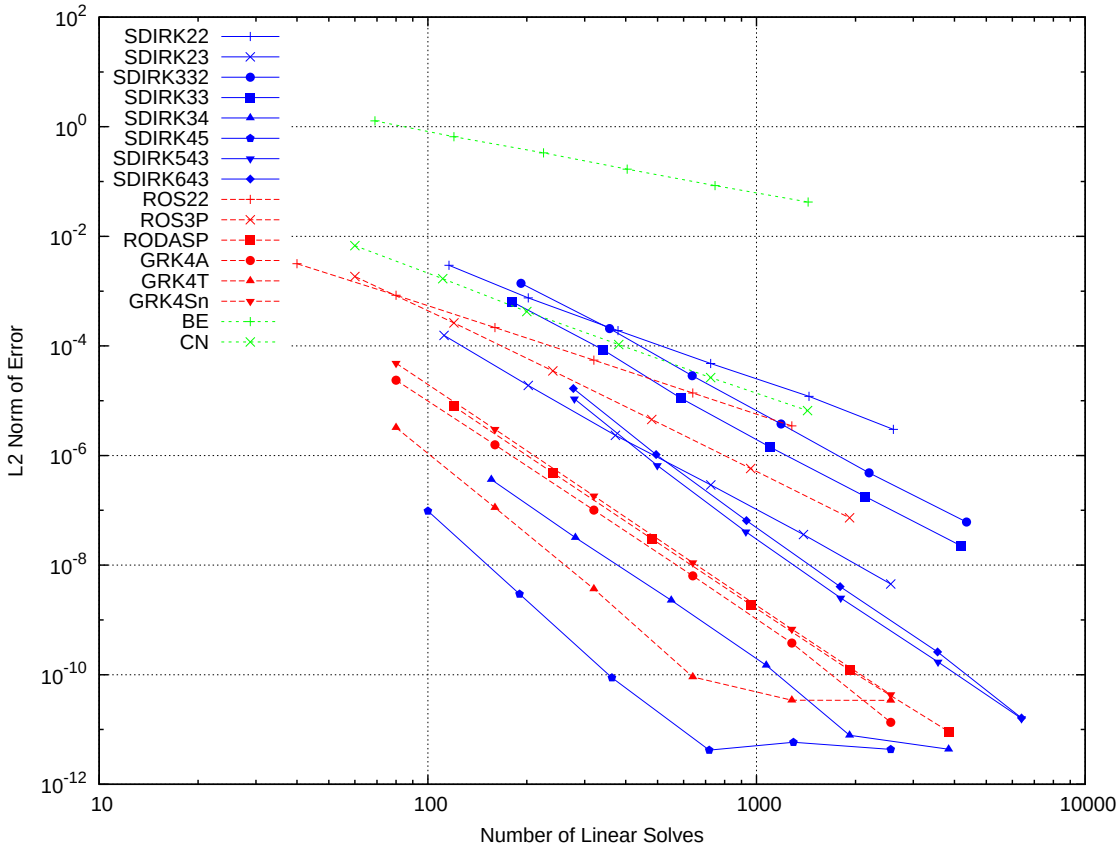


Fig. 13. Efficacy Plot for Nonlinear PRKE Case 1

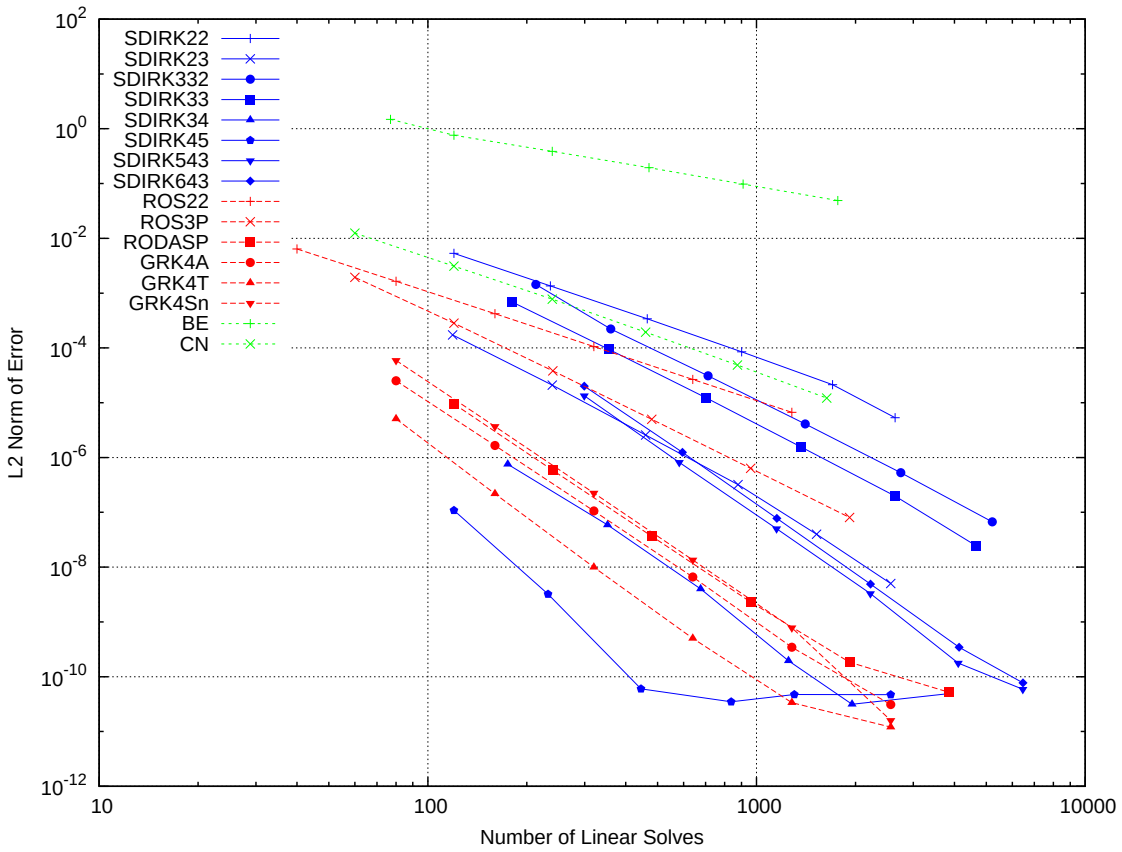


Fig. 14. Efficacy Plot for Nonlinear PRKE Case 2

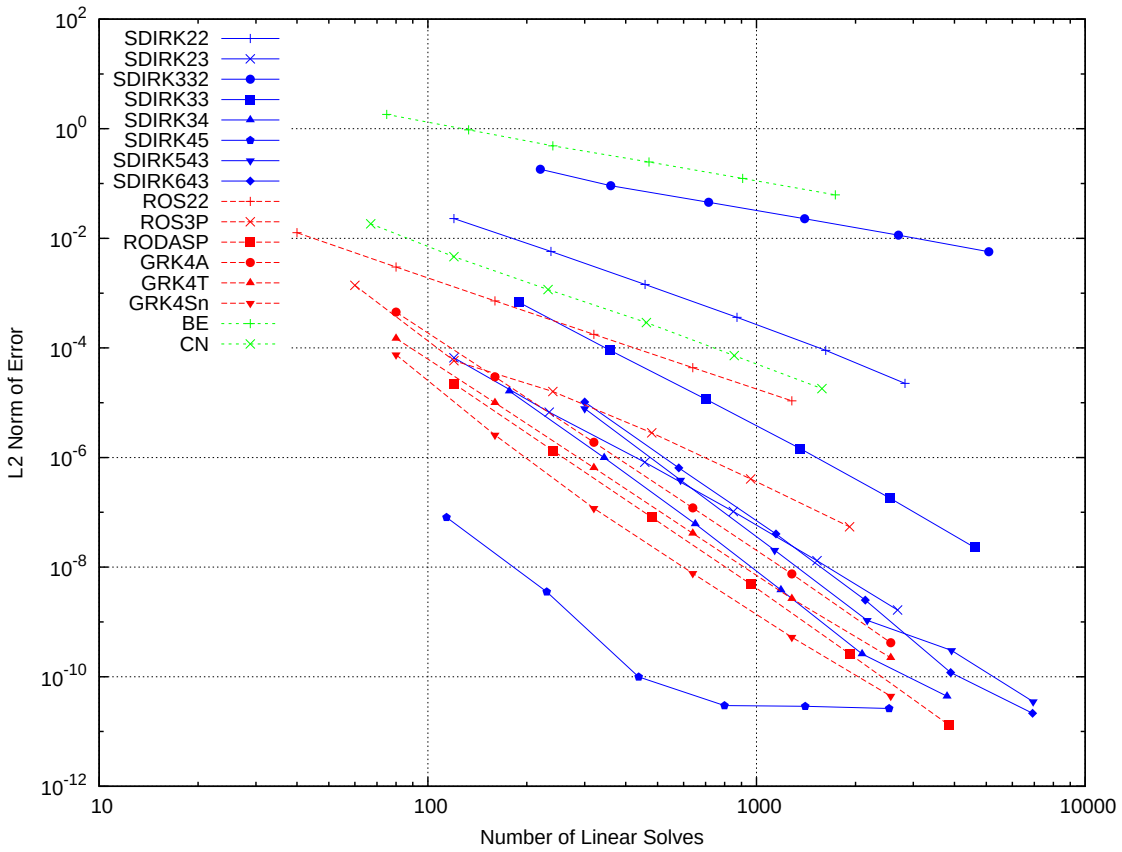


Fig. 15. Efficacy Plot for Nonlinear PRKE Case 3

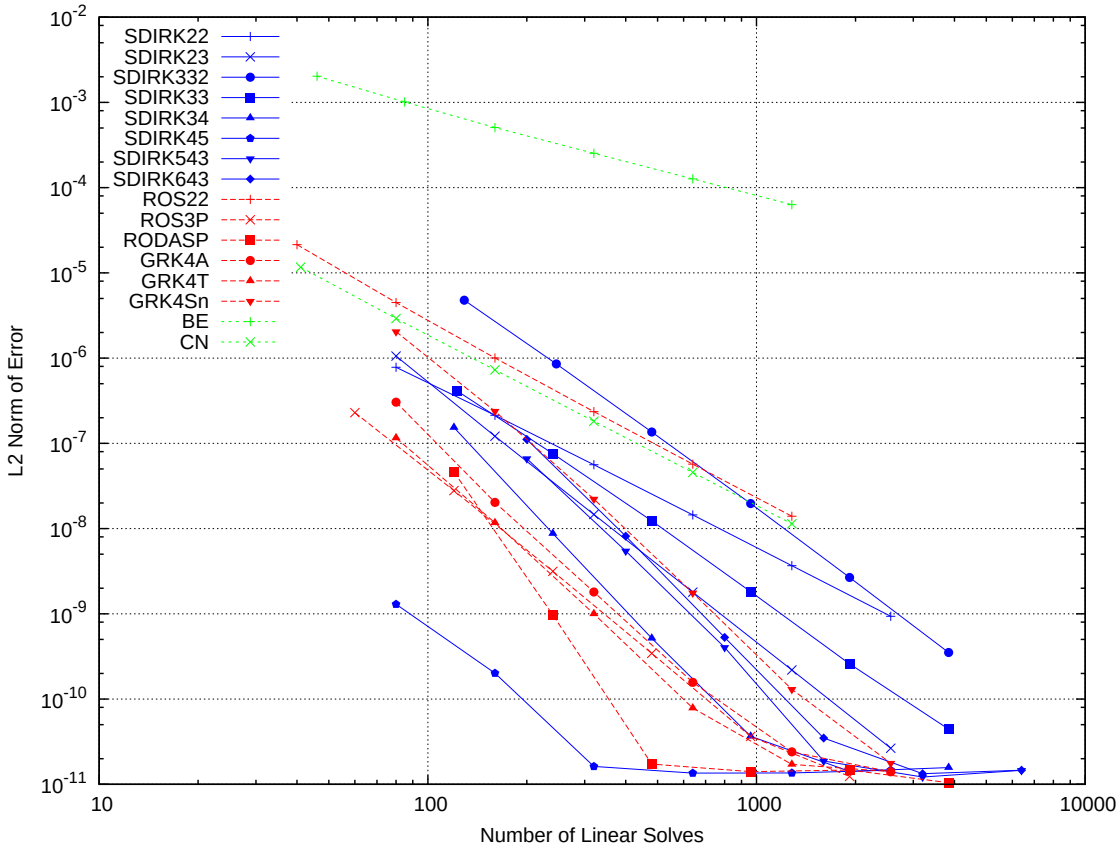


Fig. 16. Efficacy Plot for Nonlinear PRKE Case 4

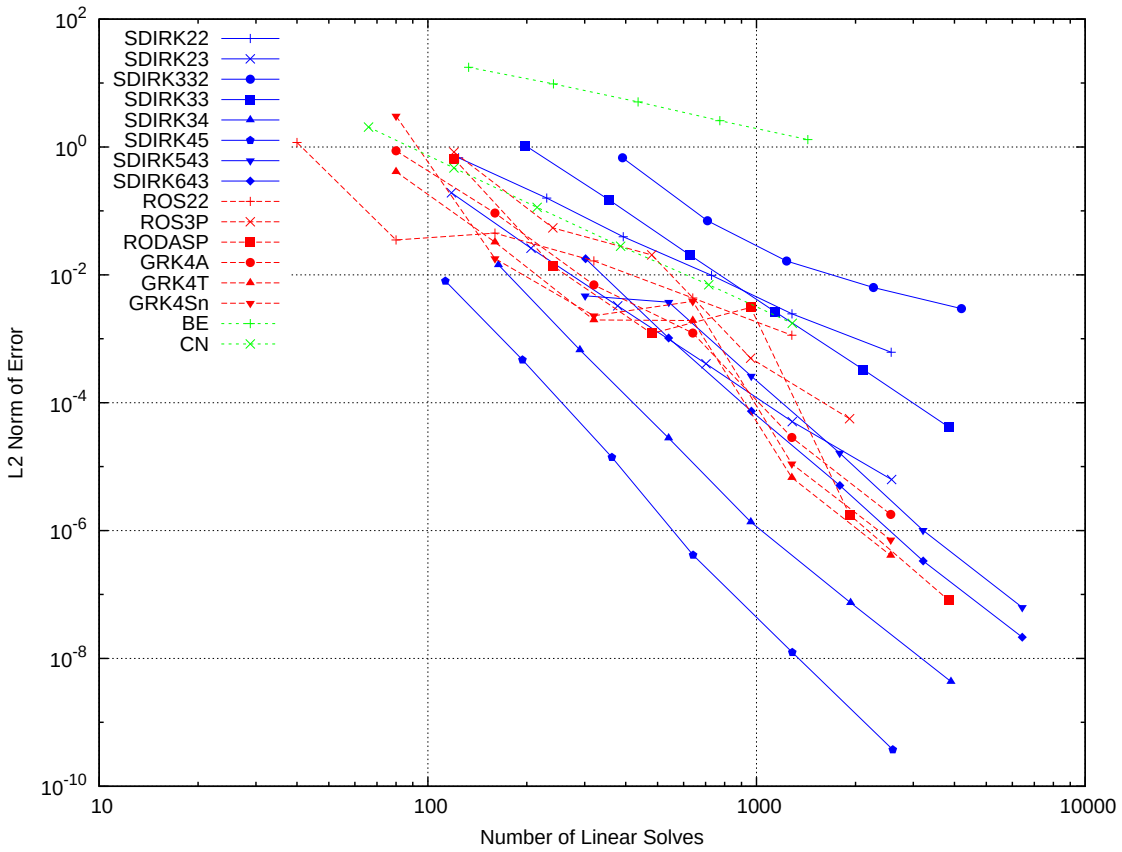


Fig. 17. Efficacy Plot for Nonlinear PRKE Case 5

Nonlinear radiative transfer efficacy results

The radiative transfer problem is very stiff, and thus traditional constant time-stepping is extremely unfavorable because very fine steps are necessary for stability at the beginning of the transient. Thus, time step size adaptation was necessary to compute the final numerical solution. Since the time step size is no longer constant, other parameters such as tolerances must be varied to produce runs with varying degrees of work and accuracy for efficacy plots. These plots were produced but later found to be invalid due to implementation errors, so they are not included in this thesis. These errors concern the computation of the Jacobian matrix, which was

computed numerically using finite differences instead of analytically, as the PRKE were. Thus, no valid results are available for Rosenbrock methods for the radiative transfer problem.

CHAPTER IV

CONCLUSION

The linear PRKE benchmarks proved that the temporal integration methods of interest all converged correctly for at some of the benchmarks, both autonomous and non-autonomous. Nonlinear PRKE efficacy plotting showed that Rosenbrock methods were generally obtained solutions of a given accuracy more efficiently than SDIRK methods of the same convergence order. The exceptions were the explicit stage SDIRK methods SDIRK23 and SDIRK34, but it was shown in convergence studies of the linear benchmarks that these methods were unstable for stiff systems. SDIRK22, which contains no explicit stages, actually outperformed ROS22 in Case 4, which suggests that Rosenbrock methods aren't conclusively superior to SDIRK methods. Some non-autonomous systems such as that in Case 5 showed regions of varying stability for Rosenbrock methods.

The radiative transfer problem was incorrectly simulated for Rosenbrock methods due to an error in the finite difference approximation of the Jacobian matrix. Thus, no conclusions can be drawn from the radiative transfer simulations.

Although no valid radiative transfer results were obtained, Rosenbrock methods were shown to be more efficient than stable Runge-Kutta methods for the majority of the nonlinear PRKE cases simulated. Explicit-stage SDIRK methods outperformed Rosenbrock methods in those cases in which those SDIRK methods were stable. Thus, for a given nonlinear problem, it is suggested to use trial and error to determine the most efficient method that can obtain reliably stable solutions. This research has proved that Rosenbrock methods create stable, accurate solutions very efficiently, making it a viable choice for the solution of tightly coupled nonlinear systems.

REFERENCES

[1] P. Kaps, G. Wanner, A study of Rosenbrock-type methods of high order, Numerical Mathematics 38 (1981) 279–298.

[2] A. Wolfbrandt, A study of Rosenbrock processes with respect to order conditions and stiff stability, Numerische Mathematik 32 (1979) 1–16.

[3] H. Shintani, Modified Rosenbrock methods for stiff systems, Hiroshima Mathematics Journal 12 (3) (1982) 543–558.

[4] P. Kaps, P. Rentrop, Generalized Runge-Kutta methods of order four with stepsize control for stiff ordinary differential equations, Numerische Mathematik 33 (1979) 55–68.

[5] J. C. Verwer, E. J. Spee, J. C. Blom, W. Hundsdorfer, A second-order Rosenbrock method applied to photochemical dispersion problems, SIAM Journal of Scientific Computation 20 (4) (1999) 1456–1480.

[6] M. Günther, M. Hoschek, ROW methods adapted to electric circuit simulation packages, Journal of Computational and Applied Mathematics 82 (1997) 159–170.

[7] J. P. Meijaard, Application of Runge-Kutta-Rosenbrock methods to the analysis of flexible multibody systems, Multibody System Dynamics 10 (2003) 263–288.

[8] J. Sánchez, On the numerical solution of the point reactor kinetics equations by generalized Runge-Kutta methods, Nuclear Science and Engineering 103 (1989) 94–99.

[9] M. T. Sissaoui, J. Koclas, A. Hébert, Solution of the improved and generalized quasistatic methods by Kaps and Rentrop integration scheme with stepsize control, Annals of Nuclear Energy 22 (12) (1995) 763–774.

[10] X. Yang, T. Jevremovic, Revisiting the Rosenbrock numerical solutions of the reactor point kinetics equation with numerous examples, Nuclear Technology & Radiation Protection 1 (2009) 3–12.

[11] A. E. Aboanber, Y. M. Hamada, Generalized Runge-Kutta method for two- and three-dimensional space-time diffusion equations with a variable time step, Annals of Nuclear Energy 35 (6) (2008) 1024–1040.

[12] R. B. Lowrie, A comparison of implicit time integration methods for nonlinear relaxation and diffusion, Journal of Computational Physics 196 (2004) 566–590.

[13] C. C. Ober, J. N. Shadid, Studies on the accuracy of time-integration methods for the radiation-diffusion equations, Journal of Computational Physics 195 (2004) 743–772.

[14] E. E. Lewis, Fundamentals of Nuclear Reactor Physics, Elsevier, Burlington, MA, 2008.

[15] J. C. Ragusa, V. S. Mahadevan, Consistent and accurate schemes for coupled neutronics thermal-hydraulics reactor analysis, Nuclear Engineering and Design 239 (2009) 566–579.

[16] F. P. Incropera, D. P. Dewitt, T. L. Bergman, A. S. Lavine, Fundamentals of Heat and Mass Transfer, John Wiley & Sons, Hoboken, NJ, 2007.

[17] V. A. Mousseau, D. A. Knoll, W. J. Rider, Physics-based preconditioning and the Newton-Krylov method for non-equilibrium radiation diffusion, Journal of

Computational Physics 160 (2000) 743–765.

[18] V. A. Mousseau, D. A. Knoll, New physics-based preconditioning of implicit methods for non-equilibrium radiation diffusion, Journal of Computational Physics 190 (2003) 42.

[19] V. A. Mousseau, D. A. Knoll, Temporal accuracy of the nonequilibrium radiation diffusion equations applied to two-dimensional multimaterial simulations, Nuclear Science and Engineering 154 (2006) 174–189.

[20] R. H. Szilard, G. C. Pomraning, Numerical transport and diffusion methods in radiative transfer, Nuclear Science and Engineering 112 (1992) 256.

[21] J. D. Hoffman, Numerical Methods for Engineers and Scientists, Marcel Dekker, New York, NY, 2001.

[22] E. Hairer, S. Nørsett, G. Wanner, Solving Ordinary Differential Equations II: Stiff and Differential-algebraic Problems, Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1993.

[23] G. Dahlquist, Åke Björk, Numerical Methods, Prentice-Hall, Englewood Cliffs, NJ, 1974.

CONTACT INFORMATION

Name:	Joshua Edmund Hansel
Professional Address:	c/o Dr. Jean Ragusa Department of Nuclear Engineering Texas A&M University 3133 TAMU College Station, TX 77843-3133
Email Address:	joshhansel@tamu.edu
Education:	B.S., Nuclear Engineering, Texas A&M University, May 2011 Undergraduate Research Scholar National Society of Collegiate Scholars