

# University Honors College

## HONORS CONTRACT

**Course Eligibility: See Page Two**

**Please Print/Type**

**SEMESTER:** ☒ Fall ☐ Spring ☐ Summer **YEAR:** 2018 **ID#** M01250797

**DUE  
SEPT. 30th**

**DUE  
FEB. 15th**

**DUE  
MAY 30th**

STUDENT NAME	Cole	Jackson	Lane	E-MAIL	jlc2de@mtmail.mtsu.edu
	last	first	middle		

<b>MAJOR</b>	<b>Physics</b>	<b>CELL PHONE</b>	013-971-0888
		(if no cell #, then home #)	

<b>MINOR</b>	<b>Mathematics</b>	<b>GRADUATION SEMESTER</b>	<b>Fall 2018</b>	<b>GPA</b>	<b>3.875</b>
--------------	--------------------	----------------------------	------------------	------------	--------------

<b>COURSE TITLE</b>	<b>MATH 3180 - Introduction to Numerical Analysis</b>	<b>CREDIT HOURS</b>	<b>3.000</b>
---------------------	---	---------------------	--------------

**COURSE RUBRIC and NUMBER** N/A **CRN#** 80907

**INSTRUCTOR** Dr. Suk Jai Seo **E-MAIL** Suk.Seo@mtsu.edu

Describe the additional course requirements. ***Specifically*** address how the work proposed by this contract is different from and/or expands upon the requirements for the regular course.

- ☒ The regular class syllabus is attached
  - ☒ A one-page detailed description of the Honors component is attached (see back page)
- The Honors Contract will not be approved without BOTH of these documents!**

We, the undersigned, hereby agree to pursue the project detailed above during this semester. As a result of the successful completion of this work, an Honors designation will be applied to the above-named course as it appears on the student's official transcript.

**Approval Signatures**

<hr style="border: 0; border-top: 1px solid black; margin-bottom: 5px;"/> Student _____ Date _____	<hr style="border: 0; border-top: 1px solid black; margin-bottom: 5px;"/> Instructor _____ Date _____
--	---

Honors College Academic Advisor	Date
---------------------------------	------

Honors College Dean or Associate Dean \_\_\_\_\_ Date \_\_\_\_\_

## NOTIFICATION OF COMPLETION

The attached e-mail notification certifies that \_\_\_\_\_ has completed the requirements above and has earned University Honors credit for the identified course.

Honors College Executive Secretary \_\_\_\_\_ Date \_\_\_\_\_

## Guidelines for the Honors Contract

### UPPER-DIVISION CREDIT

Honors students may obtain Honors credit in a Non-Honors upper-division course in his or her major, concentration, or minor with an Honors Contract. Students may receive Honors credit in unlimited upper-division courses, but Honors College graduates can **only apply a maximum of four (4) hours of Honors Contract hours** toward the eleven-hour, upper-division Honors College requirement.

### LOWER-DIVISION CREDIT

Honors students may obtain Honors credit in a Non-Honors lower-division course (2000 level) that is **never** offered as an Honors section. This does not apply if an Honors course isn't offered the semester you want to take it (i.e. a course offered every fall, but not spring, would not qualify in the spring). Students may receive Honors credit in unlimited lower-division courses (2000 level), but Honors students can **only apply a maximum of six (6) hours of Honors Contract hours** toward the 18-hour, lower-division Honors College requirement.

**FRESHMAN LEVEL 1000 COURSES DO NOT QUALIFY.**

The Honors Contract should create a partnership of mutual benefit to both the student and the faculty member. For an Honors student, the Contract should involve a project or activity that delves deeper in the course material and results in a better appreciation and understanding of the subject matter that can be used as a positive experience in preparation for future goals. Faculty members can use the Honors Contract to try innovative or interesting activities or projects that are not practical to do with an entire class. In all cases, the Honors component in the Honors Contract should target one or more of the following areas: scholarship, leadership, or service. **Questions regarding the nature of a specific Honors Contract can be directed to Dr. John Vile, ([John.Vile@mtsu.edu](mailto:John.Vile@mtsu.edu)), Dean, 615-898-2152, or Dr. Philip Phillips ([Philip.Phillips@mtsu.edu](mailto:Philip.Phillips@mtsu.edu)), Interim Associate Dean, at 615-898-2699.**

The Honors Contract must **explicitly state** the work the student will undertake to earn the Honors designation. That work should be more rigorous than and/or go above and beyond what is expected of the other students in the course. For instance:

- Students may undertake an additional, or more demanding research project;
- give a special presentation to the class;
- participate in a workshop;
- expand a paper for presentation or submission at a conference;
- work as a research assistant;
- be involved in an internship;
- conduct field work;
- create/exhibit exceptional art work; or
- give a performance or concert.

The contract should also state **specifically** how the work proposed by the Contract is different from the requirements of the other students in the course. For example:

Katie will write a research paper. **WILL NOT MERIT APPROVAL**

Katie will write a longer research paper. **WILL NOT MERIT APPROVAL**

Katie will complete additional projects to be decided upon as the course progresses. **WILL NOT MERIT APPROVAL**

The correct way to address the Honors Contract should be similar to, or as detailed as:

For Honors credit, Katie will write a 15-page research paper on a theory of ethics not covered in this class. This extra writing assignment will require Katie to use at least two other outside resources and to apply in-depth research methods and critical thinking.

At the end of the semester, the professor should send an e-mail confirmation (to [Karen.Demonbreum@mtsu.edu](mailto:Karen.Demonbreum@mtsu.edu)) indicating that the student has completed the Honors component of this course. This should be done no later than seven (7) days after the final exam for the course. The Honors College will then notify the Records Office to assign the Honors designation on the student's transcript.

# Introduction to Numerical Analysis, CSCI/MATH 3180

## Fall 2018

**Instructor:** Dr. Suk Jai Seo

**Office:** KOM 303-A (904-8292)

**Class time:** TR 2:40-4:05pm

**Office hours:** TR 12:45 – 2:00 pm, TR 4:05-5:00pm

Other times by appointment and generally available seven days a week via email. See below for the details.

**Email address:** [Suk.Seo@mtsu.edu](mailto:Suk.Seo@mtsu.edu)

**D2L:** The D2L page for this class will contain links to course information, including this syllabus and some assignments. Access D2L through PipelineMT or directly using <https://elearn.mtsu.edu/>.

**Catalog Description:** Topics include series approximation, finite differences interpolation, summation, numerical differentiation and integration, iteration, curve fitting, systems of equations and matrices, and error analysis.

**Prerequisites:** MATH 1920 and CSCI 1170, each with a grade of C or higher.

**Textbook:** Numerical Mathematics and Computing (7th Edition) by Cheney and Kincaid

**Learning Outcomes:** Upon completion of this course, students should be able to

- understand a variety of methods by which a modern digital computer can be used to solve numerical problems
- understand how errors arise in numerical computing and methods for detecting, predicting and controlling errors
- have sharpened programming and problem solving skills

### Course Communication:

Students are responsible for monitoring for MTMAIL **daily**. Following MTSU's FERPA-based [e-mail policies](#), all course-related e-mail will be sent to your [MTMAIL](#) account; in turn, you are **required** to use your **MTMAIL** account when communicating with the instructor.

Emails sent to the instructor must contain the name of the sender, the subject, and the detailed message.

**Do not try to contact the instructor through D2L.**

Use [Suk.seo@mtsu.edu](mailto:Suk.seo@mtsu.edu)

### Use of personal electronic devices in the classroom:

Cell phones, laptops, tablets, and other electronic devices must be turned off and put away during class unless the instructor determines that these devices are allowed to be used in the class. Students are not permitted to take photos or record any part of a class/lab unless explicitly granted permission by the instructor or the MTSU Disability Access Center. Sanctions for violation of this policy will be determined by the instructor and may include dismissal from the class, attendance penalties or loss of class participation points, zero grades on quizzes or examinations, failure in the class, or other penalties that the instructor determines to be appropriate.

All course materials are protected by the law of copyright. Students are not authorized to use, reuse, reproduce, distribute, broadcast or publish the course materials, or any part of the course materials, in any medium, including via the internet and social media sites. This means that students are not allowed to photograph or reproduce course materials, and that students are not allowed to record lectures, except as provided for by an approved ADA request.

**Academic honesty:** *The Computer Science Department's [Policy on Academic Integrity](#) applies to this course. All work for this class (including exams, homework, and labs) is to be done on an individual basis. Unless otherwise directed, work alone on lab assignments. The penalty for unauthorized collaboration will range from a grade of zero for an assignment to a failing grade for the course.*

**Attendance:** Attendance is required and absences do not excuse one from class responsibilities. ***If for some unavoidable reason you must miss class, obtain class notes, handouts, and assignments from another class member.*** You are expected to be on time for class. Consistent lateness to class is disruptive and is considered to be

disrespectful. It is best to come late, however, rather than not at all!

Attendance is determined/judged/counted by the student being in attendance through the lecture and that student signing the name on the daily attendance sheet. It is each student's responsibility to locate and sign this sheet each class period during the class.

**Students failing to attend the first two class meetings will be dropped from the class.**

**Grading policy:**

- **Examination grades** – there will be four examinations counting 50 points each and a final examination counting 100 points. These exams will cover lectures, assigned readings, and labs. **No make-up exams will be given.** *If you miss a regularly scheduled exam, 50% of the final exam score will replace this exam.* If an exam is not missed then 50% of the final exam score can replace the lowest test score.
- **Laboratory assignments** - There will be ten lab assignments and they will be worth 100 points in total.
- **Quizzes** - There will be a quiz daily and they will be worth 100 points in total.

**Point System:** 500 total assigned points

- A: (450-500 pts)
- B: (400-449 pts)
- C: (350-399 pts)
- D: (300-349 pts)
- F: (0-299 pts)

**WITH THE FOLLOWING EXCEPTIONS:**

1. The total points of the five exams must be at least 180 points (60 %) to pass the course.
2. The total points of the Lab Assignments must be at least 60 points (60 %) to pass the course.
3. The total points of the quizzes must be at least 60 points (60 %) to pass the course.

**NOTE:** Any questions concerning a grade on a lab or exam must be handled within five days of the time the item was returned.

**Attendance Bonus:**

Absences	0	1	2
Bonus points	15	10	5

**Important dates:**

September 6	Exam 1	October 31	Last day to drop with a "W"
September 9	Last day to drop without a grade	November 15	Exam 4
September 27	Exam 2	December 4	Last Day of Class
October 25	Exam 3	December 11	Final Exam (3:30-5:30)

**Reasonable accommodation for students with disabilities:** If you have a disability that may require assistance or accommodation, or you have questions related to any accommodations for testing, note takers, readers, etc., please speak with the instructor as soon as possible. Any student interested in reasonable accommodations can consult the *Disability & Access Center (DAC)* website [www.mtsu.edu/dac](http://www.mtsu.edu/dac). Students may also contact the *DAC* for assistance at 615-898-2783 or [dacemail@mtsu.edu](mailto:dacemail@mtsu.edu). Middle Tennessee State University is committed to campus access in accordance with Title II of the Americans with Disabilities Act and Section 504 of the Vocational Rehabilitation Act of 1973.

**FINANCIAL AID NOTICE:**

Do you have a lottery scholarship? To retain the Tennessee Education Lottery Scholarship eligibility, you must earn a cumulative TELS GPA of 2.75 after 24 and 48 attempted hours and a cumulative TELS GPA of 3.0 thereafter. A grade of C, D, F, FA, or I in this class may negatively impact TELS eligibility.

If you drop this class, withdraw, or if you stop attending this class you may lose eligibility for your lottery scholarship, and you will not be able to regain eligibility at a later time.

For additional Lottery rules, please refer to your Lottery Statement of Understanding form (<http://www.mtsu.edu/financial-aid/forms/LOTFEV.pdf>) or contact your MT One Stop Enrollment Coordinator (<http://www.mtsu.edu/one-stop/counselor.php>).

**PROBLEMS, COMPLAINTS, OR SUGGESTIONS:**

If you are having problems with the course, or have a complaint or suggestion you would like to voice, please bring this to the attention of the course instructor as soon as possible.

**FREE TUTORING!**

Learn how to study, get help with understanding difficult course material, receive better test grades, or simply improve your grade point average. Take advantage of our FREE tutoring that is available to you as an MTSU student. Tutoring is available in study skills and learning strategies, and over 180 courses including biology, history, computer information systems, physics, math, psychology, chemistry, economics, recording industry, and many more. The central location for tutoring is the Tutoring Spot, located in Walker Library. Tutoring is also conducted at various other campus sites. For available tutoring opportunities, visit <http://mtsu.edu/studentsuccess/tutoring.php#on>. For questions, call the Tutoring Spot at 615-904-8014.

**TENTATIVE Schedule (CSCI/MATH 3180)**

<u>Dates</u>	<u>Topic</u>	<u>Labs/Exams</u>
8/28, 30	Visual Studio 2015 Maple Chapter 1: Mathematical Preliminaries	LAB1
9/4, 6	Chapter 4: Numerical Differentiation	EXAM1
9/11,13	Chapter 7: Initial Value Problems	LAB2
9/18, 20	Chapter 7: Initial Value Problems	LAB3
9/25, 27	Chapter 2: Linear Systems	EXAM2
10/2, 10/4	Chapter 2: Linear Systems	LAB4
10/9, 11	Chapter 4: Interpolation	LAB5
10/18	Chapter 4: Interpolation	LAB6
10/23, 25	Chapter 6: Spline Functions	EXAM3
10/30, 11/1	Chapter 6: Spline Functions	LAB7
11/6, 8	Chapter 3: Nonlinear Equations	LAB8
11/13, 15	Chapter 3: Nonlinear Equations	EXAM4
11/20, 22	Chapter 5: Numerical Integration	LAB9
11/27	Chapter 5: Numerical Integration	LAB10
12/4	Review	
12/11	Final Exam (3:30-5:30pm)	FINAL EXAM

# Honors Contract Proposal

## Evolutionary Runge-Kutta<sup>1</sup>

Jackson L. Cole

*Department of Physics and Astronomy, Middle Tennessee State University*

Fall 2018

<sup>1</sup>[https://github.com/jacksonlanecole/rk\\_genetic](https://github.com/jacksonlanecole/rk_genetic)

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Important Note . . . . .	4
<b>2</b>	<b>Purpose</b>	<b>5</b>
<b>3</b>	<b>Goals</b>	<b>5</b>
3.1	General purpose explicit Runge-Kutta algorithm . . . . .	5
3.2	Evolutionary algorithm to optimize $s$ -stage Butcher Tableau using DEAP[1] . . . . .	6
3.3	Symplectic integrator for comparison . . . . .	6
<b>4</b>	<b>Methods</b>	<b>6</b>
<b>5</b>	<b>Progress</b>	<b>7</b>

# List of Figures

1	Full Butcher Tableau for explicit fourth-order Runge-Kutta (RK4) . . . . .	2
2	Butcher Tableau for explicit fourth-order Runge-Kutta (RK4) . . . . .	3
3	Generic Runge-Kutta algorithm . . . . .	5

## 1 Introduction

When analyzing physical systems, there are often cases when analytical solutions to ordinary differential equations describing the systems are either exceedingly difficult to find or simply do not exist. In these cases, it is advantageous to use numerical methods to iteratively solve the systems. While there are several methods for solving ordinary differential equations, the “workhorse” of initial value problem solvers is the fourth-order Runge-Kutta (RK4). [3]

Essentially, the RK4 method involves computing derivatives at four different points in the parameter space, and updating the solution at the iterated parameter to take into account the “weighted average” of the computed derivatives. In other words, if we define a function  $f(t, x)$ , we can say that the derivatives are calculated as shown in equation (1).

$$\begin{aligned}
 k_1 &= f(t_i, x_i) \\
 k_2 &= f\left(t_i + \frac{1}{2}h, x_i + \frac{1}{2}k_1\right) \\
 k_3 &= f\left(t_i + \frac{1}{2}h, x_i + \frac{1}{2}k_2\right) \\
 k_4 &= f(t_i + h, x_i + k_3)
 \end{aligned} \tag{1}$$

The solution is updated

$$x = x_i + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4) h \quad (2)$$

and the time is naturally updated by

$$t = t_i + h. \quad (3)$$

However, while the choice of the bolded leading coefficient seems to come from a desire to appropriately average the derivatives, Holmes [3] points out that this constant is largely arbitrary, but is as standard chosen to be  $\frac{1}{6}$  to match the results of other methods. This derivation is not of importance in the case of this project.

To be more explicit in the definition of the scheme of the RK4, we could express it a bit differently, as shown in equation (4).

$$\begin{aligned} k_1 &= f(t_i, x_i, (0)k_1, (0)k_2, (0)k_3, (0)k_4) \\ k_2 &= f(t_i, x_i, (\frac{1}{2})k_1, (0)k_2, (0)k_3, (0)k_4) \\ k_3 &= f(t_i, x_i, (\frac{1}{2})k_1, (\frac{1}{2})k_2, (0)k_3, (0)k_4) \\ k_4 &= f(t_i, x_i, (0)k_1, (0)k_2, (1)k_3, (0)k_4) \\ x &= x_i + [(\frac{1}{6})k_1 + (\frac{2}{6})k_2 + (\frac{2}{6})k_3 + (\frac{1}{6})k_4]h \end{aligned} \quad (4)$$

If we extract the coefficients of the step sizes, the internal coefficients, and the weights of the derivatives in the bottom line of equation (4), with some abstraction, we obtain what is known as the Butcher Tableau describing the scheme of the RK4. This is shown in figure 1. Because this is clearly just a lower triangular matrix, we can more simply express it as in figure 2. It should be noted that explicit Runge-Kutta routines have the requirement that its associated Butcher Tableau is strictly lower diagonal, as this removes the possibility of needing to solve implicit equations while calculating successive derivatives. Further, a sample C++ snippet has been given for running a an RK4 routine as shown below in listing 1.

0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
1	0	0	1	0
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

Figure 1: Full Butcher Tableau for explicit fourth-order Runge-Kutta (RK4). This form comes from an abstraction of the full RK4 scheme.



0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

Figure 2: Butcher Tableau for explicit fourth-order Runge-Kutta (RK4). This form of the tableau is found in many texts, so I will omit a reference here.

Listing 1: Fourth-order Runge-Kutta (RK4)

---

```

1 // C++
2 for (int i = 0; i < nSteps; i++) {
3     k1 = f(t, x);
4     k2 = f(t + (1/2.)*h, x + (1/2.)*k1);
5     k3 = f(t + (1/2.)*h, x + (1/2.)*k2);
6     k4 = f(t + h, x + k3);
7     x = x + h*(k1 + 2*k2 + 2*k3 + k4)/6;
8     t = t + h;
9 }

```

---

Although the Runge-Kutta method is much more accurate than lower order methods or methods like Euler, an issue that arises in the study of physical systems is that we are often more aware of the conditions of our system than is our integrator. When we look at systems in celestial mechanics, like the three-body problem that has no analytical solution, we should acknowledge that under the assumption that the system is closed, the total mechanical energy should remain constant. Operating with this knowledge, the validity of the solution provided by the RK4 must be called into question as the total energy is *not* conserved over long time integrations. [3]

When modeling physical systems, a common method of preserving conserved quantities is to use a class of integrators known as *symplectic* integrators. While the scope of this project does not necessarily extend to symplectic integrators outside of the possible implementation of one for comparison of the validity of this method, they should receive some discussion. In general, many physical systems can be described in the Hamiltonian formulation of mechanics; this is a far more flexible approach to mechanics than is Newtonian or even Lagrangian formulation, especially in terms of the generalization of coordinates and the conservation of naturally conserved quantities. **The following derivation of the Hamiltonian is based on that found in Taylor [4].**

In general, we can arrive at what is known as the Lagrangian for a particular system, which can be

defined by

$$\mathcal{L} = T - U \quad (5)$$

where  $T$  is the kinetic energy expression for the system, and  $U$  is the potential energy expression. We also assume that the Lagrangian is a function of  $n$  generalized coordinates

$$\mathcal{L} = (q_1, q_2, q_3, \dots, q_n, \dot{q}_1, \dot{q}_2, \dot{q}_3, \dots, \dot{q}_n) = T - U \quad (6)$$

which means that in terms of the generalized coordinates, Lagrange's equations of motion can be written as second order differential equations as

$$\frac{\partial \mathcal{L}}{\partial q_i} = \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_i} \quad \text{where } i = [1, \dots, n] \quad (7)$$

However, using the Hamiltonian approach allows us to more easily express the equations of motion as two first order differential equations. We can define the conjugate momentum (momenta in multiple dimensions) is expressed as

$$p_i = \frac{\partial \mathcal{L}}{\partial \dot{q}_i} \quad (8)$$

Now with this defined, the Hamiltonian is defined as

$$\mathcal{H} = \sum_i^n p_i \dot{q}_i - \mathcal{L}. \quad (9)$$

Without derivation, the equations of motion can be expressed as

$$\dot{q} = \frac{\partial \mathcal{H}}{\partial p} \quad \dot{p} = -\frac{\partial \mathcal{H}}{\partial q}. \quad (10)$$

Using these equations of motion, integration schemes can be defined that iteratively solve the system on a symplectic manifold on which the system is defined in the first place, which means that quantities that are conserved in the physical system will also be conserved in the iterative solution.

## 1.1 Important Note

Symplecticity and geometrical integration are two topics about which I know *very little*, but I feel as though I understand the idea that the solutions to Hamilton's equations lie on a symplectic manifold, and that symplectic integrators iteratively solve the system on a symplectic manifold. The underlying math is mostly enigmatic to me, but I feel as though implementing a standard symplectic integrator would be doable as

long as I can find an expression for the system's Hamiltonian.

## 2 Purpose

While symplectic integrators are the go-to choice for numerically solving systems in which a Hamiltonian can be defined, Runge-Kutta methods are still widely used for solving physical systems as well. Further, the Runge-Kutta method should be able to be optimized such that conserved quantities in *any* system would be respected. To be more general, in any system in which a cost function can be defined, the hope is that a method can be developed that improves the results on a system specific basis. The primary directive of this project is to explore the use of evolutionary algorithms to evolve the Butcher Tableau describing an  $s$ -stage Runge-Kutta scheme to improve explicit Runge-Kutta methods in systems in which a cost function can be defined. In celestial mechanics, for example, the obvious examples are gravitationally bound systems in which the energy of the system must be conserved to ensure the stability of orbits.

## 3 Goals

### 3.1 General purpose explicit Runge-Kutta algorithm

In order to have the ability to optimize the weights, nodes, and internal coefficients of the tableau describing a Runge-Kutta scheme, a general purpose Runge-Kutta algorithm must first be developed. Without specific details about processes in the Runge-Kutta algorithm, the general idea can be visualized in figure 3.

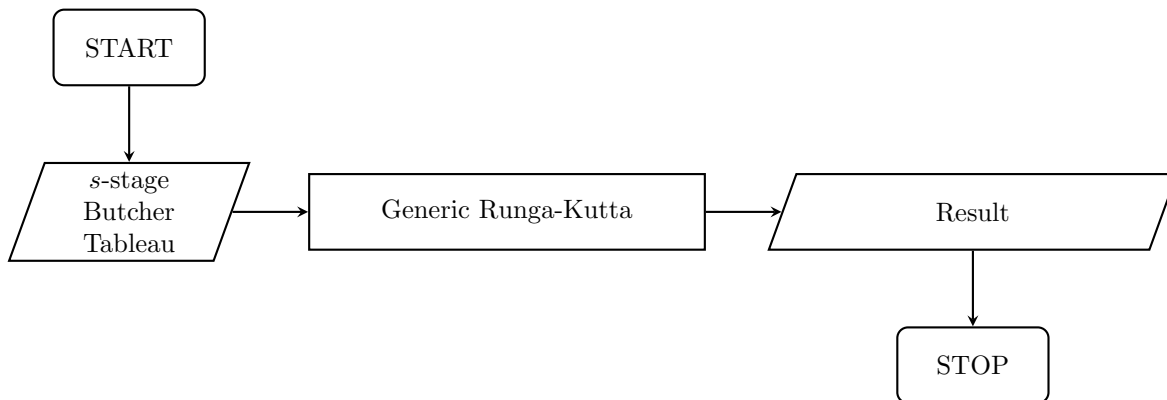


Figure 3: Generic Runge-Kutta algorithm

If this routine works as intended, then if given the Butcher Tableau for the classic fourth-order Runge-Kutta (RK4) as in figure 2, the integrator will integrate using a classic RK4 routine. Although there are

also implicit Runge-Kutta methods that work with non-lower-triangular Butcher Tableaus, I will not be implementing an implicit Runge-Kutta routine in this project.

### 3.2 Evolutionary algorithm to optimize $s$ -stage Butcher Tableau using DEAP[1]

The evolution of the Butcher Tableau will be done in Python using the DEAP (Distributed Evolutionary Algorithms in Python) framework. This is a framework developed “for rapid prototyping and testing of ideas,” and to “make algorithms explicit and data structures transparent.”[1] This framework provides a method to create individuals using user-defined data types, which is helpful for quickly generating a solution for evolving the tableaus.

Further, a common problem with evolutionary techniques and techniques in machine learning is that while they work exceptionally well in many cases, they can quickly become an intellectually unsatisfying black-box. The hope is that using a dedicated evolutionary algorithm framework will allow some degree of transparency into the actual optimization process.

The repository for the DEAP project can be found at <https://github.com/DEAP/deap>.

### 3.3 Symplectic integrator for comparison

While I am largely unfamiliar with workhorse symplectic integrators, I found several that were often mentioned by [2]. The implementation of one of these integrators will not necessarily be a focal point of this project, but having one implemented for comparison to this method will aid in confirming that this method works (or fails miserably).

## 4 Methods

Both the generic Runge-Kutta integrator and the symplectic integrator will be written in C++, and the genetic routine for optimizing the Butcher Tableau will be written in Python. C++ is undoubtedly the far better choice for performing potentially computationally expensive tasks, and writing the integrators in C++ opens up the possibility of further optimization through parallelization of the code. Python can handle multiprocessing through the `multiprocessing` package, but can only execute one thread at a time due to the Global Interpreter Lock. Further, despite its ease of use, Python is simply too slow to perform these types of iterative computations efficiently.

That being said, the natural choice for writing a evolutionary algorithm quickly is Python. As was mentioned, DEAP [1] will be used for development of the evolutionary algorithm for this project.

Clearly, this project must then be written in a way that allows a C++ RK implementation to be called from Python. There are several libraries that can aid in this process (Boost, ctypes). That being said, in many cases it seems that it can be difficult to expose C++ data structures to Python (while I have no direct source for this, I have heard this mentioned many times over the years), so the C++ will likely have to be written in a largely procedural way, such that it may only contain the RK method itself. Any higher level structures could then be written in Python.

At this point, further describing the methods would likely not be productive, as I still need to try out several ideas.

## 5 Progress

As of right now, a generic Runge-Kutta algorithm is nearly complete. It accepts a Butcher Tableau abstracted to a multidimensional vector of the form `std::vector< std::vector<double> >`, a function of the form  $f = f(t, x)$ , a step size, upper and lower bounds of the time interval, and an initial value of the function at lower bound of the interval.

The program runs, but returns an incorrect solution for the ODE. Once I am able to remedy the issue with the Runge-Kutta routine, I can begin working on implementing a symplectic integrator for comparison and then wrapping the generic Runge-Kutta routine for interfacing with Python.

The repository for this project can be found at [https://github.com/jacksonlanecole/rk\\_genetic](https://github.com/jacksonlanecole/rk_genetic), and can be run using `./run` after cloning the repository. This command will build the project and run it.

## References

- [1] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171–2175, jul 2012.
- [2] E. Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations = Ji he shu zhi ji fen*. Shi jie tu shu chu ban gong si, 2017.
- [3] Mark H. Holmes. *Introduction to Scientific Computing and Data Analysis*. Springer International Publishing, 2018.
- [4] John R. Taylor. *Classical mechanics*. Univ. Science Books, 2005.