Sensitivity to Numerical Integration Scheme in Calculation of Transport Barriers

Arne Magnus Tveita Løken

Department of Physics, Norwegian University of Science and Technology, N-7491 Trondheim, Norway

November 20, 2017

Abstract

Sammendrag

Preface

Table of Contents

List of Figures

List of Tables

Notation

1 Theory

1.1 SOLVING SYSTEMS OF ORDINARY DIFFERENTIAL EQUATIONS

In physics, like other sciences, modeling a system often equates to solving an initial value problem. An initial value problem can be described in terms of a differential equation of the form

$$\dot{x}(t) = f\big(t, x(t)\big), \quad x(t_0) = x_0, \tag{1.1}$$

where x is an unknown function (scalar or vector) of time t. The function f is defined on an open subset Ω of $\mathbb{R} \times \mathbb{R}^n$. The initial condition (t_0, x_0) is a point in the domain of f, i.e., $(t_0, x_0) \in \Omega$. In higher dimensions (i.e., $n > 1$) the differential equation (1.1) is replaced by a family of equations

$$\begin{gathered} \dot{x}_i(t) = f_i\big(t, x_1(t), x_2(t), \ldots, x_n(t)\big), \quad x_i(t_0) = x_{i,0}, \quad i = 1, \ldots, n \\ \mathbf{x}(t) = \big(x_1(t), x_2(t), \ldots, x_n(t)\big) \end{gathered} \tag{1.2}$$

The system if nonlinear if the function f in equation (1.1), alternatively, if at least one of the functions f_i in equation (1.2), is nonlinear in one or more of its arguments.

1.1.1 *The Runge-Kutta family of numerical methods*

For nonlinear systems, analytical solutions frequently do not exist. Thus, such systems are often analyzed by means of numerical methods. In numerical analysis, the Runge-Kutta family of methods are a frequently used collection of implicit and explicit iterative methods, used in temporal discretization in order to obtain numerical approximations of the *true* solutions of systems like (1.1). The German mathematicians C. Runge and M. W. Kutta developed the first of the family's methods at the turn of the twentieth century (Hairer, Nørsett, and Wanner 1993, p.134 in the 2008 printing). The general scheme of what is now known as a Runge-Kutta method is as follows:

Definition 1. Let s be an integer and $a_{1,1}, a_{1,2}, \ldots, a_{1,s}, a_{2,1}, a_{2,2}, \ldots, a_{2,s}, \ldots, a_{s,1}, a_{s,2}, \ldots, a_{s,s}$, $b_1, b_2, \ldots, b_s$ and $c_1, c_2, \ldots, c_s$ be real coefficients. Let h be the numerical step length used in the temporal discretization. Then, the method

$$\begin{aligned} k_i &= f\left(t_n + c_i h, x_n + h \sum_{j=1}^{s} a_{i,j} k_j\right), \quad i = 1, \ldots, s \\ x_{n+1} &= x_n + h \sum_{i=1}^{s} b_i k_i \end{aligned} \tag{1.3}$$

is called an *s-stage Runge-Kutta method* for the system (1.1).

The main reason to include multiple stages s in a Runge-Kutta method, cf. definition 1, is to improve the numerical accuracy of the computed solutions. Hairer, Nørsett, and Wanner (1993, p.2 in the 2010 printing) define the *order* of a Runge-Kutta method as follows:

Definition 2. A Runge-Kutta method (1.3) is said to be of *order p* if, for sufficiently smooth systems (1.1),

$$\|x_{n+1} - x(t_{n+1})\| \leq Kh^{p+1} \tag{1.4}$$

i.e., if the Taylor series for the exact solution $x(t_{n+1})$ and the numerical solution x_{n+1} coincide up to (and including) the term h^p.

It is easy to show that if the local error of a Runge-Kutta method is of order p, cf. definition 2, the global error, i.e., the total accumulated error resulting of applying the algorithm a number of times, is expected to scale as h^p. Showing this is left as an exercise for the interested reader.

In definition 1, the matrix $(a_{i,j})$ is commonly called the *Runge-Kutta matrix*, while b_i and c_i are known as the *weights* and *nodes*, respectively. Since the 1960s, it has been customary to represent Runge-Kutta methods (1.3) symbolically, by means of mnemonic devices known as Butcher tableaus (Hairer, Nørsett, and Wanner 1993, p.134 in the 2008 printing). The Butcher tableau for a general s-stage Runge-Kutta method, introduced in definition 1, is presented in table 1.1.

Table 1.1: Butcher tableau representation of a general s-stage Runge-Kutta method.

c_1	$a_{1,1}$	$a_{1,2}$	$\dots$	$a_{1,s}$
c_2	$a_{2,1}$	$a_{2,2}$	$\dots$	$a_{2,s}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_s	$a_{s,1}$	$a_{s,2}$	$\dots$	$a_{s,s}$
	b_1	b_2	$\dots$	b_s

For explicit Runge-Kutta methods, the Runge-Kutta matrix $(a_{i,j})$ is lower triangular. Similarly, for fully implicit Runge-Kutta methods, the Runge-Kutta matrix is upper triangular. Unlike explicit methods, implicit methods require the solution of a linear system at every time level, making them more computationally demanding than their explicit siblings. The main selling point of implicit methods is that they are more numerically stable than explicit methods. This property means that implicit methods are particularly well-suited for *stiff* systems, i.e., physical systems with highly disparate time scales (Hairer and Wanner 1996, p.2 in the 2010 printing). For such systems, most explicit methods are highly numerically unstable, unless the numerical step size is made exceptionally small, rendering most explicit methods practically useless. For *nonstiff* systems, however, implicit methods behave similarly to their explicit analogues in terms of numerical accuracy and convergence properties.

During the first half of the twentieth century, a substantial amount of research was conducted in order to develop numerically robust, high-order, explicit Runge-Kutta methods. The idea was that using such methods would mean one could resort to larger time increments h without sacrificing precision in the computational solution. However, the number of stages s grows quicker than linearly as a function of the required order p. It has been proven that, for $p \geq 5$, no explicit Runge-Kutta method of order p with $s = p$ stages exists (Hairer, Nørsett, and Wanner 1993, p.173 in the 2008 printing). This is one of the reasons for the attention shift from the latter half of the 1950s and onwards, towards so-called *embedded* Runge-Kutta methods.

The basic idea of embedded Runge-Kutta methods is that they, aside from the numerical approximation x_{n+1}, yield a second approximation $\widehat{x}_{n+1}$. The difference between the two approximations then yields an estimate of the local error of the less precise result, which can be used for automatic step size control. The trick is to construct two independent, explicit Runge-Kutta methods which both use the *same* function evaluations. This results in practically obtaining the two solutions for the price of one, in terms of computational complexity. The Butcher tableau of an embedded, general, explicit Runge-Kutta method is illustrated in table 1.2.

Table 1.2: Butcher tableau representation of general, embedded, explicit Runge-Kutta methods.

0					
c_2	$a_{2,1}$				
c_3	$a_{3,1}$	$a_{3,2}$			
$\vdots$	$\vdots$	$\vdots$	$\ddots$		
c_s	$a_{s,1}$	$a_{s,2}$	$\dots$	$a_{s,s-1}$	
	b_1	b_2	$\dots$	b_{s-1}	b_s
	$\widehat{b}_1$	$\widehat{b}_2$	$\dots$	$\widehat{b}_{s-1}$	$\widehat{b}_s$

For embedded methods, the coefficients are tuned such that

$$x_{n+1} = x_n + h \sum_{i=1}^{s} b_i k_i \tag{1.5a}$$

is of order p, and

$$\widehat{x}_{n+1} = x_n + h \sum_{i=1}^{s} \widehat{b}_i k_i \tag{1.5b}$$

is of order $\widehat{p}$, typically with $\widehat{p} = p \pm 1$. Which of the two approximations x_{n+1} or $\widehat{x}_{n+1}$ varies depending on the embedded method in question. Continuing the integration with the higher order result is commonly referred to as *local extrapolation.* The details on how to implement adaptive stepsize control are not immediately relevant for the topic of this project, but can be found in Hairer, Nørsett, and Wanner (1993, pp.167-168 in the 2008 printing).

1.1.2 *The Runge-Kutta methods under consideration*

Naturally, an abundance of Runge-Kutta methods exist. Many of them are fine-tuned for specific constraints, such as problems of varying degrees of stiffness. It is neither possible nor meaningful to investigate them all in the context of general flow dynamics. For this reason, I consider two classes of explicit Runge-Kutta methods, namely singlestep and adaptive stepsize methods. From both classes, I include four different general-purpose ODE solvers of varying order.

Singlestep methods

The singlestep methods under consideration are the classical, explicit Runge-Kutta methods of orders one through to four, i.e., the *Euler*, *Heun*, *Kutta* and *classical Runge-Kutta* methods. The Euler method is 1st order accurate, and requires a single function evaluation of the right hand side of the ordinary differential equation (1.1) or (1.2) at each time step. Its Butcher tableau representation can be found in table 1.3. It is the simplest explicit method for numerical integration of ordinary differential equations. The Euler method is often used as a basis to construct more complex methods, such as the Heun method, which is also known as the *improved Euler method* or the *explicit trapezoidal rule.* The Heun method is 2nd order accurate, and requires two function evaluations at each time step. Its Butcher tableau representation can be found in table 1.4.

Table 1.3: Butcher tableau representation of the explicit Euler method.

0	
	1

Table 1.4: Butcher tableau representation of the explicit Heun method.

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

The Kutta method is 3rd order accurate, and requires three function evaluations of the right hand side of the ordinary differential equation (1.1) or (1.2) at each time step. Its Butcher tableau representation can be found in table 1.5. The classical Runge-Kutta method is 4th order accurate, and perhaps the most well-known and frequently used of the four singlestep schemes discussed in this project. One reason for its popularity is that it is exceptionally stable numerically (i.e., of the aforementioned singlestep methods, the classical Runge-Kutta method has the largest numerical stability domain). Another is that, as mentioned previously, for $p \geq 5$, no explicit Runge-Kutta method of order p with $s = p$ stages exist (Hairer, Nørsett, and Wanner 1993, p.173 in the 2008 printing) – which means that the required number of function evaluations grows at a disproportional rate with the required accuracy order. For systems with right hand sides which are computationally costly to evaluate, this means that one frequently is able to obtain the desired numerical accuracy more effectively e.g. by using the classical Runge-Kutta method with a finer step length. The Butcher tableau representation of the classical Runge-Kutta method can be found in table 1.6.

Table 1.5: Butcher tableau representation of the explicit Kutta method.

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

Table 1.6: Butcher tableau representation of the explicit, classical Runge-Kutta method.

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

Adaptive stepsize methods

The adaptive stepsize methods under consideration are the Bogacki-Shampine 3(2) and 5(4) methods, and the Dormand-Prince 5(4) and 8(7) methods. The digit outside of the parentheses indicates the order of the solution which is used to continue the integration, while the digit within the parentheses indicates the order of the interpolant solution. Note that the concept of *order* does not translate directly from singlestep methods, as a direct consequence of the adaptive time step. Generally, lower order methods are more suitable than higher order methods for cases where crude approximations of the solution are sufficient. Bogacki and Shampine argue that their methods outperform other methods of the same order (Bogacki and Shampine 1989; Bogacki and Shampine 1996), a notion which, for the 5(4) method, is supported by Hairer, Nørsett, and Wanner (1993, p.194 in the 2008 printing).

Butcher tableau representations of the aforementioned adaptive stepsize methods can be found in tables 1.7–1.10, the latter of which has been typeset in landscape orientation for the reader's convenience. Three of the methods, namely the Bogacki-Shampine 3(2) and 5(4) methods, in addition to the Dormand-Prince 5(4) method, possess the so-called *First Same As Last* property. This means that the last function evaluation of an accepted step is exactly the same as the first function evaluation of the next step. This is readily apparent from their Butcher tableaus, where the *b* coefficients correspond exactly with the last row the Runge-Kutta matrix. The *First Same As Last* property reduces the computational cost of a successive step.

Table 1.7: Butcher tableau representation of the Bogacki- Shampine 3(2) embedded Runge-Kutta method. The b coefficients correspond to a 3$^{\text{rd}}$ order accurate solution used to continue the integration. The $\widehat{b}$ coefficients correspond to a 2$^{\text{nd}}$ order accurate interpolant, which can be used to estimate the error of the numerical approximation, and to dynamically adjust the time step. The *First Same As Last* property is apparent from the fact that the b coefficients correspond exactly to the last row of coefficients in the Runge-Kutta matrix. For reference, see Bogacki and Shampine (1989).

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{3}{4}$	0	$\frac{3}{4}$		
1	$\frac{2}{9}$	$\frac{1}{3}$	$\frac{4}{9}$	
	$\frac{2}{9}$	$\frac{1}{3}$	$\frac{4}{9}$	
	$\frac{7}{24}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{8}$

Table 1.8: Butcher tableau representation of the Bogacki- Shampine 5(4) embedded Runge-Kutta method. The b coefficients correspond to a 5th order accurate solution used to continue the integration. The two rows of $\widehat{b}$ coefficients correspond to two independent 4th order accurate interpolants. They can be used to estimate the error of the numerical approximation, and to dynamically adjust the time step. The fact that two independent interpolants are included is a part of the reason for which the method nearly behaves like a 6th order method (Hairer, Nørsett, and Wanner 1993, p.194 in the 2008 printing). The *First Same As Last* property is apparent from the fact that the b coefficients correspond exactly to the last row of coefficients in the Runge-Kutta matrix. For reference, see Bogacki and Shampine (1996).

0								
$\frac{1}{6}$	$\frac{1}{6}$							
$\frac{2}{9}$	0	$\frac{2}{27}$	$\frac{4}{27}$					
$\frac{3}{7}$	$\frac{183}{1372}$	$\frac{-162}{343}$	$\frac{1053}{1372}$					
$\frac{2}{3}$	$\frac{68}{297}$	$\frac{-4}{11}$	$\frac{42}{143}$	$\frac{1960}{3861}$				
$\frac{3}{4}$	$\frac{597}{22528}$	$\frac{81}{352}$	$\frac{63099}{585728}$	$\frac{58653}{366080}$	$\frac{4617}{20480}$			
1	$\frac{174197}{959244}$	$\frac{-30942}{79937}$	$\frac{8152137}{19744439}$	$\frac{666106}{1039181}$	$\frac{-29421}{29068}$	$\frac{482048}{414219}$		
1	$\frac{587}{8064}$	0	$\frac{4440339}{15491840}$	$\frac{24353}{124800}$	$\frac{387}{44800}$	$\frac{2152}{5985}$	$\frac{7267}{94080}$	
	$\frac{587}{8064}$	0	$\frac{4440339}{15491840}$	$\frac{24353}{124800}$	$\frac{387}{44800}$	$\frac{2152}{5985}$	$\frac{7267}{94080}$	
	$\frac{6059}{80640}$	0	$\frac{8559189}{30983680}$	$\frac{26411}{124800}$	$\frac{-927}{89600}$	$\frac{443}{1197}$	$\frac{7267}{94080}$	
	$\frac{2479}{34992}$	0	$\frac{123}{416}$	$\frac{612941}{3411720}$	$\frac{43}{1440}$	$\frac{2272}{6561}$	$\frac{79937}{1113912}$	$\frac{3293}{556956}$

Table 1.9: Butcher tableau representation of the Dormand-Prince 5(4) embedded Runge-Kutta method. The *b* coefficients correspond to a 5th order accurate solution used to continue the integration. The $\widehat{b}$ coefficients correspond to a 4th order interpolant, which can be used to estimate the error of the numerical approximation, and to dynamically adjust the time step. The *First Same As Last* property is apparent from the fact that the *b* coefficients correspond exactly to the last row of the Runge-Kutta matrix. For reference, see Hairer, Nørsett, and Wanner (1993, p.178 in the 2008 printing).

0							
$\frac{1}{5}$	$\frac{1}{5}$						
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$					
$\frac{4}{5}$	$\frac{44}{45}$	$\frac{-56}{15}$	$\frac{32}{9}$				
$\frac{8}{9}$	$\frac{19372}{6561}$	$\frac{-25360}{2187}$	$\frac{64448}{6561}$	$\frac{-212}{769}$			
1	$\frac{9017}{3168}$	$\frac{-355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$\frac{-5103}{18656}$		
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$\frac{-2187}{6784}$	$\frac{11}{84}$	
	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$\frac{-2187}{6784}$	$\frac{11}{84}$	
	$\frac{5179}{57600}$	0	$\frac{7571}{16695}$	$\frac{393}{640}$	$\frac{-92097}{339200}$	$\frac{187}{2100}$	$\frac{1}{40}$

Table 1.10: Butcher tableau for the Dormand-Prince 8(7) embedded Runge-Kutta method. The b coefficients correspond to an 8th order accurate solution used to continue the integration. The $\widehat{b}$ coefficients correspond to a 7th order interpolant, which can be used to estimate the error of the numerical approximation, and to dynamically adjust the time step. Note that the provided coefficients are rational approximations, accurate to about 24 significant decimal digits – sufficiently well-resolved for most numerical applications. For reference, see Prince and Dormand (1981).

0													
$\frac{1}{18}$	$\frac{1}{18}$												
$\frac{1}{12}$	$\frac{1}{48}$	$\frac{1}{16}$											
$\frac{1}{8}$	$\frac{1}{32}$	0	$\frac{3}{32}$										
$\frac{5}{16}$	$\frac{5}{16}$	0	$\frac{-75}{64}$	$\frac{75}{64}$									
$\frac{3}{8}$	$\frac{3}{80}$	0	0	$\frac{3}{16}$	$\frac{3}{20}$								
$\frac{59}{400}$	$\frac{29443841}{614563906}$	0	0	$\frac{77736358}{692538347}$	$\frac{-28693883}{1125000000}$	$\frac{23124283}{1800000000}$							
$\frac{93}{200}$	$\frac{16016141}{946692911}$	0	0	$\frac{61564180}{158732637}$	$\frac{22789713}{633445777}$	$\frac{545815736}{2771057229}$	$\frac{-180193667}{1043307555}$						
$\frac{5490023248}{9719169821}$	$\frac{39632708}{573591083}$	0	0	$\frac{-433636366}{683701615}$	$\frac{-421739975}{2616292301}$	$\frac{100302831}{723423059}$	$\frac{790204164}{839813087}$	$\frac{800635310}{3783071287}$					
$\frac{13}{20}$	$\frac{246121993}{1340847787}$	0	0	$\frac{-37695042795}{15268766246}$	$\frac{-309121744}{1061227803}$	$\frac{-12992083}{490766935}$	$\frac{6005943493}{2108947689}$	$\frac{393006217}{1396673457}$	$\frac{123872331}{1001029789}$				
$\frac{1201146811}{1299019798}$	$\frac{-1028468189}{846180014}$	0	0	$\frac{8478235783}{508512851}$	$\frac{1311729495}{1432422823}$	$\frac{-10304129995}{1701304382}$	$\frac{-48777925059}{3047939560}$	$\frac{15336726248}{1032824649}$	$\frac{-45442868181}{339846796}$	$\frac{3065993473}{5917172653}$			
1	$\frac{185892177}{718116043}$	0	0	$\frac{-3185094517}{667107341}$	$\frac{-477755414}{1098053517}$	$\frac{-703635378}{230739211}$	$\frac{5731566787}{1027545527}$	$\frac{5232866602}{850066563}$	$\frac{-4093664535}{808688257}$	$\frac{3962137247}{1805957418}$	$\frac{65686358}{487910083}$		
1	$\frac{403863854}{491063109}$	0	0	$\frac{-5068492393}{434740067}$	$\frac{-411421997}{543043805}$	$\frac{652783627}{914296604}$	$\frac{11173962825}{925320556}$	$\frac{-13158990841}{6184727034}$	$\frac{3936647629}{1978049680}$	$\frac{-160528059}{685178525}$	$\frac{248638103}{1413531060}$	0	
	$\frac{1400451}{335480064}$	0	0	0	0	$\frac{-59238493}{1068277825}$	$\frac{181606767}{758867731}$	$\frac{56129285}{797845732}$	$\frac{-1041891430}{1371343529}$	$\frac{760417239}{1151165299}$	$\frac{118820643}{751138087}$	$\frac{-528747749}{2220607170}$	$\frac{1}{4}$
	$\frac{13451932}{455176623}$	0	0	0	0	$\frac{-808719486}{976000145}$	$\frac{1757004468}{5645159321}$	$\frac{656045339}{265891186}$	$\frac{-3867574721}{1518517206}$	$\frac{465885868}{322736535}$	$\frac{53011238}{667516719}$	$\frac{2}{45}$	0

1.2 SETUP

We consider flow in two-dimensional dynamical systems of the form

$$\dot{\mathbf{x}} = \mathbf{v}(t, \mathbf{x}), \quad \mathbf{x} \in \mathcal{U}, \quad t \in [t_0, t_1], \tag{1.6}$$

i.e., systems defined for the finite time interval $[t_0, t_1]$, on an open, bounded subset $\mathcal{U}$ of $\mathbb{R}^2$. In addition, the velocity field $\mathbf{v}$ is assumed to be smooth in its arguments. Depending on the exact nature of the velocity field $\mathbf{v}$, analytical particle trajectories, i.e., solutions of system (**??**), may or may not be computed.

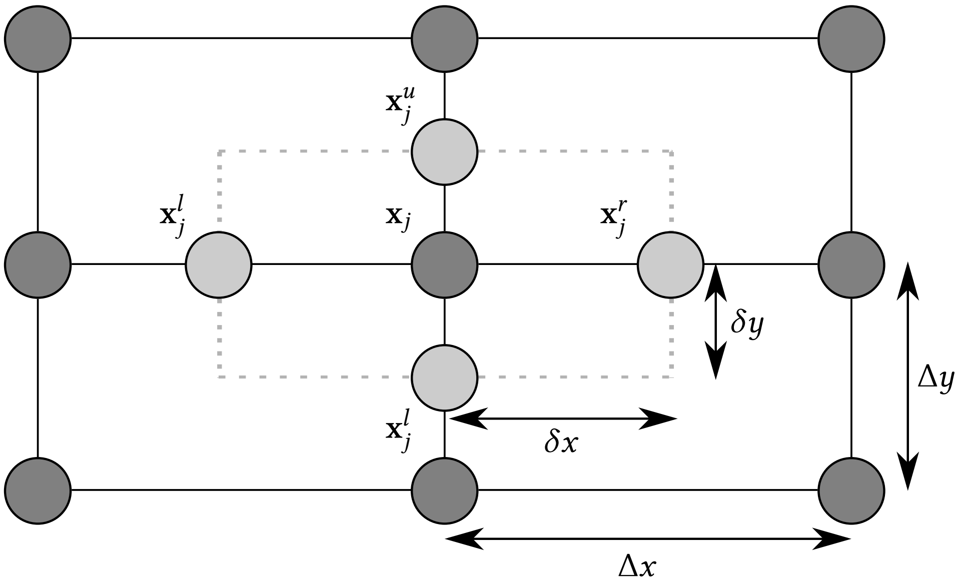

Figure 1.1: Dawg

$$\boldsymbol{A}\mathbf{x} = \mathbf{b} \tag{1.7}$$

2 Introduction

Only interested in hyperbolic LCSs as they are the only ones relevant for transport barriers.

2.1 COMPLEX SYSTEMS –> NEED SHORTCUTS

2.2 INTUITIVELY, WHAT IS AN LCS?

2.3 LCS DEFINITION

2.4 DIFFERENT TYPES OF LCSS

2.5 HYPERBOLIC LCSS

–> Connect to application

3 Theory

3.1 SOLVING ODE SYSTEMS

—> General ODE systems —> Numerical integrators dump –> Interpolation necessary for discrete systems

3.2 FLOWMAPS

—> Introduce system and limitations –> Introduce the concept of a flow map

3.3 LCS DEFINITION

–> Different kinds of LCSs (hyperbolic, elliptic and parabolic, cf. LCS tool) –> More mathematical definitions? Ask Thör

3.4 FTLE AS LCS PREDICTOR

–> Prone to false positives and negatives –> Definition somewhat arbitrary (what is a ridge)? –> Strogatz' motivation as a simple explanation of why we consider it at all?

3.5 IDENTIFY HYPERBOLIC LCS FROM VARIATIONAL THEORY

–> Mathematically involved.

4 Tool!

4.1 ADVEKSJON

–> Si noe om system, glatte vektorfelt/hastighetsfelt –> Integrasjonsteknikker

4.2 CG TENSORS

–> Auxiliary grid –> Extended grids i fire retninger –> Beregn CG tensors –> Centered differencing, consistently for all main particles –> Har med gitterpunkter på utsiden av hoveddomenet for å inkludere diskontinuitet i oppførsel i hastighetsfeltet

4.3 EIGENVALUES/EIGENVECTORS

–> Auxiliary grid –> Laplacian, extended grid layer 2 for centered differencing

4.4 IDENTIFY AB DOMAIN

–> Klargjør måten vi tolket Laplacian på

4.5 COMPUTE STRAINLINES

–> Define G0 along vertical and horizontal lin –> Avoid redundant computations of trajectories –> Integrate forwards and backwards —-> (Notice that strainlines "fall out" of AB domain, likely due to num. error) –> Special linear interpolation with local direction correction –> Higher order spline interpolations are inappropriate because of oriental –> discontinuities (in case of vectors) and great variance (in case of evals) –> Linear spline interpolation without orientation fix caused random turns –> at discontinuities. –> Stop criteria –> Alpha scaling introduced by Haller gave unpredictable leaps —–> After linear interp? –> Used just one integrator here, because [...] –> Choice of integration step (needs test!) –> Note that this step is very sensitive to the flow map details, –> components in the strain tensors down to the 10¯15 level. –> LCS results sensitive to continious failure length, needed to increase —–> it in order to replicate results from Haller due to different AB domain

4.6 IDENTIFY INTERSECTIONS

–> Which lines and why (maximize intersections with as few lines as possible) –> Include all intersetions between a strainline and a vert / horz linear

4.7 IDENTIFY NEIGHBORS

–> Neighbor length essential for LCS results

4.8 SELECT LCSS

—> Identify LCS as local maxima of λ_2 which are also long enough –> Needs at least one neighbor other than itself –> **Cut tail of strainlines which exit AB domain and do not return** –> That part is no LCS! –> Parts/sections of strainlines may qualify as LCSs

5 Experiments

–> What did we try and why?

References

Bogacki, P. and Shampine, L. (1989). “A 3(2) Pair of Runge-Kutta Formulas”. In: *Applied Mathematics Letters* 2.4, pp. 321–325. ISSN: 0893-9659.

Bogacki, P. and Shampine, L. (1996). “An Efficient Runge-Kutta (4, 5) Pair”. In: *Computers & Mathematics with Applications* 32.6, pp. 15–28. ISSN: 0898-1221.

Hairer, E., Nørsett, S. P., and Wanner, G. (1993). *Solving Ordinary Differential Equations I: Nonstiff Problems*. 2nd ed. Springer-Verlag Berlin Heidelberg. ISBN: 978-3-540-56670-0.

Hairer, E. and Wanner, G. (1996). *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. 2nd ed. Springer-Verlag Berlin Heidelberg. ISBN: 978-3-642-05221-7.

Prince, P. and Dormand, J. (1981). “High order embedded Runge-Kutta formulae”. In: *Journal of Computational and Applied Mathematics* 7.1, pp. 67–75. ISSN: 0377-0427.

A Haller er en dust

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est.

Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.