UPPSALA
UNIVERSITET

PROJECT REPORT

Project - Stability analysis of finite element methods and finite difference methods for the advection equation

Investigation of stability limits for continuous and discontinuous higher order Galerkin methods and finite difference methods with a central scheme using explicit time stepping

Andrea Lindgren, Mohammed Mosa, Zackeus Zetterberg

Project in Computational Science: Report

February 25, 2021

Institutionen för informationsteknologi

Abstract

The aim of this paper is to investigate the time step restrictions for higher order finite difference methods and (continuous and discontinuous) finite element methods. This is done when solving the advection equation over a one dimensional spatial domain with cyclic boundary conditions. The problem is reformulated into a semidescrete linear system of equations. By using the eigenvalues of this system in relation to the stability region of the used time stepping method such as the classic forth order explicit Runge-Kutta time stepping method, the maximum stable time step are obtained. Then we consider the effect on this when using evenly spaced nodes respectively Legendre-Gauss-Lobatto nodes as the polynomial interpolation points. Also, the impact on the time step when increasing the order of the method and applying a stabilization technique (artificial viscosity method) is investigated. An efficiency number is used in order to compare the time step restrictions for the methods of interest. Using Legendre Gauss-Lobatto nodes instead of evenly spaced nodes gives no specific improvement on the time step restrictions. Increasing the order of the methods as well as applying stabilization techniques result in further restrictions on the time step. Using a finite element method with Gauss Lobatto points together with Gauss Lobatto quadrature gives the least restrictions on the time step. Among the considered methods, the discontinuous finite element method has the most time step restrictions.

Acknowledgements

We would like to express our special thanks to our supervisors Gunilla Kreiss and Benjamin Weber, without whom we would have struggled a lot more than we did.

Contents

1 Introduction **4**
- 1.1 Motivation 4
- 1.2 Background 4
- 1.3 Aim 5

2 Model problem: The advection equation **5**

3 Theory **6**
- 3.1 Stability Analysis 6
- 3.2 Methods of Eigenvalue Computations 7
- 3.3 Stabilization Techniques 7
- 3.4 Spatial Discretization 8
- 3.5 Time Discretization 12
- 3.6 Matrix Fourier Transform 12
- 3.7 The efficiency number 14
- 3.8 Computational Improvements 15

4 Results **16**

5 Discussion **18**
- 5.1 Dependence on polynomial interpolation points 18
- 5.2 Effect when increasing the order of the methods 18
- 5.3 Consequence of applying stabilization 19

6 Conclusion **19**

1 Introduction

1.1 Motivation

When solving initial value problems using time stepping methods, all time steps are performed sequentially. Therefore, when solving a system up to a specified time, a larger step-size reduces the required number of computations. Depending on the system and the time stepping scheme, there may be a maximum time step that can be chosen while still producing stable results. Obtaining accurate knowledge about this limit allows for better performance, which can be of significant importance when solving large PDE systems.

In order to use explicit time stepping effectively we require sharp estimates of the stability limits for the time step. We would also like to develop a better understanding of which properties of the method affects the stability.

This project is concerned with stability analysis of different high order numerical methods for the advection equation on a one-dimensional (1D) periodic spatial domain. The main methods of spatial discretization that are considered are continuous Lagrange finite element method (FEM), finite difference method (FDM) and discontinuous Galerkin finite element method (DG-FEM). The classical forth-order explicit Runge-Kutta method (RK4) as well as the explicit Euler's Forward method are used to discretize the Advection equation in time separately. Artificial viscosity was applied as a stabilization technique only for FEM in order to see the effects on the maximum stable time step. Fourier stability analysis and different methods of eigenvalue computations were instructive to use for understanding the stability regions of the methods. FEM, FDM and DG-FEM have been investigated by computational experiments and mathematical analysis.

1.2 Background

Numerical methods are important when we examine the world around us. Many of the problems and processes can be described by using partial difference equations (PDE) which is a commonly used way to model real life events to some extent. Not all PDEs can be solved analytically and therefore numerical methods are needed to give a good approximation of the solution.

An important property of the PDEs is that the equation has to be well posed. In order to ensure this, we must supply an initial condition as well as proper boundary conditions. Also, the solution has to be stable, which implies that it does not grow in an unbounded fashion. For stability reasons, the product of the time step and the eigenvalues of the matrix differential equation must lie inside the stability region of the used time stepping scheme.

A recent approach to stability analysis for higher order methods is used for the second order wave equation, see [4]. One could notice that for any choice of the polynomial degree, interior interpolation points for a Lagrange basis led to the same time step restriction, regardless of the spacing between interpolation points. In [4], one could even conclude that the DG-SIP (discontinuous Galerkin basis using symmetric interior penalties) basis have a significantly more restricted time step than the Lagrange continuous basis. Compared to what is

done in [4], this project does not consider continuous Galerkin with Hermite basis.

1.3 Aim

This project aims to investigate the restrictions of the maximum stable time step for FD, FEM and DG-FEM. More specifically, we aim at developing a better understanding of the time step limits when the order of the methods increases. Considering FEM, we also want to understand the effects of using different polynomial interpolation points as well as applying stabilization to the problem.

2 Model problem: The advection equation

In this project the following advection equation is considered on a periodic 1D spatial domain.

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0, & x \in [0,1], \quad t > 0, \\ u(x+1,t) = u(x,t), & x \in [0,1), \quad t \geq 0, \\ u(x,0) = u_0(x), & x \in [0,1). \end{cases} \tag{1}$$

The periodic boundaries of this form define a ring-shaped domain and are useful for testing the stability of the discretization without effects from boundary conditions.

The standard convection-reaction-diffusion equation is used to model the time evolution of chemical or biological species in a flowing medium like water or air. The model of evolution can be described by partial differential equations and one can derive them from mass balances. In the numerical analysis literature, different terms "advection" or "convection"are sometimes used. In meteorology, advection is the passive transport by horizontal wind whereas convection refers to vertical transport, which is usually caused by localized vertical heat gradients. A more general advection reaction diffusion model can be written as:

$$\frac{\partial}{\partial t}u(x,t) + \frac{\partial}{\partial x}(a(x,t)u(x,t)) = \frac{\partial}{\partial x}(\alpha(x,t)\frac{\partial}{\partial x}u(x,t)) + f(x,t,u(x,t)), \tag{2}$$

where $u(x,t)$ represents the concentration of a certain species, $a(x,t)$ is a velocity of a flowing medium that the species is carried along, $\alpha(x,t)$ is the diffusion coefficient, $f(x,t,u(x,t))$ describes a local change in the concentration due to sources, sinks and chemical reactions. The advection and diffusion coefficients are assumed to be given linear functions independent of the concentration. The simple constant coefficient case of equation (2) in one-dimensional space is given by:

$$u_t + au_x = 0, \tag{3}$$

where a is a constant, and in our model of equation (1), a is equal to one. $u(x,t) = u(x-at,0)$ satisfies the equation and shows that when we have an initial prescribed initial profile $u(x,0)$, it will be shifted in time with velocity a, without any change of shape. The lines $x - at = const$ in the x-t plane are the characteristics of the advection equation, and along these characteristics the solution $u(x,t)$ is constant [7].

3 Theory

There are many different numerical methods for solving PDEs. The numerical methods that have been investigated in this project are finite difference, finite element and discontinuous finite element.

Finite difference methods, FD, are straightforward techniques to approximate the differentials in the PDEs by forward, backward or central scheme. In this approach, a grid is laid down in space and spatial derivatives are approximated by difference methods, which is further discussed in Section 2.5.

FD methods are simple to use since the discretization of general problems and operators are often intuitive. Also, this method leads to very efficient schemes for many problems. The explicit semidiscrete form gives flexibility in the choice of time stepping methods. The method is sufficiently robust and efficient to be used for a variety of problems. By using a local polynomial approximation of the solution and the flux, of higher degree, the method can be relatively straightforwardly extended to higher order approximations.

However, the simple underlying structure causes some complications that are introduced around boundaries and discontinuous internal layers (e.g., discontinuous material coefficients). This makes the method not suited when dealing with complex geometries, both in terms of general computational domains and internal discontinuities. This also applies to local order and grid size changes that reflect local features of the solution [2].

FEM is a very popular method to discretise various PDEs. It has well-established mathematical theory for various PDEs. It is also very useful in solving PDEs over complex domains such as cars and airplanes. The finite element method is based on rewriting the governing PDE into an equivalent variational problem. The next step is meshing the domain into smaller elements and looking for approximate solutions at the mesh nodes, using in our case a linear (first order) basis function over each element [1]. In order to improve accuracy, one can use higher order basis although this affects the time step restrictions, investigated in this project.

The discontinuous Galerkin finite element method, DG-FEM, has something which is unique compared to FEM and FDM. First, the approximated solution of the DG method is allowed to be discontinuous between elements. As in FEM, DG-FEM assumes that on each element there is a local solution and that there is a locally defined basis function such as Lagrange polynomial in which the solution can be expressed. Then the global solution can be obtained by the local solutions defined on each element DG-FEM allows also different element sizes as FEM and the former method is considered to be a variation of the later. This variation can be explained by adding additional degrees of freedom to the element while maintaining shared nodes along the faces of the elements.

3.1 Stability Analysis

For PDEs, there are three commonly used ways for proving stability: the Fourier Transform analysis, eigenvalue analysis and the energy method. In general, the Fourier analysis applies only to linear constant coefficient problems, while the

energy method can be used for more general problems with variable coefficients and nonlinear terms [1]. When applying the eigenvalue analysis in this project, the product of the time step and the eigenvalues of the matrix differential equation must lie inside the stability region of the used time stepping scheme.

The energy method is usually performed in three steps: the first step is to multiply the homogeneous version of the PDE by the complex conjugate of the solution vector and integrate by parts. The second step is to add the conjugate transpose, and the third step is to employ the correct number of boundary conditions [3]. In this project, we used the energy method to find a stable numerical flux for the DG-FEM, otherwise we concentrated on the Fourier stability analysis.

3.2 Methods of Eigenvalue Computations

There are some direct methods available when computing eigenvalues for certain matrices having special structures as the Fourier analysis. However, the iterative methods such as the power method or the inverse power method are almost always needed. The power method when applied to a matrix A, would return the biggest eigenvalue. Applying the same method to the inverse of A would give the smallest eigenvalue.

Another option is to use the polynomial method to compute the spectrum of A, which is the set of the eigenvalues. Although, this method require finding the characteristic polynomial of the matrix and solving it, which is not appropriate for large sized matrices. This method can be implemented in Matlab by the functions poly() and roots() [5]. Another method, the eig() function computes the eigenvalues as well as the eigenvectors and this is the method that has been used in this project. This method uses QR iteration in order to find the eigenvalues and the eigenvectors of a matrix.

Investigating the eigenvalues allows a better understading of the stability of the numerical method. If the product of the eigenvalues and the time step lies outside the stability region of the time stepping scheme, then we need to use some stabilisation technique.

3.3 Stabilization Techniques

Stabilization techniques can be applied to methods that have an asymmetric stencil or a symmetric stencil in general. One example that has a symmetric stencil is explicit Euler's method which does not include any part of the imaginary axis in the stability region. Another method where stabilization techniques can be applied is the standard FEM. For the more general convection-reaction-diffusion equation, the standard FEM and FDM may have some difficulties in handling boundary layers, which is a quick change that takes place over a small distance around the boundary. This may trigger oscillations throughout the whole computational domain that renders the finite element approximation or the finite difference approximation useless.

Boundary layers does not usually occur for problems with periodic boundaries as the problem (the advection equation) considered in this paper. Although, since we want to examine the effect on the time step when applying stabilization

even though it is not necessary, we need to modify the standard Galerkin finite element method (GFEM). There are different ways of stabilization techniques, for example:

- Isotropic stabilization (artificial diffusion): This stabilization technique adds a diffusion term in order to stabilize the problem. Adding more diffusion, by increasing the diffusion parameter, will reduce the oscillations of the solution. The idea is to add diffusion as little as possible to guarantee stabilization without sacrificing accuracy. One way to reach this is to limit the smallest value of α to the mesh size h. Then having a finer mesh will automatically lead to a decrease of the stabilization. Due to the perturbation of the equation, this method is first order accurate in h. In this project we focus on this method and we investigate what effects this method has on the maximum stable time step for FEM.
- Least Squares stabilization: This technique is a more accurate way to gain stability and furthermore it can be applied to both FEM and DG-FEM. However, this method can not be applied to FDM as artificial diffusion can. The idea of this technique is to use least squares minimization to obtain the normal equations of the Least Squares method. The Galerkin Least Squares (GLS) method is obtained by combining the standard Galerkin and the Least Squares method. This implies that we replace a test function v by $v + \delta Lv$, where δ is a parameter to be chosen suitably (i.e. for maximal accuracy), and L is the differential operator. Doing this, we aim to combine the accuracy of the Galerkin method with the stability of the Least Squares method [6]. It is instructive to mention that most forms of this method need a time derivative in L, which makes this approach not appropriate for numerical methods with semidiscrete forms.

3.4 Spatial Discretization

In this section, we discuss the discretization in space for the methods of interest.

Higher Order Finite Difference with Central Scheme

The finite difference methods considered in this project are of central difference type. These are formed by using Taylor expansions of $u(x \pm jh)$, for $j = 1, 2, ...$; to approximate the spatial derivatives in the differential equation, for example:

$$\begin{cases} j = 1 : \frac{u(x+h)-u(x-h)}{2h} = u^{'}(x) + \mathcal{O}(h^2), \\ j = 2 : \frac{-u(x+2h)+8u(x+h)-8u(x-h)+u(x-2h)}{12h} = u^{'}(x) + \mathcal{O}(h^4). \end{cases} \tag{4}$$

These spatial derivatives can be expressed as a matrix-vector multiplication between a different matrix and the solution vector. For these methods, increasing the order of accuracy reduces the sparsity of the matrix.

Higher Order Galerkin Finite Element Method

This section describes the higher order Galerkin FEM with uses Lagrange basis polynomials.

For equation (2) with constant α and no source term f, we get the following equation and formulate the standard Galerkin finite element method.

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} - \alpha \frac{\partial^2 u}{\partial x^2} = 0, & x \in [0,1], \quad t > 0, \\ u(x+1,t) = u(x,t), & x \in [0,1), \quad t \geq 0, \\ u(x,0) = u_0(x), & x \in [0,1). \end{cases} \tag{5}$$

We define the vector space V:

$$V = \left\{ v : \|v(\cdot,t)\| + \left\| v^{'}(\cdot,t) \right\| < \infty,\ v(x+1,t) = v(x,t) \right\}, \tag{6}$$

where $\|\cdot\| = \|\cdot\|_{L^2(I)}$ denotes the usual L^2 norm, and the norm $\|v\| = \|v(\cdot,t)\|$ is a function of t and not a function of x. We introduce the interval $I = [0,1]$ and $J = (0,T]$, where $T > 0$ is a finite end time for the system. A regular grid of $n+1$ node points $\{x_i\}_{i=0}^{n}$ defining a partition (mesh). Multiplying the PDE in (5) by a test function $v = v(x,t)$ and integrating by parts, we get the following variational formulation: Find $u(x,t)$ such that for every fixed $t \in J, u \in V$ we have:

$$\int_0^1 \frac{\partial u}{\partial t} v dx + \int_0^1 \frac{\partial u}{\partial x} v dx + \alpha \int_0^1 \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} = 0 \quad \forall v \in V,\ t \in J. \tag{7}$$

The mesh of the interval I is divided into sub-intervals $I_i = [x_{i-1}, x_i],\ i = 1, 2, ..., n$ of length $h_i = x_i - x_{i-1}$ and periodic boundary conditions are considered, which means that $I_0 = I_n,\ I_{n+1} = I_1$. Next, let $V_h \subset V$ be the space of piecewise continuous polynomials of degree p on this mesh.

$$V_h = \left\{ v : v \in C^0(I), v|_{I_i} \in P_p(I_i), v(x+1,t) = v(x,t) \right\}, \tag{8}$$

The space discrete counterpart of the variational formulation (the standard Galerkin finite element method) takes the form: Find u_h such that for every fixed $t \in J$, $u_h \in V_h$ we have:

$$\int_0^1 \frac{\partial u_h}{\partial t} v dx + \int_0^1 \frac{\partial u_h}{\partial x} v dx + \alpha \int_0^1 \frac{\partial u_h}{\partial x} \frac{\partial v}{\partial x} dx = 0 \quad \forall v \in V_h,\ t \in J. \tag{9}$$

We make an ansatz:

$$u_h(x,t) = \sum_{j=1}^{n} \sum_{k=0}^{p-1} \xi_j^k(t) \phi_j^k(x), \tag{10}$$

where $\phi_j^k(x)$ is the kth basis function of the jth element, which is constant in time. Due to this, it is possible to discretize the spatial and the temporal domains separately. The coefficients $\xi_j(t)$ are the solutions that we want to find, which are assumed to be continuous real-valued functions on the time interval $[0,\ T]$.

Lagrange polynomials of degree p on a reference interval [0,1] are constructed as done in [4], using a set of local interpolation points $z_0 < z_1 < ... < z_p$. The $p+1$ polynomials $\{\ell_j\}_{j=0}^{p}$ are defined as follows:

$$\ell_j(z) := \prod_{\substack{0 \leq m \leq p \\ m \neq j}} \frac{z - z_m}{z_j - z_m}. \tag{11}$$

The basis functions $\phi_i^k(x)$ can be defined as follows

$$\phi_i^0(x) = \begin{cases} \ell_p\left(\frac{2x-x_{i-1}-x_i}{h_i}\right) & x \in I_i, \\ \ell_0\left(\frac{2x-x_i-x_{i+1}}{h_{i+1}}\right) & x \in I_{i+1}, \\ 0, & \text{otherwise}. \end{cases} \tag{12}$$

$$\phi_i^k(x) = \begin{cases} \ell_k\left(\frac{2x-x_i-x_{i+1}}{h_{i+1}}\right) & x \in I_{i+1}, \\ 0, & \text{otherwise}. \end{cases} \tag{13}$$

Substituting the ansatz in equation (10) from the standard GFEM of (9), we get:

$$M\frac{\partial\vec{\xi}}{\partial t} = -(L+\alpha A)\vec{\xi}, \quad t \in J, \tag{14}$$

where

$$M_{i,j} = \int_0^1 \phi_i(x)\phi_j(x)dx, \tag{15}$$

$$L_{i,j} = \int_0^1 \phi_i(x)\frac{\partial\phi_j(x)}{\partial x}dx, \tag{16}$$

$$A_{i,j} = \int_0^1 \frac{\partial\phi_i(x)}{\partial x}\frac{\partial\phi_j(x)}{\partial x}dx, \tag{17}$$

for $i, j = 1, 2, ..., n \cdot p$. The equation system (14) is a system of $(n-1)p$ coupled linear ordinary differential equations (ODEs) for the $(n-1)p$ coefficients $\xi_j^k(t)$ to be determined.

Discontinuous Galerkin Finite Element Method

The Discontinous Galerkin Finite Element Methods, DG-FEM, uses finite element spaces which consists of piecewise continuous polynomials that are defined on a partition of the computational domain. The finite element formulation is constructed by basis functions using Jacobi polynomials. These Jacobi polynomials are in the form

$$P_n^{(0,0)}(z) = \frac{1}{2^n n!}\frac{d^n}{dz^n}(z^2-1)^n, \tag{18}$$

The basis functions $\phi_i^k(x)$ can be defined as

$$\phi_i^k(x) = \begin{cases} P_k^{(0,0)}\left(\frac{2x-x_i-x_{i+1}}{h_{i+1}}\right) & x \in I_{i+1}, \\ 0, & \text{otherwise}. \end{cases} \tag{19}$$

Because of the ability for the solutions to be discontinuous between element boundaries, additional terms in the weak form are necessary to enforce the proper continuity conditions between the adjacent elements.

The DG-FEM has some useful properties, for example that the mass matrix is local rather than global and thus can be inverted at very little cost, yielding a semi discrete scheme that is explicit. Furthermore, by carefully designing the numerical flux to reflect the underlying dynamics, DG-FEM has flexibility to ensure stability for wavedominated problems.

Consider the problem from equation (5) and the vector space in equation (6). Following the steps in Section 2.5.2 we obtain the weak formulation which is given by equation (7).

The mesh of the interval I is divided into sub intervals $I_i = [x_{i-1}, x_i],\ i = 1, 2, ..., n$ of length $h_i = x_i - x_{i-1}$ and periodic boundary is taken into account by letting $I_0 = I_n$ and $I_{n+1} = I_1$. Then we let $V_h \subset V$ be the space of piecewise discontinuous polynomial elements of degree p on this mesh.

$$V_h = \{v : v|_{I_i} \in P_p(I_i),\ \forall I_i \in I\}\,. \tag{20}$$

Note that the solutions are not forced to be continuous between the elements.

The average operator and the jump operator is needed on the boundary of the intervals in order let the information flow between the intervals. To define these we consider two adjacent elements K^+ and K^- that share the boundary E. The average operator and the jump operator of a function v on the boundary E are defined respectively by:

$$\begin{cases} \langle v(x) \rangle = \frac{v^+(x)+v^-(x)}{2} & x \in E, \\ [v(x)] = v^+(x) - v^-(x) & x \in E, \end{cases} \tag{21}$$

Where $v^\pm = v|_{K^\pm}$. The linear flux is denoted to be $f(u) = u$, although in a more general case when we have a coefficient a in front of the spatial derivative in the advection equation, the linear flux is $f(u) = au$. We make an ansatz:

$$u_h(x,t) = \sum_{i=1}^{n} \sum_{k=0}^{p} \xi_i^k(t)\phi_i^k(x),$$

where n is the number of intervals, p is the degree of the polynomials, ξ is the solution of polynomial k on interval i and ϕ is the corresponding basis function.

We use a numerical flux ζ_i on the boundaries instead of the linear flux, since the solution at the interfaces between elements is multiply defined and we need to choose which solution or combination of solutions is correct.

From this we obtained the local semi discrete formulation:

$$M^i \frac{\partial \xi_i}{\partial t} - S^i \xi_i = \zeta_i \phi_i(x_r^i) + \zeta_i \phi_i(x_l^i), \tag{22}$$

where $M_{rs}^i = (\phi_r^i, \phi_s^i)_{I_i}$ and $S_{rs}^i = (\phi_r^i, \frac{\partial \phi_s^i}{\partial x})_{I_i}$ are the local mass and stiffness matrices.

By integrating by parts, the matrix S^i is transposed and we get the following equation:

$$M^i \frac{\partial \xi_i}{\partial t} + S^i \xi_i = (\xi_i - \zeta_i)\phi_i(x_r^i) - (\xi_i - \zeta_i)\phi_i(x_l^i). \tag{23}$$

3.5 Time Discretization

For linear systems of equations on the form $\frac{\partial u}{\partial t} = f(t, u)$ where the spatial domain is discretized, the time domain can be discretized with for example an explicit Runge Kutta method. In order for the time stepping method to be stable, its stability region must contain $\Delta t \cdot \lambda$ for a given time step, Δt, and all eigenvalues, λ, for the given transformation $f(t, \cdot)$.

3.6 Matrix Fourier Transform

Here we describe the matrix Fourier transform for the FEM and FD schemes.

Finite Element Method

The analysis of time step constraints for FEM for periodic boundary value systems can be formulated via eigenvalue problems for periodic block tridiagonal matrices [4]. That is why we consider the following eigenvalue problem, for the case of $\alpha = 0$.

$$\lambda M \mathbf{x} = -L\mathbf{x}, \tag{24}$$

for all $\lambda \in \mathcal{C}$ being the eigenvalues of the generalized eigenvalue problem, equation (24), and M and L being the $Nq \times Nq$ matrices defined by equation (15) and equation (16) respectively. We multiply with the complex number i in order to get a self adjoint matrix on the right hand side, but since M is already a self-adjoint positive definite matrix we rewrite the system into:

$$i\lambda M \mathbf{x} = -iL\mathbf{x},$$

by denoting $i\lambda = \tilde{\lambda}$ and $\tilde{L} = -iL$ we obtain

$$\tilde{\lambda} M \mathbf{x} = \tilde{L}\mathbf{x}, \tag{25}$$

where both M and $\tilde{L}$ are self adjoint matrices now. Since both M and $\tilde{L}$ are self adjoint they both have purely real eigenvalues. Thus $\tilde{\lambda}$ has to be real valued which tells us that λ is completely imaginary. Therefore, a part of the imaginary axis needs to lie inside any feasible time stepping methods stability region in order to be stable.

The matrices M and $\tilde{L}$ have the following periodic block tridiagonal structure:

$$M = \begin{bmatrix} M_1 & M_2 & & & & & M_2^* \\ M_2^* & M_1 & M_2 & & & & \\ & M_2^* & M_1 & M_2 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & M_2^* & M_1 & M_2 \\ M_2 & & & & & M_2^* & M_1 \end{bmatrix}, \tag{26}$$

$$\tilde{L} = \begin{bmatrix} \tilde{L}_1 & \tilde{L}_2 & & & & & \tilde{L}_2^* \\ \tilde{L}_2^* & \tilde{L}_1 & \tilde{L}_2 & & & & \\ & \tilde{L}_2^* & \tilde{L}_1 & \tilde{L}_2 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \tilde{L}_2^* & \tilde{L}_1 & \tilde{L}_2 \\ \tilde{L}_2 & & & & & \tilde{L}_2^* & \tilde{L}_1 \end{bmatrix}, \tag{27}$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{N-1} \end{bmatrix} \quad \mathbf{x}_0, \ldots, \mathbf{x}_{N-1} \in \mathbb{C}^q, \tag{28}$$

where M_2^* is the conjugate transpose of M_2 and $\tilde{L}_2^*$ is the conjugate transpose of $\tilde{L}_2$. Both M_1 and $\tilde{L}_1$ are self adjoint.

Inserting the matrices from equation (26) and equation (27) into equation (25) as well as imposing periodicity on x with $\mathbf{x}_{-1} = \mathbf{x}_{N-1}, \mathbf{x}_N = \mathbf{x}_0$ gives us $\forall j = 0, 1, \ldots, N-1$ the following equation:

$$\lambda(M_2^* \mathbf{x}_{j-1} + M_1 \mathbf{x}_j + M_2 \mathbf{x}_{j+1}) = \tilde{L}_2^* \mathbf{x}_{j-1} + \tilde{L}_1 \mathbf{x}_j + \tilde{L}_2 \mathbf{x}_{j+1}. \tag{29}$$

We use a discrete Fourier transform of $\mathbf{x}_j$ and we get:

$$\mathbf{x}_j = \sum_{k=0}^{N-1} \omega^{jk} \hat{\mathbf{x}}_k, \quad \omega = e^{\frac{2i\pi}{N}}. \tag{30}$$

Inserting equation (30) into equation (29) get, we get the simplified expression:

$$\sum_{k=0}^{N-1} \Big(\lambda \hat{M}\Big(\frac{k}{N}\Big) - \hat{L}\Big(\frac{k}{N}\Big) \Big) \hat{\mathbf{x}}_k = 0, \tag{31}$$

where $\hat{M}(\theta)$ and $\hat{L}(\theta)$ are self-adjoint matrices of size $q \times q$.

$$\begin{aligned} \hat{M}(\theta) &= (e^{2i\theta\pi} M_2)^* + M_1 + e^{2i\theta\pi} M_2, \\ \hat{L}(\theta) &= (e^{2i\theta\pi} \tilde{L}_2)^* + \tilde{L}_1 + e^{2i\theta\pi} \tilde{L}_2. \end{aligned} \tag{32}$$

Now we want to find the largest eigenvalue $\tilde{\lambda}$ from $M^{-1}\tilde{L}$. To do this we consider the maximum positive eigenvalue which is denoted eig max(). Each eigenvalue, λ, has a complex conjugate, but since we are interested in finding $\tilde{\lambda}$, the eigenvalues will be pairwise positive and negative and only one of them is needed to obtain the upcoming time step restriction.

$$\text{eig max}(M^{-1}\tilde{L}) = \max_{\theta = 0, \frac{1}{N}, \ldots, \frac{N-1}{N}} \Big[\text{eig max}\Big(\hat{M}(\theta)^{-1} \hat{L}(\theta) \Big) \Big]. \tag{33}$$

Regarding time stepping, we could have chosen any time stepping method which includes a part of the imaginary axis. So we let the explicit Runge Kutta 4 be the method of choice. The stability region of this method limits the biggest value on the complex axis to be $\sqrt{8}$ and due to this, we get the following restriction on the time step.

$$0 \leq \Delta t \text{ eig } \max(M^{-1}\tilde{L}) \leq \sqrt{8}. \tag{34}$$

Finite Difference with Central Scheme

For the finite difference case, we have that the mass matrix M is an identity matrix and therefore $M_2 = M_2^* = 0$. The matrix L is the type of differential operator discussed in section 2.5. The $\tilde{L}$ matrix has a similar tridiagonal structure as the Finite Element counterpart, but it is banded and therefor, every row is identical surrounding the main diagonal. This gives us a scalar relation of the eigenvalues.

3.7 The efficiency number

We use an efficiency number in order to compare our methods. For the FD case, we know that a finer grid will increase the restrictions on the time step. Therefore, we use this efficiency number in this case:

$$\tilde{C}_{eff} = CFL = \frac{a\Delta t}{\Delta x}, \tag{35}$$

where a is the advection speed which is 1 in our case.

Regarding the FEM, three cases have been considered:

- Lagrange polynomials with evenly spaced interpolation points
- Lagrange polynomials with interpolation points at Gauss-Lobatto points.
- Discrete Galerkin method with Jacobi polynomials and upwind flux.

For the case of Gauss-Lobatto distributed points we use exact integration of the polynomials in one test case. We also use Guass-Lobatto quadrature in another test case. When using evenly spaced points we use exact integration. For the discontinuous Galerkin method we used Gauss Lobatto quadrature. For all these cases, we use higher order polynomials as we increase the order of the method. This implies more degrees of freedom within each element. Because of this, we use a rescaled efficiency number, as introduced in [4]:

$$\tilde{C}_{eff} = \frac{a\ \Delta t\ N_{d.o.f}}{\ell}, \tag{36}$$

where $N_{d.o.f}$ is the total number of degrees of freedom which is equal to $N \cdot q$ for the finite element method. ℓ is the width of the domain. The number of degrees of freedoms within each element is denoted by q and N is the amount of elements used. For our problem $\ell = 1$ and $a = 1$.

The efficiency number is normalized by a scaling factor C in order to compare the methods. For the FEM with equidistant grid, we want the rescaled $C_{eff} = 1$.

We can compute the maximum eigenvalues for this case by using equation (34) and the efficiency number without rescaling, $\tilde{C}_{eff} = \frac{\delta t}{h}$. Using this, we obtain the restriction that for all N we have $\frac{3\sqrt{7}}{4^2-1} \cdot \tilde{C}_{eff} \leq 1$ for $p = 1$. In order to compare the methods of interest, we scale the efficiency number as well as the CFL number with this factor $\frac{3\sqrt{7}}{4^2-1}$.

3.8 Computational Improvements

Gaussian Quadrature

Gaussian quadrature is a powerful technique for numerical integration and it is one of the spectral methods. Quadrature refers to the use of an algorithm for the numerical calculation of the value of a definite integral in one or more dimensions.

Gaussian quadrature (Gauss-Legendre rule) is based on using wights c_i and nodes x_i that depend on the number on points m, in order to approximate the integral of some function $f(x)$. The formula to be used for the approximation is the following:

$$\int_{-1}^{1} f(x)dx \simeq \sum_{i=1}^{m} c_i f(x_i), \tag{37}$$

which has degree of accuracy $2m - 1$, which means that the above formula is exact for any polynomial $f(x)$ with degree up to $2m - 1$. To integrate $f(x)$ on the interval $[a, b]$, we used simply a change of variable (which is also used for adapting Gauss-Lobatto points depending on the interval being used):

$$\tilde{c_i} = c_i \frac{b-a}{2}, \quad \tilde{x_i} = \frac{(b-a)x_i + (b+a)}{2}, \tag{38}$$

then the formula in equation (37) can be written as:

$$\int_{a}^{b} f(x)dx \simeq \sum_{i=1}^{m} \tilde{c_i} f(\tilde{x_i}). \tag{39}$$

Gauss-Lobatto Points and Quadrature

A variant of the Gaussian quadrature is to include the endpoints of the interval $[-1, 1]$ (or the endpoints of $[a, b]$, in case we use a change of variable as stated in equation (38)), as nodes, which can facilitate the application of boundary conditions in a solution of boundary value problems. The difference between Gaussian and Gauss-Lobatto is that the later has two degrees of freedom less and consequently, slightly lower accuracy than the former, since two nodes have been restricted to coincide with the internal end points [8].

When using Gauss-Lobatto points and quadrature together with Lagrange basis functions, the basis functions begin to approximate an orthogonal basis. This leaves us with a diagonal mass matrix. Which would speed up the explicit time stepping.

4 Results

In the experiments, presented below, the order of accuracy is limited to 11 because the matrices, arising in all methods except DG-FEM are very ill-conditioned. We observed that the mass matrix became ill-conditioned for higher orders than 11, which can be due to numerical errors which have been accumulated. These errors could be reduced by using symbolic Matlab, although it results in a long running time for the computations.

In the figures below, we show the rescaled efficiency number respectively the rescaled reciprocal of the efficiency number towards the order of accuracy. Tables showing the values that are plotted in the following figures, are found in the appendix.

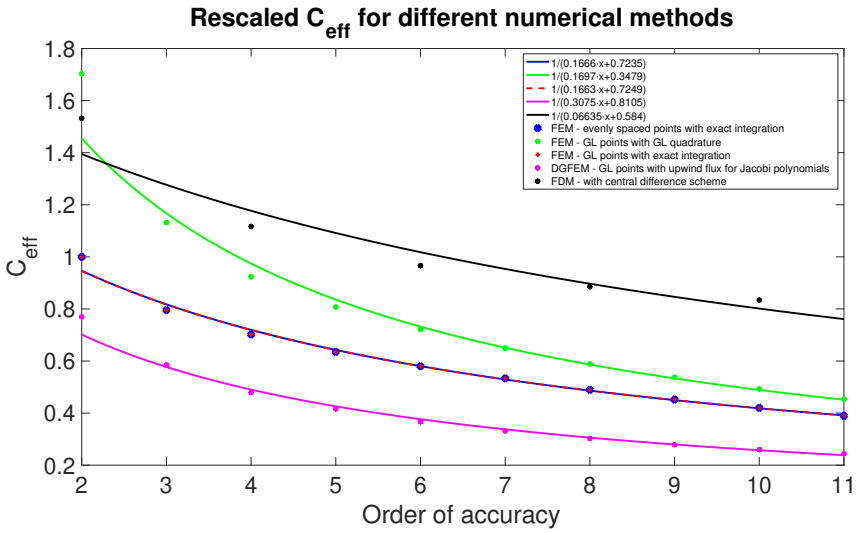

Figure 1: Dependence of the rescaled C_{eff} on the order of accuracy for different numerical methods.

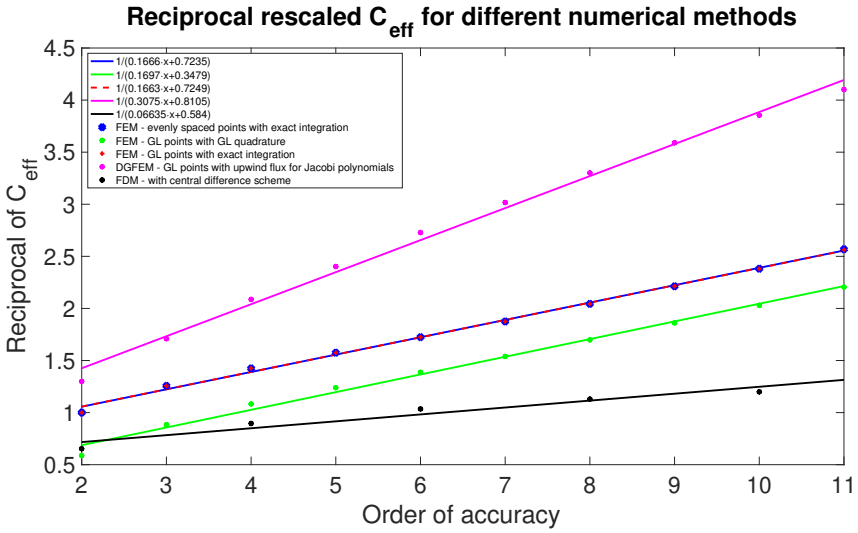

Figure 2: Dependence of the reciprocal of the rescaled C_{eff} on the order of accuracy for different numerical methods.

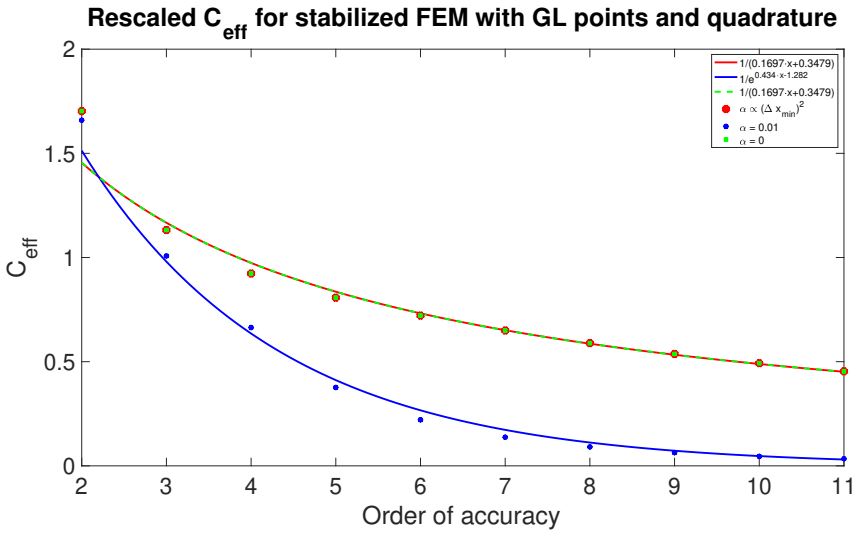

Figure 3: Dependence of the rescaled C_{eff} on the order of accuracy for Finite Element Method with stabilizing diffusion term, Gauss-Lobatto points and quadrature, for varying values of the stabilization factor α.

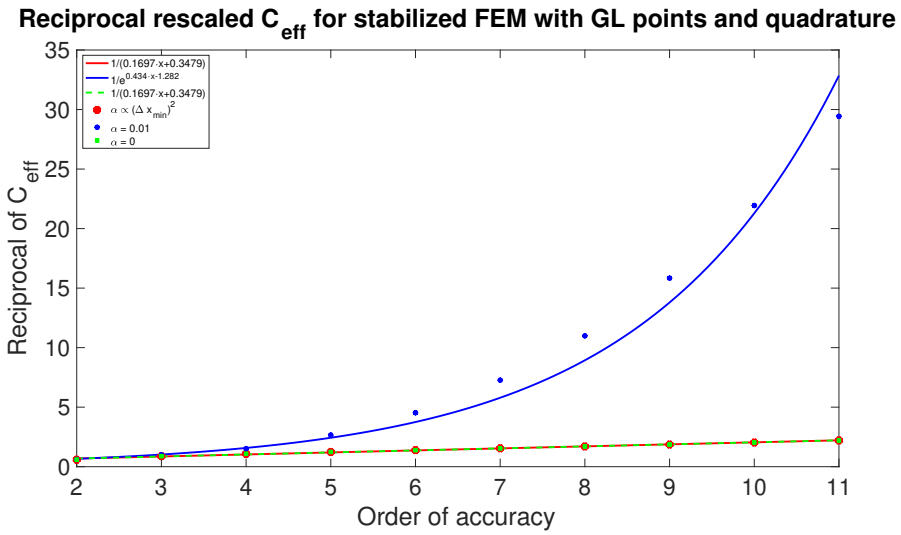

Figure 4: Dependence of the reciprocal of the rescaled C_{eff} on the order of accuracy for Finite Element Method with stabilizing diffusion term, Gauss-Lobatto points and quadrature, for varying values of the stabilization factor α.

5 Discussion

5.1 Dependence on polynomial interpolation points

Figure 1 shows how C_{eff} depends on the order of accuracy for FEM with both evenly spaced points and Gauss Lobatto points using exact integration. The graph clearly shows that these two methods have almost the same value for C_{eff} for every order of accuracy. Considering the regression lines for these two methods, there is a slight difference between them which probably is due to numerical errors. This means that there is no dependence between the maximum stable time step and the different interpolation points that were used.

5.2 Effect when increasing the order of the methods

Considering Figure 1 and Figure 2 it is clear that the rescaled C_{eff} decreases for all the methods of interest which implies that the restrictions on the time step increase with the order of accuracy for these methods. All the regression lines show that the reciprocal of C_{eff} has a linear trend. Due to this, all the restrictions on the time step will increase linearly with the order of accuracy.

The FD method has the highest C_{eff} value for almost every order of accuracy. One can also see that the regression line for this method decreases the least among the methods of interest. This implies that this method has the least restrictions on the time step (except for order 2) and also that the time step restrictions increase the least with the order.

The regression line to the FEM with GL points and quadrature decrease the fastest but it has the highest C_{eff} when the order of accuracy is 2. This suggests

that the restrictions on the time step for this method increase the most with the order although it has overall the second least restrictions on the time step. Also for the order of accuracy being 2, it works well since then it allows us to use a rather large time step.

One can see that C_{eff} for the methods FEM with evenly spaced points and exact integration respectively FEM using GL points and exact integration has almost the same value for each order of accuracy. The method using evenly spaced points decrease slightly faster but this we might disregard this due to numerical errors. These methods have overall the second most restrictions on the time step.

Figure 1-4 also show how C_{eff} of the DG-FEM behaves as the order of accuracy increase. The regression line corresponding to this method is below the regression lines for the other methods. This implies that the DG-FEM takes more restrictions on the time step than the other methods, no matter of the order of accuracy.

Regarding FEM with evenly spaced points and the DG-FEM in Figure 1, we can see an interesting behaviour. FEM may be seen as a more constrained method due to its continuity compared to the DG-FEM. This constraint on the method, leads to that this method has fewer degrees of freedom in the solution compared to DG-FEM. For these methods, the efficiency number depends on the time step, Δt, times the number of degrees of freedom, $N_{d.o.f}$. Since the value of $N_{d.o.f}$ is smaller for FEM, it seems that the time step restrictions becomes more loose for this method. On the contrary, having less constraints on the method, seems to imply tighter time step restrictions.

5.3 Consequence of applying stabilization

The effect of applying stabilization to the problem is shown in Figure 3 and Figure 4. These figures describe the rescaled C_{eff} respectively its reciprocal plotted towards the order of accuracy. Here, the method of interest is the stabilized FEM with Gauss-Lobatto points and quadrature. The figures show how the C_{eff} for this method against the order of accuracy depends on different values of the diffusion parameter α. We can see that the overall C_{eff} when using stabilization, $\alpha = 0.01$, is lower than without stabilization, $\alpha = 0$ or $\alpha \propto (\Delta x_{min})^2$. This suggests that the restrictions on the time step are higher for the stabilized problem. In Figure 4 it is clear that for no stabilization or $\alpha \propto (\Delta x_{min})^2$, the reciprocal of the C_{eff} increases linearly with the order of accuracy. When more stabilization is applied, the linear behaviour changes to exponential. This implies that the restrictions on the time step increase faster for the stabilized problem.

6 Conclusion

To conclude, the following observations are worth noting. The use of Gauss-Lobatto points instead of evenly spaced points did not affect the restrictions of the time step.

Next, the restrictions on the maximum stable time step increase with the order

of accuracy for every method that was considered. Furthermore, the use of stabilization causes even more restrictions on the time step as the order of accuracy increases.

Using FEM with Gauss-Lobatto points together with Gauss-Lobatto quadrature gives the least restrictions on the time step, while among the considered methods, DG-FEM has the most time step restrictions of the considered methods.

All considered methods produce ill-conditioned matrices when using higher orders than 11. An exception to this is the DG-FEM method. Therefore, an interesting development for this project would be to investigate even higher orders of accuracy.

References

[1] Jichun Li and Yi-Tung Chen. *Computational Partial Differential Equations Using MATLAB*. USA. Chapman and Hall/CRC. 2009.

[2] Hesthaven Jan and Warburton Tim. *Nodal Discontinuous Galerkin Methods. Algorithms, Analysis, and Applications*. USA. Springer. 2008.

[3] Gustafsson Bertil. *High Order Difference Methods for Time Dependent PDE*. Sweden. Springer . 2008.

[4] Weber Benjamin et all. *Stability analysis of finite element methods for the second order wave equation*. Sweden. October 1. 2020.

[5] Chapra Steven. *Applied numerical methods with MATLAB for engineers and scientists* . Singapore. Third edition. MC Graw Hill education 2012.

[6] M. G. Larson and F. Bengzon. *The Finite Element Method: Theory, Implementation, and Applications* . Sweden. Springer. 2013.

[7] W. Hundsdorfer and J. Verwer. *Numerical Solution of Time-Dependent Advection-DiffusionReaction Equations*. Berlin. Springer. 2003.

[8] N. Kovvali. *Theory and Applications of Gaussian Quadrature Methods*. Morgan and Claypool. 2011.

Appendix

Table 1: Values of the C_{eff} against the order of accuracy for FEM with evenly spaced points and exact integration, FEM with GL points and quadrature, FEM with GL points and exact integration, DG-FEM and FDM.

Order of Accuracy	C_{uni}	$C_{gaussLobatto}$	$C_{gaussLobattoExact}$	C_{dfem}	CFL
2	1.0000	1.7028	1.0000	0.7693	1.5323
3	0.7958	1.1319	0.7958	0.5851	-
4	0.7019	0.9232	0.7019	0.4792	1.1166
5	0.6349	0.8074	0.6349	0.4163	-
6	0.5799	0.7213	0.5799	0.3665	0.9661
7	0.5334	0.6497	0.5334	0.3315	-
8	0.4893	0.5889	0.4893	0.3029	0.8854
9	0.4519	0.5372	0.4519	0.2786	-
10	0.4200	0.4926	0.4200	0.2594	0.8339
11	0.3894	0.4536	0.3902	0.2438	-

Table 2: Values of the reciprocal of C_{eff} against the order of accuracy for FEM with evenly spaced points and exact integration, FEM with GL points and quadrature, FEM with GL points and exact integration, DG-FEM and FDM.

Order of Accuracy	$\frac{1}{C\_uni}$	$\frac{1}{C\_gaussLobatto}$	$\frac{1}{C\_gaussLobattoExact}$	$\frac{1}{C\_dfem}$	$\frac{1}{CFL}$
2	1.0000	0.5873	1.0000	1.2999	0.6526
3	1.2567	0.8835	1.2567	1.7092	-
4	1.4248	1.0832	1.4248	2.0869	0.8955
5	1.5749	1.2385	1.5749	2.4023	-
6	1.7244	1.3863	1.7244	2.7282	1.0350
7	1.8748	1.5391	1.8748	3.0162	-
8	2.0436	1.6981	2.0436	3.3009	1.1294
9	2.2128	1.8615	2.2128	3.5898	-
10	2.3810	2.0301	2.3809	3.8552	1.1991
11	2.5682	2.2048	2.5628	4.1011	-

Table 3: Shows the values of the C_{eff} against the order of accuracy for the stabilized FEM using evenly spaced points and with different values on the diffusion term.

Order of Accuracy	$\alpha = 0$	$\alpha = 0.01$	$\alpha = 0.001$	$\alpha \propto (\Delta x_{min})^2$
2	1.7028	1.6592	1.7024	1.7028
3	1.1319	1.0075	1.1304	1.1319
4	0.9232	0.6638	0.9204	0.9232
5	0.8074	0.3762	0.8018	0.8074
6	0.7213	0.2209	0.7121	0.7213
7	0.6497	0.1376	0.6368	0.6497
8	0.5889	0.0910	0.5730	0.5889
9	0.5372	0.0631	0.5140	0.5372
10	0.4926	0.0456	0.4614	0.4926
11	0.4536	0.0340	0.4122	0.4536

Table 4: Shows the values of the reciprocal of the C_{eff} against the order of accuracy for the stabilized FEM using evenly spaced points and with different values on the diffusion term.

Order of Accuracy	$\alpha = 0$	$\alpha = 0.01$	$\alpha = 0.001$	$\alpha \propto (\Delta x_{min})^2$
2	0.5873	0.6027	0.5874	0.5873
3	0.8835	0.9926	0.8847	0.8835
4	1.0832	1.5065	1.0864	1.0832
5	1.2385	2.6583	1.2473	1.2385
6	1.3863	4.5269	1.4043	1.3863
7	1.5391	7.2693	1.5702	1.5391
8	1.6981	10.9899	1.7453	1.6981
9	1.8615	15.8390	1.9456	1.8615
10	2.0301	21.9367	2.1674	2.0301
11	2.2048	29.4318	2.4260	2.2048