

# Dispersion Analysis of symplectic integrators for the 1D wave equation

Ladislav Jacobe de Naurois

24.12.2014

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Dispersion Relation</b>	<b>2</b>
2.1	Dispersion Relation of Semi-Discrete Evolution schemes . . . . .	3
2.2	Dispersion Relation of Fully-Discrete Evolution schemes . . . . .	4
<b>3</b>	<b>Spatial Discretization for Lagrangian Finite Elements</b>	<b>7</b>
3.1	Linear Lagrangian Finite Elements . . . . .	8
3.2	Quadratic Lagrangian Finite Elements . . . . .	10
3.3	Cubic Lagrangian Finite Elements . . . . .	11
<b>4</b>	<b>Time-stepping</b>	<b>13</b>
4.1	Leapfrog integration . . . . .	13
4.2	Symplectic Euler . . . . .	14
4.3	Higher order symplectic integrators . . . . .	17
4.4	Empirical Convergence Rates for symplectic time-stepping algorithms . . . . .	18
<b>5</b>	<b>Dispersion Relation of the fully discrete evolution problem</b>	<b>20</b>
5.1	Leapfrog integration . . . . .	20
5.2	Comparison - LFEM with symplectic integrators . . . . .	22
5.3	Comparison - QFEM with symplectic integrators . . . . .	23
5.4	Comparison - CFEM with symplectic integrators . . . . .	23
5.5	Semi-implicit Euler time-stepping . . . . .	24
5.6	Symplectic $2^{nd}$ -order time-stepping . . . . .	25
5.7	Symplectic $3^{rd}$ -order time-stepping . . . . .	26
5.8	Symplectic $4^{th}$ -order time-stepping . . . . .	27
<b>6</b>	<b>Conclusion</b>	<b>27</b>

<b>7 Appendix</b>	<b>28</b>
7.1 Code - Leapfrog time-stepping (model problem) . . . . .	28
7.2 Code - Symplectic time-stepping (model problem) . . . . .	29
7.3 Code - (Animation) Linear Finite Element for Leapfrog time-stepping . . . . .	30
7.4 Code - (Animation) Linear Finite Element for symplectic time-stepping . . . . .	32
7.5 Code - Dispersion relation for semi-discrete scheme . . . . .	34
7.6 Code - Dispersion relation for Leapfrog time-stepping . . . . .	36
7.7 Code - Dispersion relation for symplectic time-stepping algorithms . . . . .	38
<b>References</b>	<b>41</b>

# 1 Introduction

The subject of this term paper is a quantitative analysis of numerical approximations to the one dimensional wave equation. To that effect, let  $T \in (0, \infty)$  and  $u : \mathbb{R} \times [0, T] \rightarrow \mathbb{R}$  be a function which is twice continuously differentiable in both space and time with globally bounded derivatives and let  $c \in (0, \infty)$ . Furthermore, let  $u_0, v_0 \in C^1(\mathbb{R}, \mathbb{R})$ . Then  $u \in C^2(\mathbb{R} \times [0, T], \mathbb{R})$  is said to be a solution of the one dimensional wave equation with Dirichlet boundary conditions and initial conditions  $u_0$  and  $v_0$ , if it holds that for all  $t \in [0, T]$

$$\lim_{x \rightarrow -\infty} u(x, t) = \lim_{y \rightarrow -\infty} u(y, t) = 0 \tag{1}$$

for all  $x \in \mathbb{R}$  it holds that

$$u(x, 0) = u_0(x) \text{ and } \frac{\partial u}{\partial t}(x, 0) = v_0(x) \tag{2}$$

and for all  $(y, s) \in \mathbb{R} \times [0, T]$  it holds that

$$\frac{\partial^2 u}{\partial t^2}(y, s) = c^2 \frac{\partial^2 u}{\partial x^2}(y, s) \tag{3}$$

It is a well-known fact from introductory analysis, that for arbitrary functions  $u_0, v_0 \in C^1([0, T] \times \mathbb{R}, \mathbb{R})$ , a classical solution to the one dimensional wave equation with Dirichlet boundary conditions (cf. (1)) and initial value problem (IVP) (cf. (2)) is given via the so-called **D'Alembert formula**

$$u(x, t) = \frac{1}{2} [u_0(x - ct) + u_0(x + ct)] + \frac{1}{2c} \int_{x-ct}^{x+ct} v_0(y) dy \tag{4}$$

In the following we will approximate solutions to the one dimensional wave equation using spatial approximations by means of Lagrangian Finite Elements of linear, quadratic and cubic degree coupled with symplectic time-stepping algorithms of higher consistency orders.

The symplectic Euler algorithm as well as the Stormer-Verlet or Leapfrog time-stepping algorithm are classical examples of symplectic integrators. Both the semi-implicit Euler and Stormer-Verlet/Leapfrog time-stepping algorithms exhibit a first- and second-order consistency respectively and have the additional property that they preserve the (modified) Energy functional of the system exactly for sufficiently small step-size. For a long time, these were the only known integrators which exhibited these properties, until the work of Ronald D Ruth (cf. [2] and [3]), when canonical examples of higher-order symplectic integrators of third- and fourth-order consistency were introduced.

## 2 Dispersion Relation

Let us begin our discussion with a formal introduction to the abstract concept of (numerical) dispersion relation. To that effect, consider the following

**Definition 1.** Let  $K_t, L_x : C^2(\mathbb{R} \times [0, T]; \mathbb{R}) \rightarrow C(\mathbb{R} \times [0, T]; \mathbb{R})$  be linear differential operators in time and space respectively. An element  $u \in C^2(\mathbb{R} \times [0, T]; \mathbb{R})$  is said to be a solution to the PDE  $(K_t + L_x)u = 0$ , if it holds that for all  $x \in \mathbb{R}, t \in [0, T]$

$$(K_t + L_x)u(x, t) = 0 \quad (5)$$

**Definition 2** (Spatial and Temporal shift operators). Let  $\Delta t$  and  $\Delta x > 0$ , let  $T_{\Delta t}$  and  $S_{\Delta x}$  be operators on the  $C^2(\mathbb{R} \times [0, T]; \mathbb{R})$  with the property that for all  $(x, t) \in \mathbb{R} \times [0, T]$  it holds that

$$T_{\Delta t}u(x, t) = u(x, t + \Delta t) \text{ and } S_{\Delta x}u(x, t) = u(x + \Delta x, t) \quad (6)$$

Furthermore, assume that the differential operators in both space and time commute with the translation operators in both space and time respectively, i.e. for all  $(x, t) \in \mathbb{R} \times [0, T]$ , it holds that

$$K_t \circ T_{\Delta t} = T_{\Delta t} \circ K_t \text{ and } L_x \circ S_{\Delta x} = S_{\Delta x} \circ L_x \quad (7)$$

**Theorem 3.** Assuming the setting in the definition above, it holds that for all  $\omega, \kappa \in \mathbb{R}$  the exponential functions  $\omega \mapsto e^{i\omega t}$  and  $\kappa \mapsto e^{i\kappa x}$  are eigenfunctions of the linear differential operators  $K_t$  and  $L_x$  respectively. Specifically, we may define eigenvalues

$$\lambda_t : \omega \in \mathbb{R} \mapsto \lambda_t(\omega) \in \mathbb{C} \text{ and } \lambda_x : \kappa \in \mathbb{R} \mapsto \lambda_x(\kappa) \in \mathbb{C} \quad (8)$$

as mapping with the property that for all  $\omega, \kappa \in \mathbb{R}$ , it holds that

$$K_t \underbrace{(t \mapsto e^{i\omega t})}_{\in C^\infty([0, T]; \mathbb{C})} = \lambda_t(\omega) (t \mapsto e^{i\omega t}) \text{ and } L_x \underbrace{(x \mapsto e^{i\kappa x})}_{\in C^\infty(\mathbb{R}; \mathbb{C})} = \lambda_x(\kappa) (x \mapsto e^{i\kappa x}) \quad (9)$$

**Definition 4.** Let  $K_t$  and  $L_x$  be linear differential operators in space and time respectively. Consider the eigenvalues  $\lambda_t(\omega), \lambda_x(\kappa) : \mathbb{R} \rightarrow \mathbb{C}$  of the spatial and temporal linear differential operators. Given the PDE in (5), we may define the **continuous dispersion relation** of the PDE as

$$\lambda_t(\omega) - \lambda_x(\kappa) \stackrel{!}{=} 0 \quad (10)$$

Furthermore, the equation defining the dispersion relation may also be identified as a graph  $\Gamma \subseteq \mathbb{R} \times \mathbb{R}$ . That is, for all  $\bar{\kappa} \in \mathbb{R}$ , it holds that for arbitrary  $\omega_{\bar{\kappa}} \in \mathbb{R}$

$$(\bar{\kappa}, \omega_{\bar{\kappa}}) \in \Gamma \iff \lambda_t(\omega_{\bar{\kappa}}) - \lambda_x(\bar{\kappa}) \stackrel{!}{=} 0 \quad (11)$$

**Lemma 5.** Let  $c > 0$  and  $K_t, L_x$  be the linear differential operators in space and time respectively defined as

$$K_t : u \mapsto \frac{\partial^2 u}{\partial t^2} \text{ and } L_x : u \mapsto c^2 \frac{\partial^2 u}{\partial x^2} \quad (12)$$

Then the PDE (5) reduces to the 1D wave equation or Cauchy Problem with phase velocity  $c > 0$  and the **dispersion relation** is given via

$$\omega^2 = c^2 \kappa^2 \iff \omega = \pm c\kappa \quad (13)$$

In other words, for all  $\kappa, \omega \in \mathbb{R}$ , the (real part of) function  $\mu = e^{i(\kappa x - \omega t)} \in C^2(\mathbb{R} \times [0, T]; \mathbb{C})$  solves the PDE (3) if and only if for every  $\kappa \in \mathbb{R}$ , it holds that  $\omega \in \mathbb{R}$  is given by the dispersion relation in equation (13) above.

*Proof of Lemma 5.* For fixed  $\Delta x, \Delta t > 0$ , the chain rule (see e.g. Analysis I) concludes that the differential operators  $K_t$  and  $L_x$  satisfy property (6) for  $T_{\Delta t}, S_{\Delta x}$  translation operators in space and time respectively.

Hence, theorem 3 concludes that the eigenfunctions of both spatial and temporal differential operators are given by the exponential functions.

Fix  $\kappa \in \mathbb{R}$  and consider  $\{\mu_\omega : \omega \in \mathbb{R}\}$ , the set of functions indexed by  $\omega \in \mathbb{R}$  with the property that for fixed  $\omega \in \mathbb{R}$ ,  $\mu_\omega$  is defined as

$$\mu_\omega : \begin{cases} \mathbb{R} \times [0, T] & \rightarrow & \mathbb{C} \\ (x, t) & \mapsto & e^{i(\kappa x - \omega t)} \end{cases} \quad (14)$$

Consider the following calculation

$$c^2 \frac{\partial^2}{\partial x^2} \mu_\omega = -c^2 \kappa^2 \mu_\omega \stackrel{!}{=} -\omega^2 \mu_\omega = \frac{\partial^2}{\partial t^2} \mu_\omega \quad (15)$$

Therefore, for fixed  $\kappa \in \mathbb{R}$ , the function  $\mu_\omega$  satisfied the 1D Cauchy Problem if and only if

$$c^2 = \frac{\omega^2}{\kappa^2} \iff c = \left| \frac{\omega}{\kappa} \right| \iff \omega = \pm c\kappa \quad (16)$$

This concludes the proof of Lemma 5 □

## 2.1 Dispersion Relation of Semi-Discrete Evolution schemes

The chief goal of this term paper is to provide a concise analysis of the dispersion relation for a specific set of fully-discrete schemes introduced in Section 1. Hence, in an attempt to illustrate the concept of the dispersion relation for a fully-discrete numerical scheme, let us begin by considering a spatially semi-discrete scheme using the general theory of the Lagrangian Finite Element method and provide a general framework for determining the dispersion relation in that setting.

**Lemma 6.** *Let  $K_t, L_x$  be (continuous) linear differential operators in time and space respectively. Assume furthermore that for all  $\Delta x, \Delta t > 0$  the differential operators commute with spatial and temporal shift operators (c.f. (6)). Furthermore, let  $p \in \mathbb{N}$  and let  $\mathbf{L}_x : V_x^{\mathbb{Z}} \rightarrow V_x^{\mathbb{Z}}, V_x \cong \mathbb{C}^p$  be the corresponding (spatial) discretization of  $L_x$  on an equidistant grid  $\mathcal{M} = \mathbb{Z} \cdot \Delta x \subseteq \mathbb{R}$  with  $p$  degrees of freedom (in space).*

Then the eigenfunctions of the discrete differential operator in space are given via the map(s).

$$\kappa \mapsto \left( \vec{\xi} \cdot e^{i\kappa j \Delta x} \right) \text{ for suitable vectors } \vec{\xi} \in V_x \quad (17)$$

Furthermore, the eigenvalues are given via the map

$$\lambda_x : \mathbb{R} \rightarrow L(V_x) \quad (18)$$

such that for all  $\kappa \in \mathbb{R}$  and suitable  $\vec{\xi} \in V_x$  it holds that

$$\mathbf{L}_x \left( \vec{\xi} \cdot e^{i\kappa j \Delta x} \right)_{j \in \mathbb{Z}} = \left( \lambda_x(\kappa) \left( \vec{\xi} \right) \cdot e^{i\kappa j \Delta x} \right)_{j \in \mathbb{Z}} \quad (19)$$

**Remark 7.** Note that for fixed  $\Delta x > 0$ , the translation operator in space  $S_{\Delta x}$  commutes with  $\mathbf{L}_x$ , the discretization of the linear differential operator in space, i.e. it holds that

$$S_{\Delta x} \circ \mathbf{L}_x = \mathbf{L}_x \circ S_{\Delta x} \quad (20)$$

This fact is a consequence of the assumption on the (continuous) linear differential operator in space, which is assumed commute with the unit translation operator in space.

## 2.2 Dispersion Relation of Fully-Discrete Evolution schemes

**Definition 8.** Let  $\Delta t, \Delta x > 0$  (w.l.o.g. assume that  $\Delta t = \frac{T}{N}, N \in \mathbb{N}$ ) and let  $K_t, L_x$  be linear differential operators in time and space and assume that both operators commute with the spatial and temporal unit translation operators respectively. Define  $\mathbf{K}_t, \mathbf{L}_x$  as the discrete differential operators in space and time on the equidistant spatial and temporal grid  $\mathcal{M} = \Delta x \cdot \mathbb{Z} \subseteq \mathbb{R}$  and  $\mathcal{N} = \Delta t \cdot \{0, \dots, N\} \subseteq [0, T]$  respectively.

The fully-discrete evolution equation corresponding to the PDE (5) is governed by the update scheme that for all  $n \in \{0, \dots, N\}$  it holds that

$$(\mathbf{K}_t - \mathbf{L}_x)(\vec{\mu}^{(k)})_{k \in \{0, \dots, n+1\}} \stackrel{!}{=} 0 \quad (21)$$

Let  $k \in \mathbb{N}$ . Assuming the context of a  $k$  multi-step method, resolving the above equation, we arrive at a fully-discrete evolution scheme  $\mu^{(n+1)} = \mathfrak{G}_{\Delta t, \Delta x}(\mu^{(n)}, \dots, \mu^{(n-k+1)})$  for a suitable vector space  $V_{x,t} \cong \mathbb{C}^q$  for some  $q > 0$  and a corresponding operator (so-called lattice operator) on

$$\mathfrak{G}_{\Delta t, \Delta x} : (V_{x,t}^{\mathbb{Z}})^k \rightarrow V_{x,t}^{\mathbb{Z}} \quad (22)$$

**Remark 9.** There is some notational difficulty, in the above, as our lattice operator generally operates on some exponent of  $V_{x,t}^{\mathbb{Z}}$ , for instance in the context of single-step methods,  $\mathfrak{G}_{\Delta t, \Delta x}$  operates on  $V_{x,t}^{\mathbb{Z}}$  and in two-step methods it operates on  $(V_{x,t}^{\mathbb{Z}})^2$ . Abusing notation a bit, we will write that  $\mathfrak{G}_{\Delta t, \Delta x}$  operates on  $V_{x,t}^{\mathbb{Z} \times \mathbb{Z}}$  in order to facilitate discussion on the subject matter in the following.

**Proposition 10.** Let  $\Delta t, \Delta x > 0$  and  $\mathfrak{G}_{\Delta t, \Delta x} : V_{x,t}^{\mathbb{Z} \times \mathbb{Z}} \rightarrow V_{x,t}^{\mathbb{Z} \times \mathbb{Z}}$  the  $(\mathbb{Z} \times \mathbb{Z})$ -lattice operator of the fully-discrete evolution equation.

The eigenfunctions of this lattice operator are given via

$$(\kappa, \omega) \mapsto \left( \vec{\xi} \cdot e^{i(\kappa j \Delta x - \omega n \Delta t)} \right)_{j, n \in \mathbb{Z}} \text{ for suitable } \vec{\xi} \in V_{x,t} \quad (23)$$

Note: The vector  $\vec{\xi} \in V_{x,t}$  is called a "lattice vector".

Furthermore, for all  $\kappa, \omega \in \mathbb{R}$ , there exists an operator  $\mathfrak{g}_{\Delta t, \Delta x} : [-\pi, \pi] \times [-\pi, \pi] \rightarrow L(V_{x,t})$  such that for all lattice vectors  $\vec{\xi} \in V_{x,t}$  it holds that

$$\mathfrak{G}_{\Delta t, \Delta x} \left( \vec{\xi} \cdot (e^{i(\kappa j \Delta x + \omega n \Delta t)})_{j, n \in \mathbb{Z}} \right) = \left( \mathfrak{g}_{\Delta t, \Delta x}(\kappa, \omega) \left( \vec{\xi} \right) \cdot e^{i(\kappa j \Delta x + \omega n \Delta t)} \right)_{j, n \in \mathbb{Z}} \quad (24)$$

**Definition 11.** The vector spaces  $V_x^{\mathbb{Z}}$  (spatially semi-discrete),  $V_{x,t}^{\mathbb{Z} \times \mathbb{Z}}$  (fully-discrete) are called the space of Lattice- (or Grid-) functions. They consist of all maps from  $\mathbb{Z}$  resp.  $\mathbb{Z} \times \mathbb{Z}$  to our suitable vector space  $V_x \cong \mathbb{C}^p$  and  $V_{x,t} \cong \mathbb{C}^q$  for  $p, q \in \mathbb{N}$  respectively.

**Definition 12** (Definition of the discrete Dispersion Relation). *Let  $\Delta t, \Delta x > 0$  and consider  $\mathbf{K}_t, \mathbf{L}_x$  the discrete differential operators in space and time on an equidistant spatial grid of width  $\Delta x > 0$  and time-step  $h = \Delta t > 0$ . Let  $q \in \mathbb{N}$  and let  $V_{x,t} = \mathbb{C}^q$  such that the lattice operator*

$$\mathfrak{G}_{\Delta t, \Delta x} : V_{x,t}^{\mathbb{Z} \times \mathbb{Z}} \rightarrow V_{x,t}^{\mathbb{Z} \times \mathbb{Z}} \quad (25)$$

*corresponds to the fully-discrete evolution problem of the fully-discrete linear differential operators  $\mathbf{K}_t, \mathbf{L}_x$  and for fixed  $\kappa, \omega \in [-\pi, \pi]$ , let*

$$\mathfrak{g}_{\Delta t, \Delta x}(\kappa, \omega) : V_{x,t} \rightarrow V_{x,t} \quad (26)$$

*be the local operator corresponding to the lattice operator. Then the **(numerical) dispersion relation** for a fully-discrete evolution problem is given via*

$$\det(\mathfrak{g}_{\Delta t, \Delta x}(\kappa, \omega)) \stackrel{!}{=} 0 \quad (27)$$

**Remark 13** (Example). *Consider  $L_t$  and  $K_x$  the linear differential operators in time and space that commute with the unit translation operator in time and space respectively. Let  $\mathbf{L}_x$  be the discrete differential operator in space with corresponding mass and stiffness matrices  $\mathbf{M}$  and  $\mathbf{A}$  respectively. Corresponding to the PDE (3), consider the linear ODE*

$$\mathbf{M} \frac{\partial^2 \vec{\mu}}{\partial t^2}(t) + \mathbf{A} \vec{\mu}(t) \stackrel{!}{=} 0 \quad (28)$$

*Let  $\kappa \in \mathbb{R}$  and consider  $\tilde{\mathbf{M}} = \tilde{\mathbf{M}}(\kappa)$  and  $\tilde{\mathbf{A}} = \tilde{\mathbf{A}}(\kappa)$ ; the local operators corresponding to  $\mathbf{M}, \mathbf{A}$ . Let  $\mathbf{K}_t$  be the discrete linear differential operator in time with corresponding stability function  $\Psi_{\mathbf{K}} : V_{x,t} \rightarrow V_{x,t}$  for a suitable vector space  $V_{x,t}$ . Assume that  $\mathbf{K}_t$  defines a single-step time-stepping method.*

**Note:** *There is no true single-step method for a second-order ODE, however through a suitable transformation by means of an auxiliary variable we may transform a second-order ODE into a first-order ODE, which admits a suitable partitioned single-step time-stepping method.*

*Let  $r_1, r_2, s_1, s_2 \in \mathbb{N}$  and  $\Psi_{\mathbf{K}} : \mathbb{C}^{r_1 \times r_2} \rightarrow \mathbb{C}^{s_1 \times s_2}$  be the stability function of an arbitrary (single-step) time-stepping algorithm and let  $\mathfrak{G}_{\Delta t, \Delta x}$  be the operator on lattice functions of the discrete evolution problem*

$$\mathfrak{G}_{\Delta t, \Delta x} : V_{x,t}^{\mathbb{Z} \times \mathbb{Z}} \rightarrow V_{x,t}^{\mathbb{Z} \times \mathbb{Z}} \quad (29)$$

*Furthermore, let  $\mathfrak{g}_{\Delta x, \Delta t}$  be the local operator on lattice vectors*

$$\mathfrak{g}_{\Delta x}(\kappa) : \vec{\xi} \in V_x \rightarrow \left( \Psi_{\mathbf{K}}(\tilde{\mathbf{M}}^{-1}(\kappa) \tilde{\mathbf{A}}(\kappa)) \left( \vec{\xi} \right) \right) \in V_x \quad (30)$$

*Then the **(numerical) dispersion relation** defined via  $\det(\mathfrak{g}_{\Delta t, \Delta x}) \stackrel{!}{=} 0$  may be determined as follows.*

**0. Consider the effect of eigenfunctions**

$$\mathfrak{G}_{\Delta t, \Delta x} : \left( \vec{\xi} \cdot e^{i(\kappa j \Delta x - \omega n \Delta t)} \right)_{j,n \in \mathbb{Z}} \in V_{x,t}^{\mathbb{Z} \times \mathbb{Z}} \rightarrow \left( \Psi_{\mathbf{K}}(\mathbf{M}^{-1} \mathbf{A}) \left( \vec{\xi} \cdot e^{i(\kappa j \Delta x - \omega n \Delta t)} \right) \right)_{j,n \in \mathbb{Z}} \in V_{x,t}^{\mathbb{Z} \times \mathbb{Z}} \quad (31)$$

Furthermore, let  $n \in \mathbb{N}$  and let  $\mathfrak{g}_{\Delta x, \Delta t}$  be the local operator on lattice vectors such that for all  $\kappa, \omega \in \mathbb{R}$ , the map  $\mathfrak{g}_{\Delta x, \Delta t}(\kappa, \omega)$  is defined via

$$\mathfrak{g}_{\Delta x, \Delta t}(\kappa) : \vec{\xi} \cdot e^{i\omega n \Delta t} \in V_{x,t} \rightarrow \left( \Psi_{\mathbf{K}}(\tilde{\mathbf{M}}^{-1}(\kappa) \tilde{\mathbf{A}}(\kappa)) \left( \vec{\xi} \right) \right) \cdot e^{i\omega n \Delta t} \in V_{x,t} \quad (32)$$

**1. Diagonalize in time:** Since  $\mathfrak{G}$  provides a single-step update scheme via  $\vec{\mu}^{(n+1)} = \mathfrak{G} \vec{\mu}^n$ , for all  $\kappa \in [-\pi, \pi]$  we may define the map

$$\mathfrak{g}_{\Delta x}(\kappa) : \vec{\xi} \in V_{x,t} \rightarrow \left( \Psi_{\mathbf{K}}(\tilde{\mathbf{M}}^{-1}(\kappa) \tilde{\mathbf{A}}(\kappa)) \left( \vec{\xi} \right) \right) \in V_{x,t} \quad (33)$$

Then, for all lattice vectors  $\vec{\xi} \in V_{x,t}$  it holds that for all  $n \in \mathbb{N}$

$$\vec{\xi} \cdot \left( e^{i(\kappa j \Delta x - \omega(n+1) \Delta t)} \right)_{j \in \mathbb{Z}} = \mathfrak{G}_{\Delta t, \Delta x} \left( \vec{\xi} \cdot e^{i(\kappa j \Delta x - \omega n \Delta t)} \right)_{j \in \mathbb{Z}} \iff \vec{\xi} \cdot e^{i\omega \Delta t} = \mathfrak{g}_{\Delta x}(\kappa) \left( \vec{\xi} \right) \quad (34)$$

Hence, for fixed  $\kappa \in \mathbb{R}$ , the local operator on lattice vectors corresponding to the lattice operator  $\mathfrak{G}_{\Delta t, \Delta x}$  is given via

$$\mathfrak{g}_{\Delta t, \Delta x} : (\kappa, \omega) \in [-\pi, \pi] \times [-\pi, \pi] \rightarrow (\text{Id}_{V_{x,t}} \cdot e^{i\omega \Delta t} - \mathfrak{g}_{\Delta x}(\kappa)) \in L(V_{x,t}) \quad (35)$$

**2. Diagonalize in space:** For all  $\kappa \in \mathbb{R}$ , solving the spatially discrete generalized eigenvalue problem gives rise to eigenfunctions  $\lambda : [-\pi, \pi] \rightarrow \mathbb{C}$  such that for all  $\kappa \in [-\pi, \pi]$  it holds that

$$\det(\mathfrak{g}_{\Delta x}(\kappa) - \lambda(\kappa) \cdot \text{Id}_{V_{x,t}}) = 0 \quad (36)$$

**3. Compute the numerical dispersion relation:** For every eigenfunction  $\lambda : [-\pi, \pi] \rightarrow \mathbb{C}$ , diagonalizing in space allows us to compute the **(numerical) dispersion relation** via

$$e^{i\omega \Delta t} = \lambda(\kappa) \iff \omega = \frac{\log(\lambda(\kappa))}{i \Delta t} \quad (37)$$

Hence, we may conclude that

$$\det(\mathfrak{g}_{\Delta t, \Delta x}) \stackrel{!}{=} 0 \iff \omega = \frac{\log(\tilde{\lambda}(\kappa))}{i \Delta t} \quad (38)$$

for all  $\lambda : [-\pi, \pi] \rightarrow \mathbb{C}$  that satisfy property (36)

### 3 Spatial Discretization for Lagrangian Finite Elements

Consider a spatial discretization according to Lagrangian Finite Element on an equidistant (1D) spatial grid of width  $\Delta x > 0$ , call it  $\Delta x \cdot \mathbb{Z} \subseteq \mathbb{R}$ . Hence for  $p \in \mathbb{N}$ , and  $\mathcal{B} = \{b_j^k : j \in \mathbb{Z}, k \in \{1, \dots, p\}\}$ , a finite element basis of  $\mathbb{R}$  of degree  $p$  with grid points  $\{j \Delta x : j \in \mathbb{Z}\} = \Delta x \cdot \mathbb{Z}$ , we may compute both mass and stiffness matrix as

$$\mathbf{M} = \left( \int_{\mathbb{R}} b_i^l(x) b_j^m(x) dx \right)_{i,j \in \{1, \dots, N\}, l,m \in \{1, \dots, p\}} \quad (39)$$

$$\mathbf{A} = \left( \int_{\mathbb{R}} \dot{b}_i^l(x) \dot{b}_j^m(x) dx \right)_{i,j \in \{1, \dots, N\}, l,m \in \{1, \dots, p\}} \quad (40)$$

PDE (3) corresponds to the following ODE

$$\mathbf{M} P_{\Delta x}^p \left( \frac{\partial^2 u}{\partial t^2}(x, t) \right) = -c^2 \mathbf{A} P_{\Delta x}^p (u(x, t)) \quad (41)$$

For  $\mathcal{B} = \{b_j(x)\}_{j \in \mathbb{Z}}$  an arbitrary (countable) basis of our Finite Element space, we may define

$$V_x = \text{span}_{\mathbb{R}}(\mathcal{B}) = \left\{ \sum_{j \in \mathbb{Z}} a_j \cdot b_j(x) : (a_j)_{j \in \mathbb{Z}} \subset \mathbb{R} \right\} \subseteq C^\infty(\mathbb{R}, \mathbb{R}) \quad (42)$$

**Notation** Based on the definition of  $V_x$ , we may identify an element  $\mu(x) = \sum_{j \in \mathbb{Z}} \mu_j \cdot b_j(x) \in V_x$  with the vector  $\vec{\mu} = (\mu_j)_{j \in \mathbb{Z}} \in \mathbb{R}^{\mathbb{Z}}$ .

Finally, in order to adequately approximate  $u(x, t)$ , a solution to the wave equation (3), we will need to introduce a time component. To that effect, it would suffice to consider

$$V_{x,t} = \text{span}_{C^2([0,T], \mathbb{R})}(\mathcal{B}) = \left\{ \sum_{j \in \mathbb{Z}} a_j(t) \cdot b_j(x) : (a_j(t))_{j \in \mathbb{Z}} \subset C^2([0, T], \mathbb{R}) \right\} \subseteq C^2([0, T] \times \mathbb{R}, \mathbb{R}) \quad (43)$$

**Notation** Based on the definition of  $\mathcal{V}_t$ , we may identify an element  $\mu(t) = \sum_{j \in \mathbb{Z}} \mu_j(t) \cdot p_j(x) \in \mathcal{V}_{x,t}$  with the function  $\vec{\mu} : t \in [0, T] \mapsto \vec{\mu}(t) = (\mu_j(t))_{j \in \mathbb{Z}} \in C^2([0, T], \mathbb{R})^{\mathbb{Z}}$ .

Finally, let us define the map  $\pi_{\mathcal{B}} : C^2(\mathbb{R} \times [0, T], \mathbb{R}) \rightarrow \mathcal{V}_{x,t}$  with property that for all functions  $(u : \mathbb{R} \times [0, T] \rightarrow \mathbb{R}) \in C^2(\mathbb{R} \times [0, T], \mathbb{R})$  it holds that

$$\pi_{\mathcal{B}}(u) = \sum_{j \in \mathbb{Z}} \mu_j(t) \cdot b_j(x) \cong \vec{\mu}(t) \in \mathcal{V}_{x,t} \quad (44)$$

The framework above allows us to illustrate our notation through natural examples. Throughout the following we will (without mention) utilize the fact that spatial discretization of the 1D state space  $\mathbb{R}$  transforms the PDE (3) into the following ODE

$$\mathbf{M} \frac{\partial^2 \vec{\mu}}{\partial t^2}(t) = -c^2 \mathbf{A} \vec{\mu}(t) \iff \mathbf{M} \frac{\partial^2 \vec{\mu}}{\partial t^2}(t) + c^2 \mathbf{A} \vec{\mu}(t) = 0 \quad (45)$$

### 3.1 Linear Lagrangian Finite Elements

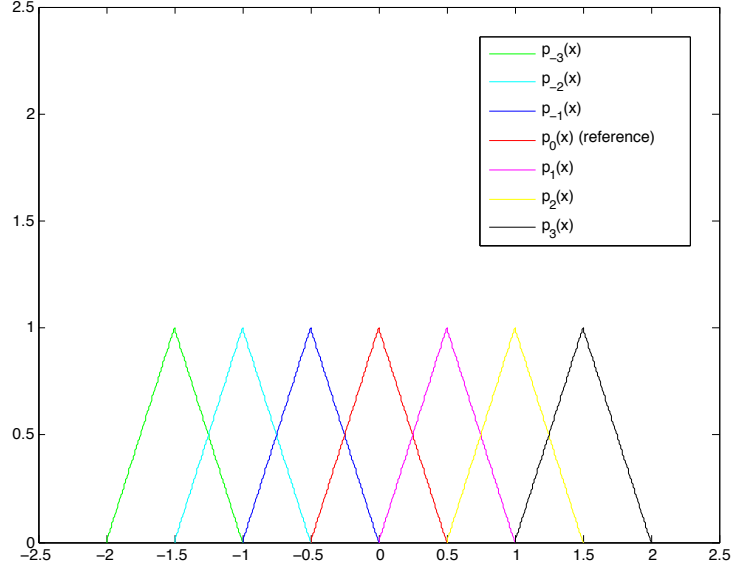
In order to begin our discussion, we must first determine the proper spatial discretization. For that purpose, let us consider an equidistant grid of width  $\Delta x > 0 : \Delta x \cdot \mathbb{Z} \subseteq \mathbb{R}$ . Then we may define the function  $p_0(x) \in C(\mathbb{R}, \mathbb{R})$  with the property that

$$\begin{aligned} \text{supp}(p_0) &= \{x \in \mathbb{R} : p_0(x) \neq 0\} \subseteq [-\Delta x, \Delta x] \subset \mathbb{R} \\ p_0(x) &= \left( \frac{x + \Delta x}{\Delta x} \right) \mathbb{1}_{[-\Delta x, 0]}(x) + \left( \frac{\Delta x - x}{\Delta x} \right) \mathbb{1}_{[0, \Delta x]}(x) \end{aligned} \quad (46)$$

Then we may define the set  $\mathcal{B}_1$  of linear basis functions on  $\mathbb{R}$  as follows

$$\mathcal{B}_1 = \{ p_j(x) = p_0(x - j\Delta x) : j \in \mathbb{Z} \} \quad (47)$$





Plot of basis functions  $p_j(x)$  for  $j \in \{-3, -2, -1, 0, 1, 2, 3\}$

Performing the relevant computations for both Mass and Stiffness Matrix for the basis  $\mathcal{B}_1$ , we arrive at the following stencil for the both mass and stiffness matrix in the case of Linear Lagrangian Finite Element (LFEM)

$$M \cong \left[\frac{1}{6}; \frac{2}{3}; \frac{1}{6}\right] \text{ and } A \cong [-1; 2; -1] \quad (48)$$

Equipped with the structure for both mass and stiffness matrix, we are now able to compute the semi-discrete dispersion relation for LFEM as follows. The matrices  $M$  (mass) and  $A$  (stiffness) are sparse; as they are tri-diagonal. Therefore, Proposition 10 leads to the following corresponding local operators  $\tilde{A}$  and  $\tilde{M}$  given via

$$\tilde{M}(\kappa) : \begin{array}{ccc} \mathbb{C}^{\mathbb{Z}} & & \longrightarrow \mathbb{C}^{\mathbb{Z}} \\ \xi \cdot (e^{i\kappa j \Delta x})_{j \in \mathbb{Z}} & \longmapsto & \frac{\xi}{3} \cos(\kappa \Delta x) + 2 (e^{i\kappa j \Delta x})_{j \in \mathbb{Z}} \end{array} \quad (49)$$

$$\tilde{A}(\kappa) : \begin{array}{ccc} \mathbb{C}^{\mathbb{Z}} & & \longrightarrow \mathbb{C}^{\mathbb{Z}} \\ \xi \cdot (e^{i\kappa j \Delta x})_{j \in \mathbb{Z}} & \longmapsto & \xi \cdot 4 \sin^2\left(\frac{\kappa \Delta x}{2}\right) (e^{i\kappa j \Delta x})_{j \in \mathbb{Z}} \end{array} \quad (50)$$

Hence, in terms of local operators on grid functions, we arrive at the following numerical dispersion relation

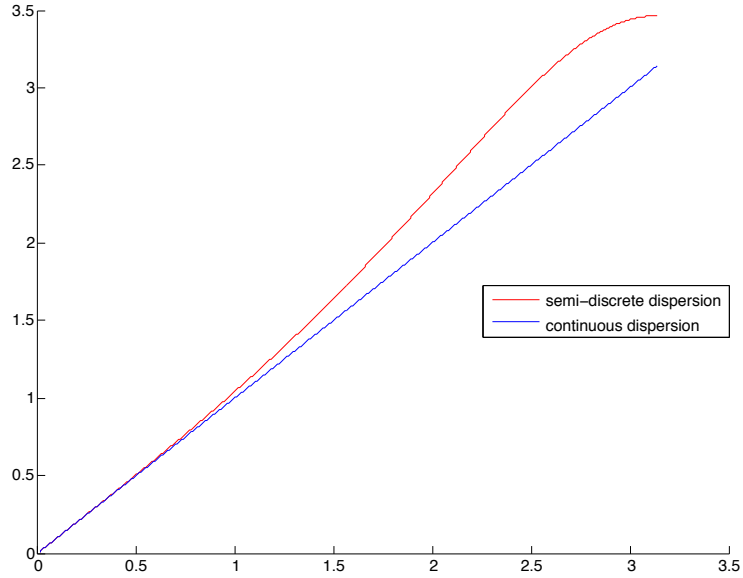
$$\left(\tilde{A}(\kappa) - c^2 \omega^2 \tilde{M}(\kappa)\right) \xi \cdot (e^{i\kappa j \Delta x})_{j \in \mathbb{Z}} \stackrel{!}{=} 0 \iff \left(\tilde{A}(\kappa) - c^2 \omega^2 \tilde{M}(\kappa)\right) \stackrel{!}{=} 0 \quad (51)$$

Now, substituting the term  $c^2 \omega^2$  for a  $\kappa$ -dependent term  $\lambda(\kappa)$ , we arrive at the following Generalized Eigenvalue Problem

$$\left(\tilde{A}(\kappa) - \lambda(\kappa) \cdot \tilde{M}(\kappa)\right) \stackrel{!}{=} 0 \quad (52)$$

This in turn results in the following semi-discrete dispersion relation for LFEM spatial discretization

$$\lambda(\kappa) = c^2 \omega^2 \iff \omega(\kappa) = \pm c \sqrt{|\lambda(\kappa)|} \quad (53)$$



Continuous dispersion relation (blue) and linear semi-discrete dispersion relation for  $\Delta x > 0$  and  $c=1$ . Both the x- and y-axis are scaled by  $\Delta x$ .

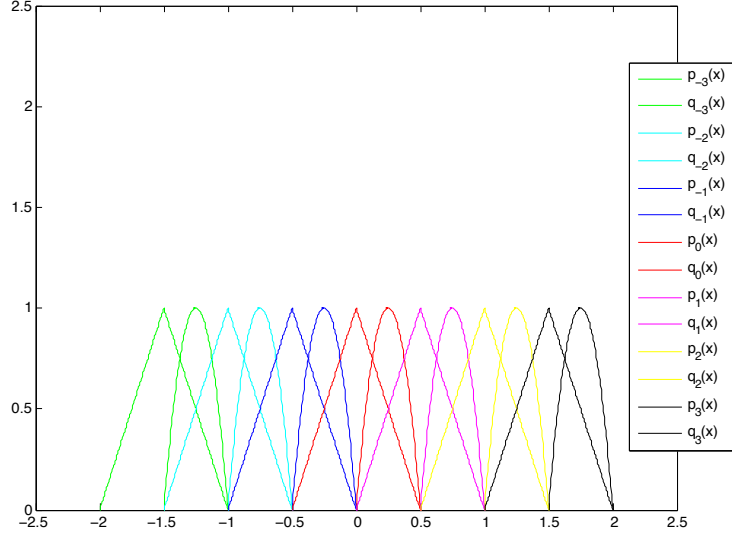
### 3.2 Quadratic Lagrangian Finite Elements

In order to perform a spatial discretization of Quadratic Lagrangian Finite Elements, we must consider quadratic basis polynomials in addition to our linear basis  $\mathcal{B}_1$ . To that effect, let us consider the polynomial  $q_0(x) \in C(\mathbb{R}, \mathbb{R})$  with the property that

$$\begin{aligned} \text{supp}(q_0) &\subseteq [0, \Delta x] \subseteq \mathbb{R} \\ q_0(x) &= 4 \frac{x(\Delta x - x)}{(\Delta x)^2} \mathbb{1}_{[0, \Delta x]} \end{aligned} \tag{54}$$

Then we may define  $\mathcal{B}_2$ , the set of basis functions up to quadratic degree as follows

$$\mathcal{B}_2 = \mathcal{B}_1 \cup \{ q_j(x) = q_0(x - j\Delta x) : j \in \mathbb{Z} \} \tag{55}$$



Plot of basis functions  $p_j(x), q_j(x)$  for  $j \in \{-3, -2, -1, 0, 1, 2, 3\}$

Performing the relevant computations for both Mass and Stiffness Matrix for the basis  $\mathcal{B}_1$ , we arrive at the following stencil for the both mass and stiffness matrix in the case of Linear Lagrangian Finite Element (QFEM)

$$A = \begin{pmatrix} [-1; 2; -1] & [0; 0] \\ [0; 0] & [0; \frac{16}{3}; 0] \end{pmatrix} \quad (56)$$

$$M = \begin{pmatrix} [\frac{1}{6}; \frac{2}{3}; \frac{1}{6}] & [\frac{1}{3}; \frac{1}{3}] \\ [\frac{1}{3}; \frac{1}{3}] & [0; \frac{8}{15}; 0] \end{pmatrix} \quad (57)$$

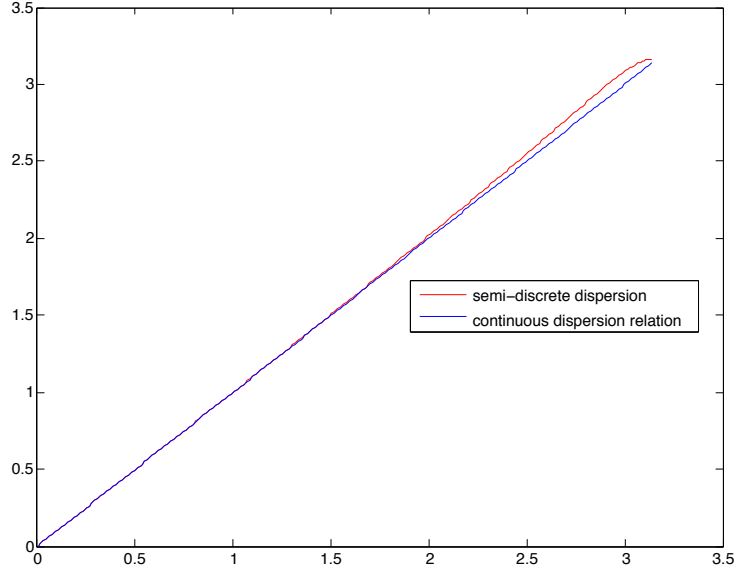
Therefore, Proposition 10 leads to the following corresponding local operators  $\tilde{A}$  and  $\tilde{M}$  given via

$$\tilde{M}(\kappa) : \quad (\mathbb{C}^2)^{\mathbb{Z}} \longrightarrow (\mathbb{C}^2)^{\mathbb{Z}} \\ \vec{\xi} \cdot (e^{i\kappa j \Delta x})_{j \in \mathbb{Z}} \longmapsto \frac{1}{3} \begin{pmatrix} (\cos(\kappa \Delta x) + 2) & (e^{-i\kappa \Delta x} + 1) \\ (1 + e^{i\kappa \Delta x}) & \frac{8}{5} \end{pmatrix} \vec{\xi} \cdot (e^{i\kappa j \Delta x})_{j \in \mathbb{Z}} \quad (58)$$

$$\tilde{A}(\kappa) : \quad (\mathbb{C}^2)^{\mathbb{Z}} \longrightarrow (\mathbb{C}^2)^{\mathbb{Z}} \\ \vec{\xi} \cdot (e^{i(\kappa j \Delta x - \omega t)})_{j \in \mathbb{Z}} \longmapsto \begin{pmatrix} 4 \sin^2(\frac{\kappa \Delta x}{2}) & 0 \\ 0 & \frac{16}{3} \end{pmatrix} \vec{\xi} \cdot (e^{i(\kappa j \Delta x - \omega t)})_{j \in \mathbb{Z}} \quad (59)$$

Analogous to the previous approach, we arrive at the following semi-discrete dispersion relation for QFEM spatial discretization

$$\lambda(\kappa) = c^2 \omega^2 \iff \omega(\kappa) = \pm c \sqrt{|\lambda(\kappa)|} \quad (60)$$



Continuous dispersion relation (blue) and quadratic semi-discrete dispersion relation for  $\Delta x > 0$  and  $c=1$ . Both the x- and y-axis are scaled by  $\Delta x$ .

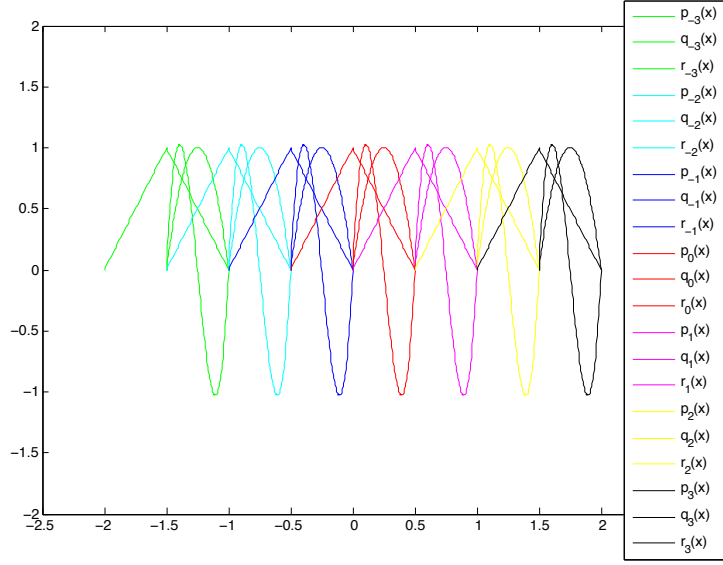
### 3.3 Cubic Lagrangian Finite Elements

In order to perform a spatial discretization of Cubic Lagrangian Finite Elements, we must consider cubic basis polynomials in addition to our basis of at most quadratic order  $\mathcal{B}_2$ . To that effect, let us consider the polynomial  $r_0(x) \in C(\mathbb{R}, \mathbb{R})$  with the property that

$$r_0(x) = \frac{3}{64(\Delta x)^3} x(x - \Delta x)\left(x - \frac{\Delta x}{2}\right) = \frac{3}{64(\Delta x)^3} \left(x^3 - \frac{3\Delta x}{2}x^2 + \frac{(\Delta x)^2}{2}x\right) \quad \text{supp}(r_0) \subseteq [0, \Delta x] \subseteq \mathbb{R} \quad (61)$$

Then we may define  $\mathcal{B}_3$ , the set of basis functions up to cubic degree as follows

$$\mathcal{B}_3 = \mathcal{B}_2 \cup \{ r_j(x) = r_0(x - j\Delta x) : j \in \mathbb{Z} \} \quad (62)$$



Plot of basis functions  $p_j(x), q_j(x), r_j(x)$  for  $j \in \{-3, -2, -1, 0, 1, 2, 3\}$

Performing the relevant computations for both Mass and Stiffness Matrix for the basis  $\mathcal{B}_1$ , we arrive at the following stencil for the both mass and stiffness matrix in the case of Linear Lagrangian Finite Element (QFEM)

$$A = \begin{pmatrix} [-1; 2; -1] & 0 & 0 \\ 0 & [0; \frac{16}{3}; 0] & 0 \\ 0 & 0 & [0; \frac{1024}{45}; 0] \end{pmatrix} \quad (63)$$

$$M = \begin{pmatrix} [\frac{1}{6}; \frac{2}{3}; \frac{1}{6}] & [\frac{1}{3}; \frac{1}{3}] & [-\frac{8}{45}; \frac{8}{45}] \\ [\frac{1}{3}; \frac{1}{3}] & [0; \frac{8}{15}; 0] & 0 \\ [\frac{8}{45}; -\frac{8}{45}] & 0 & [0; \frac{512}{945}; 0] \end{pmatrix} \quad (64)$$

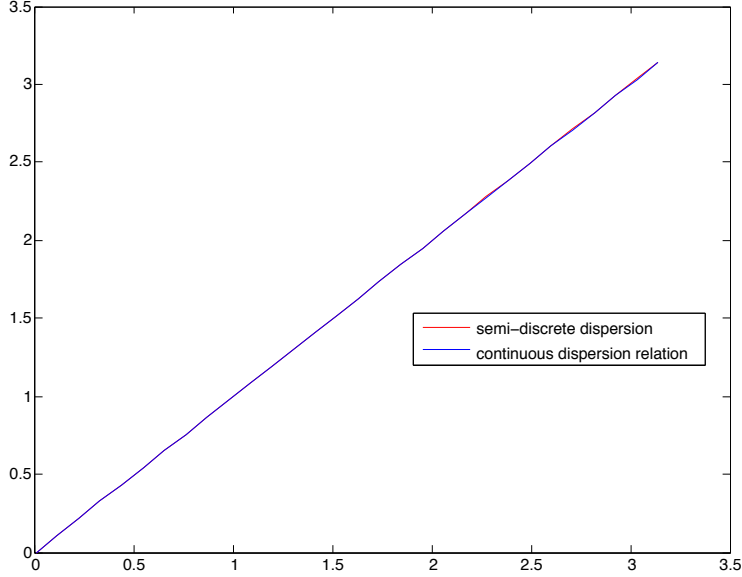
Therefore, Proposition 10 leads to the following corresponding local operators  $\tilde{A}$  and  $\tilde{M}$  given via

$$\tilde{M}(\kappa) : (\mathbb{C}^3)^{\mathbb{Z}} \longrightarrow (\mathbb{C}^3)^{\mathbb{Z}} \\ \vec{\xi} \cdot (e^{i(\kappa j \Delta x - \omega t)})_{j \in \mathbb{Z}} \longmapsto \begin{pmatrix} \frac{1}{3}(\cos(\kappa \Delta x) + 2) & \frac{1}{3}(e^{-i\kappa \Delta x} + 1) & \frac{8}{45}(1 - e^{-i\kappa \Delta x}) \\ \frac{1}{\frac{8}{45}}(e^{i\kappa \Delta x} + 1) & \frac{8}{15} & 0 \\ \frac{8}{45}(1 - e^{i\kappa \Delta x}) & 0 & \frac{512}{945} \end{pmatrix} \vec{\xi} \cdot (e^{i(\kappa j \Delta x - \omega t)})_{j \in \mathbb{Z}} \quad (65)$$

$$\tilde{A}(\kappa) : (\mathbb{C}^3)^{\mathbb{Z}} \longrightarrow (\mathbb{C}^3)^{\mathbb{Z}} \\ \vec{\xi} \cdot (e^{i(\kappa j \Delta x - \omega t)})_{j \in \mathbb{Z}} \longmapsto \begin{pmatrix} 4 \sin^2(\frac{\kappa \Delta x}{2}) & 0 & 0 \\ 0 & \frac{16}{3} & 0 \\ 0 & 0 & \frac{1024}{45} \end{pmatrix} \vec{\xi} \cdot (e^{i(\kappa j \Delta x - \omega t)})_{j \in \mathbb{Z}} \quad (66)$$

Analogous to the previous approach, we arrive at the following semi-discrete dispersion relation for CFEM spatial discretization

$$\lambda(\kappa) = c^2 \omega^2 \iff \omega(\kappa) = \pm c \sqrt{|\lambda(\kappa)|} \quad (67)$$



Continuous dispersion relation (blue) and linear semi-discrete dispersion relation for  $\Delta x > 0$  and  $c=1$ . Both the x- and y-axis are scaled by  $\Delta x$ .

## 4 Time-stepping

The following section is dedicated to a concise introduction of all symplectic integrators relevant to this term paper. We will encounter two classical examples of symplectic integrators **Leapfrog** integration and **Semi-implicit Euler** integration. By means of the semi-implicit Euler method, we will then construct higher order integrators that achieve order  $\mathcal{O}(t^2)$ ,  $\mathcal{O}(t^3)$ ,  $\mathcal{O}(t^4)$  respectively. Throughout each discussion, we will begin an initial inspection into the dispersion relation for each individual scheme applied to one of the semi-discrete schemes in the previous section.

### 4.1 Leapfrog integration

Leapfrog integration is a classical two-step method. For an arbitrary second-order ODE of the form  $\ddot{u} = f(u)$ , the numerical approximation  $u_n$  of the continuous solution  $u$  according to Leapfrog integration is given via

$$\frac{u^{n+1} - 2u^n + u^{n-1}}{(\Delta t)^2} = f(u_n) \iff u^{n+1} = f(u_n)(\Delta t)^2 + 2u^n - u^{n-1} \quad (68)$$

The Leapfrog time-step for a Lagrangian Finite Element discretization with mass and stiffness matrix  $\mathbf{A}, \mathbf{M}$  respectively is given via

$$\mathbf{M} \frac{\mu^{n+1} - 2\mu^n + \mu^{n-1}}{(\Delta t)^2} = -c^2 \mathbf{A} \mu_n \quad (69)$$

Consider a spatial discretization of order  $p = 1, 2$  or  $3$  according to the Lagrangian Finite Elements computed in the previous chapter. Recall the operators  $\tilde{\mathbf{M}}, \tilde{\mathbf{A}}$ , which have the property that for all

$\kappa \in \mathbb{R}$ , there exist local operators  $\tilde{M}(\kappa), \tilde{A}(\kappa)$ , s.t. for all  $\vec{\xi} \in \mathbb{C}^p$

$$\mathbf{M} \left( \vec{\xi} \cdot (e^{i\kappa j \Delta x})_{j \in \mathbb{N}} \right) = \left( \tilde{\mathbf{M}}(\kappa) \left( \vec{\xi} \right) \right) \cdot (e^{i\kappa j \Delta x})_{j \in \mathbb{N}} \quad (70)$$

$$\mathbf{A} \left( \vec{\xi} \cdot (e^{i\kappa j \Delta x})_{j \in \mathbb{N}} \right) = \left( \tilde{\mathbf{A}}(\kappa) \left( \vec{\xi} \right) \right) \cdot (e^{i\kappa j \Delta x})_{j \in \mathbb{N}} \quad (71)$$

Recall that from the property (7), we may defer that the eigenfunctions of the temporal differences scheme  $\frac{u^{n+1} - 2u^n + u^{n-1}}{(\Delta t)^2} = f(u^n)$  is given by the exponentials functions,  $\omega \in \mathbb{R} \rightarrow (e^{i\omega \Delta t}) \in \mathbb{C}$ . By rearranging the terms in the temporal differences scheme and for  $n \in \mathbb{N}$  by plugging in  $\mu^n := (e^{i(\kappa j \Delta x - \omega n \Delta t)})_{j \in \mathbb{Z}}$  for  $u_n$ , we arrive at:

$$\mu^{(n+1)} + \mu^{(n-1)} = 2\mu^n + (\Delta t)^2 f(\mu^n) \iff 2 \cos(\omega \Delta t) \cdot \mu^n = 2\mu^n + (\Delta t)^2 f(\mu^n) \quad (72)$$

Hence, writing the above equation in terms of our local analogs  $\tilde{\mathbf{M}}(\kappa), \tilde{\mathbf{A}}(\kappa)$ , we may conclude that for all  $\vec{\xi} \in \mathbb{C}^p$

$$\begin{aligned} (\cos(\omega \Delta t) - 1) \tilde{\mathbf{M}}(\kappa) \left( \vec{\xi} \right) &= -\frac{(\Delta t)^2 c^2}{2} \tilde{\mathbf{A}}(\kappa) \left( \vec{\xi} \right) \\ &\iff \\ \left( \sin\left(\frac{\omega \Delta t}{2}\right)^2 \right) \tilde{\mathbf{M}}(\kappa) \left( \vec{\xi} \right) &= \frac{(\Delta t)^2 c^2}{4} \tilde{\mathbf{A}}(\kappa) \left( \vec{\xi} \right) \end{aligned} \quad (73)$$

Hence, recalling the setting of the previous section, let  $\lambda : \mathbb{R} \mapsto \mathbb{R}$  with the property that for all  $\kappa \in \mathbb{R}$  it holds that for all  $\vec{\xi} \in \mathbb{C}^p$

$$\left[ \frac{c^2}{2} \tilde{\mathbf{A}}(\kappa) - \lambda(\kappa) \left( -\tilde{\mathbf{M}}(\kappa) \right) \right] \vec{\xi} \stackrel{!}{=} 0 \quad (74)$$

Then the (numerical) dispersion relation for the above (74) is given via

$$\cos(\omega \Delta t) = \lambda(\kappa) + 1 \quad (75)$$

Or, equivalently for  $\bar{\lambda} : \mathbb{R} \mapsto \mathbb{R}$  with the property that for all  $\kappa \in \mathbb{R}$  it holds that for all  $\vec{\xi} \in \mathbb{C}^p$

$$\left[ \frac{c^2}{4} \tilde{\mathbf{A}}(\kappa) - \bar{\lambda}(\kappa) \cdot \tilde{\mathbf{M}}(\kappa) \right] \vec{\xi} \stackrel{!}{=} 0 \quad (76)$$

Then the (numerical) dispersion relation for the above (76) is given via

$$\sin^2 \left( \frac{\omega \Delta t}{2} \right) = \bar{\lambda}(\kappa) \quad (77)$$

## 4.2 Symplectic Euler

We will introduce the symplectic Euler numerical integration scheme first in a general setting and then apply it to the ODE (45) in order to illustrate the proper method for determining the dispersion relation in a Lagrangian Finite Element Method spatial discretization and symplectic time-stepping. The semi-implicit Euler method is a partitioned method, which is most aptly described according to

the theory of Hamiltonians corresponding to a dynamic system. We will omit the formal introduction via Hamiltonians and refer to [1] for a more in-depth approach to the definition of symplecticity and its connection to solving partitioned Hamiltonians.

Let us begin by introducing one variation of the semi-implicit Euler method. To that effect, consider  $\mu : \mathbb{R} \times [0, T] \rightarrow \mathbb{R}$  and  $\nu : \mathbb{R} \times [0, T] \rightarrow \mathbb{R}$  with the property that for all  $(x, s) \in \mathbb{R} \times [0, T]$  it holds that

$$\frac{\partial}{\partial t} \mu(x, s) = \dot{\mu}(x, s) = \nu(x, s) \quad (78)$$

Let  $f_\mu : \mathbb{R} \rightarrow \mathbb{R}$  and  $f_\nu : \mathbb{R} \rightarrow \mathbb{R}$  bearbitrary functions that satisfy the property that for all  $(x, s) \in \mathbb{R} \times [0, T]$

$$\frac{\partial}{\partial t} \mu(x, s) = f_\mu(\nu(x, s)) \text{ and } \frac{\partial}{\partial t} \nu(x, s) = f_\nu(\mu(x, s)) \quad (79)$$

For all  $(x, s) \in \mathbb{R} \times [0, T]$  let us define  $\zeta(x, s) = \begin{pmatrix} \mu(x, s) \\ \nu(x, s) \end{pmatrix}$ . Hence, we arrive at the following ODE

$$\frac{\partial}{\partial t} \zeta(x, s) = \begin{pmatrix} f_\mu(\nu(x, s)) \\ f_\nu(\mu(x, s)) \end{pmatrix} \quad (80)$$

The **symplectic Euler** method applied to equation (80) provides an update scheme given via

$$\zeta^{n+1} = \begin{pmatrix} \mu^{n+1} \\ \nu^{n+1} \end{pmatrix} = \begin{pmatrix} \mu^n \\ \nu^n \end{pmatrix} + \Delta t \begin{pmatrix} f_\mu(\nu^{n+1}) \\ f_\nu(\mu^n) \end{pmatrix} \quad (81)$$

We may rewrite the above equation via

$$\begin{pmatrix} \mu^{n+1} \\ \nu^{n+1} \end{pmatrix} - \Delta t \begin{pmatrix} f_\mu(\nu^{n+1}) \\ 0 \end{pmatrix} = \begin{pmatrix} \mu^n \\ \nu^n \end{pmatrix} + \Delta t \begin{pmatrix} 0 \\ f_\nu(\mu^n) \end{pmatrix} \quad (82)$$

**Proposition 14** (Stability Function of the semi-implicit Euler method). *The semi-implicit Euler method applied to the ODE (80) is given via*

$$\Psi(z) = \left( \begin{pmatrix} \mathbb{1} & 0 \\ 0 & \mathbb{1} \end{pmatrix} - z \cdot \begin{pmatrix} 0 & \mathbb{1} \\ 0 & 0 \end{pmatrix} \right)^{-1} \cdot \left( \begin{pmatrix} \mathbb{1} & 0 \\ 0 & \mathbb{1} \end{pmatrix} + z \cdot \begin{pmatrix} 0 & 0 \\ \mathbb{1} & 0 \end{pmatrix} \right) \quad (83)$$

*Proof of Proposition 14.* **Ansatz** Apply the above scheme to the following ODE

$$\begin{pmatrix} \dot{\mu} \\ \dot{\nu} \end{pmatrix} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} \mu \\ \nu \end{pmatrix} \quad (84)$$

Hence, if we apply the symplectic Euler to the above ODE, we arrive at the following update scheme:

$$\begin{pmatrix} \mu^{n+1} \\ \nu^{n+1} \end{pmatrix} = \begin{pmatrix} \mu^n \\ \nu^n \end{pmatrix} + (\Delta t) \begin{pmatrix} 0 & \lambda_2 \\ \lambda_1 & 0 \end{pmatrix} \begin{pmatrix} \mu^n \\ \nu^{n+1} \end{pmatrix} \quad (85)$$

$$\left( \begin{pmatrix} \mathbb{1} & 0 \\ 0 & \mathbb{1} \end{pmatrix} - \underbrace{(\Delta t)\lambda_2}_{=z} \begin{pmatrix} 0 & \mathbb{1} \\ 0 & 0 \end{pmatrix} \right) \begin{pmatrix} \mu^{n+1} \\ \nu^{n+1} \end{pmatrix} = \left( \begin{pmatrix} \mathbb{1} & 0 \\ 0 & \mathbb{1} \end{pmatrix} + \underbrace{(\Delta t)\lambda_1}_{=z} \begin{pmatrix} 0 & 0 \\ \mathbb{1} & 0 \end{pmatrix} \right) \begin{pmatrix} \mu^n \\ \nu^n \end{pmatrix} \quad (86)$$



$$\begin{pmatrix} \mu^{n+1} \\ \nu^{n+1} \end{pmatrix} = \left( \begin{pmatrix} \mathbb{1} & 0 \\ 0 & \mathbb{1} \end{pmatrix} - \underbrace{(\Delta t)\lambda_2}_{=z} \begin{pmatrix} 0 & \mathbb{1} \\ 0 & 0 \end{pmatrix} \right)^{-1} \left( \begin{pmatrix} \mathbb{1} & 0 \\ 0 & \mathbb{1} \end{pmatrix} + \underbrace{(\Delta t)\lambda_1}_{=z} \begin{pmatrix} 0 & 0 \\ \mathbb{1} & 0 \end{pmatrix} \right) \begin{pmatrix} \mu^n \\ \nu^n \end{pmatrix} \quad (87)$$

This completes the proof of Proposition 14.  $\square$

Finally, combining the above with the abstract process of finding the numerical dispersion relation for a general single-step method with arbitrary stability function according to equation (37), we may determine the **numerical dispersion relation** of the symplectic Euler method applied to arbitrary ODEs of the form (80). It also turns out that this framework allows us to determine the numerical dispersion relation for the linear ODE (45); the numerical approximation of the wave equation (3) utilizing Lagrangian Finite Elements in space and semi-implicit Euler time-stepping.

To that effect, recall the spatial discretizations introduced in Section 3 and let  $\mathbf{M}$ ,  $\mathbf{A}$  be the mass and stiffness matrix respectively. Then it holds that for all  $s \in [0, T]$

$$\mathbf{M} \frac{\partial^2 \vec{\mu}}{\partial t^2}(s) + c^2 \mathbf{A} \vec{\mu}(s) = 0 \Rightarrow \underbrace{\begin{pmatrix} Id_{\mathbb{R}^z} & 0 \\ 0 & \mathbf{M} \end{pmatrix}}_{\overline{\mathbf{M}}} \frac{\partial \vec{\zeta}}{\partial t}(s) = \underbrace{\begin{pmatrix} 0 & Id_{\mathbb{R}^z} \\ -A & 0 \end{pmatrix}}_{\overline{\mathbf{A}}} \vec{\zeta} \quad (88)$$

Hence, we may rewrite the linear ODE (45) in terms of the extended matrices  $\overline{\mathbf{M}}, \overline{\mathbf{A}}$  as a first-order ODE

$$\frac{\partial \vec{\zeta}}{\partial t}(s) = \overline{\mathbf{M}}^{-1} \overline{\mathbf{A}} \vec{\zeta} \quad (89)$$

Finally, applying the symplectic Euler method to the above ODE (cf. (81)) yields

$$\vec{\zeta}^{n+1} = \left( \Psi(\overline{\mathbf{M}}^{-1} \overline{\mathbf{A}}) \right) \vec{\zeta}^n = \left( \overline{\mathbf{M}} - (\Delta t) \begin{pmatrix} 0 & Id_{\mathbb{C}^z} \\ 0 & 0 \end{pmatrix} \right)^{-1} \left( \overline{\mathbf{M}} + (\Delta t) \begin{pmatrix} 0 & 0 \\ -A & 0 \end{pmatrix} \right) \begin{pmatrix} \mu^n \\ \nu^n \end{pmatrix} \quad (90)$$

Or, equivalently

$$\left( \overline{\mathbf{M}} - (\Delta t) \begin{pmatrix} 0 & Id_{\mathbb{C}^z} \\ 0 & 0 \end{pmatrix} \right) \vec{\zeta}^{n+1} = \left( \overline{\mathbf{M}} + (\Delta t) \begin{pmatrix} 0 & 0 \\ -A & 0 \end{pmatrix} \right) \vec{\zeta}^n \quad (91)$$

Writing everything the above in terms of the local operators associated to the Lattice Operator, we may conclude that for arbitrary  $\kappa \in \mathbb{R}$  and  $\lambda : \kappa \in \mathbb{R} \rightarrow \lambda(\kappa) \in \mathbb{C}$  such that for all  $\xi \in \mathbb{C}^p$  it holds that

$$\left[ \begin{pmatrix} \mathbb{1}_{\mathbb{C}^p} & 0 \\ -\Delta t \tilde{A}(\kappa) & \tilde{\mathbf{M}}(\kappa) \end{pmatrix} - \lambda(\kappa) \cdot \begin{pmatrix} \mathbb{1}_{\mathbb{C}^p} & -(\Delta t) \mathbb{1}_{\mathbb{C}^p} \\ 0 & \tilde{\mathbf{M}}(\kappa) \end{pmatrix} \right] \xi \stackrel{!}{=} 0 \quad (92)$$

Then the **numerical dispersion relation** for the semi-implicit Euler method coupled with Lagrangian Finite Element discretization in space is given via

$$e^{i\omega\Delta t} = \lambda(\kappa) \iff \omega = \Re \left( \frac{\log(\lambda(\kappa))}{i\Delta t} \right) \quad (93)$$

### 4.3 Higher order symplectic integrators

In the previous subsection, we introduced the simplest of all single-step symplectic integrators; the semi-implicit Euler method. This will be our main tool when constructing higher order symplectic integrators. The process of constructing higher order symplectic integrators rests on calculating the proper coefficients according to the Baker-Campbell-Hausdorff formula. Once these coefficients are determined, the entire algorithm of arbitrary higher order runs like an iterated (weighted) semi-implicit Euler method. Therefore, let us begin by defining the weights for these higher-order schemes.

We have already seen the **first-order coefficients** to be  $a = a_1 = 1$  and  $b = b_1 = 1$

The **second-order coefficients** are calculated to be  $a = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$  and  $b = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

The **third-order coefficients** are calculated to be  $a = \begin{pmatrix} 2/3 \\ -2/3 \\ 1 \end{pmatrix}$  and  $b = \begin{pmatrix} 7/24 \\ 3/4 \\ -1/24 \end{pmatrix}$

The **fourth-order coefficients** are calculated to be  $a = \begin{pmatrix} \frac{2+2^{1/3}+2^{-1/3}}{6} \\ \frac{1-2^{1/3}-2^{-1/3}}{6} \\ \frac{1-2^{1/3}-2^{-1/3}}{6} \\ \frac{2+2^{1/3}+2^{-1/3}}{6} \end{pmatrix}$  and  $b = \begin{pmatrix} 0 \\ \frac{1}{2-2^{1/3}} \\ \frac{1}{2-2^{2/3}} \\ \frac{1}{2-2^{1/3}} \end{pmatrix}$

**Definition 15** (Definition of symplectic time-stepping algorithms of arbitrary order). *Let  $d \in \mathbb{N}, d > 1$ . Then the symplectic time-stepping algorithm applied to the Hamiltonian ODE*

$$\frac{\partial}{\partial t} \begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} \dot{p} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} F(q, t) \\ G(p) \end{pmatrix} \quad (94)$$

is given via the following algorithm

#### Symplectic time-stepping algorithm

Input: order =  $d > 1$ ,  $\Delta t > 0$ ,  $F : \mathbb{R}^Z \rightarrow \mathbb{R}^Z$ ,  $G : \mathbb{R}^Z \rightarrow \mathbb{R}^Z$ ,  $F_0, G_0 \in \mathbb{R}^Z$  and  $T > 0$

Output:  $p(T), q(T)$ , the inertia and displacement vectors at time  $T > 0$

$p_1 = F_0, q_1 = G_0$

$N = \text{ceil}(T/\Delta t)$

for  $i = 1$  to  $N$  do

  % Begin with integration method update

$p_{in} = p_i$

$q_{in} = q_i$

    for  $j = 1$  to  $d$  do

$p_{out} = p_{in} + b_j \Delta t \cdot F(q_{in})$

$q_{out} = q_{in} + a_j \Delta t \cdot G(p_{out})$

$p_{in} = p_{out}$

$q_{in} = q_{out}$

    end

$p_{i+1} = p_{out}$

$q_{i+1} = q_{out}$   
 end  
**return**  $p_N, q_N$

**Proposition 16.** *Let  $d \in \{1, 2, 3, 4\}$ . Then the stability function  $\Psi_d : \mathbb{C} \rightarrow \mathbb{C}^{2 \times 2}$  corresponding to the symplectic time-stepping algorithm of order  $d$  is given via the property that for all  $z \in \mathbb{C}$  it holds that*

$$\Psi_d(z) = \prod_{j=1}^d \Psi^j(z) \quad (95)$$

Where  $\Psi^j(z)$  is given as a "weighted" version of the stability function of the semi-implicit Euler method, with weights  $a_j$  and  $b_j$  respectively; specifically, for all  $z \in \mathbb{C}$

$$\Psi^j(z) = \left( \begin{pmatrix} \mathbb{1} & 0 \\ 0 & \mathbb{1} \end{pmatrix} - a_j z \cdot \begin{pmatrix} 0 & \mathbb{1} \\ 0 & 0 \end{pmatrix} \right)^{-1} \cdot \left( \begin{pmatrix} \mathbb{1} & 0 \\ 0 & \mathbb{1} \end{pmatrix} + b_j z \cdot \begin{pmatrix} 0 & 0 \\ \mathbb{1} & 0 \end{pmatrix} \right) \quad (96)$$

Hence, the stability function of the symplectic integrator of order  $d$  is given via

$$\prod_{j=1}^d \left( \begin{pmatrix} \mathbb{1} & 0 \\ 0 & \mathbb{1} \end{pmatrix} - a_j z \cdot \begin{pmatrix} 0 & \mathbb{1} \\ 0 & 0 \end{pmatrix} \right)^{-1} \cdot \left( \begin{pmatrix} \mathbb{1} & 0 \\ 0 & \mathbb{1} \end{pmatrix} + b_j z \cdot \begin{pmatrix} 0 & 0 \\ \mathbb{1} & 0 \end{pmatrix} \right) \quad (97)$$

Furthermore, assume that for all  $n \in \{1, \dots, N\}$ ,  $\vec{\zeta}^n$  the discrete evolution of the higher order symplectic integrator (of order  $d$ ) applied to the linear ODE (45). Then the update scheme for all  $n \in \{1, \dots, N\}$  for this fully-discrete evolution is given via

$$\vec{\zeta}^{n+1} = \Psi_d(\overline{\mathbf{M}}^{-1} \overline{\mathbf{A}}) \vec{\zeta}^n \quad (98)$$

Hence, recalling the setting in section 2, we may may the above equation in terms of local operators on lattice vectors. Hence for all eigenfunction  $\lambda: [-\pi, \pi] \rightarrow \mathbb{C}$  and all  $\kappa \in \mathbb{R}$  it holds that

$$\left[ \prod_{j=1}^d \left[ \begin{pmatrix} \mathbb{1}_{\mathbb{C}^p} & -(\Delta t) a_j \mathbb{1}_{\mathbb{C}^p} \\ 0 & \tilde{\mathbf{M}}(\kappa) \end{pmatrix}^{-1} \begin{pmatrix} \mathbb{1}_{\mathbb{C}^p} & 0 \\ -\Delta t b_j \tilde{\mathbf{A}}(\kappa) & \tilde{\mathbf{M}}(\kappa) \end{pmatrix} - \lambda(\kappa) \cdot \begin{pmatrix} \mathbb{1}_{\mathbb{C}^p} & 0 \\ 0 & \mathbb{1}_{\mathbb{C}^p} \end{pmatrix} \right] \vec{\xi} \stackrel{!}{=} 0 \quad (99)$$

**Remark 17.** Notice that for  $d = 1$ , the equation above coincides with the equation derived in the case of the semi-implicit Euler method.

Solving this Generalized Eigenvalue Problem leads to the **numerical dispersion relation**

$$\omega(\kappa) = \left( \frac{\log(\lambda(\kappa))}{i\Delta t} \right) \quad (100)$$

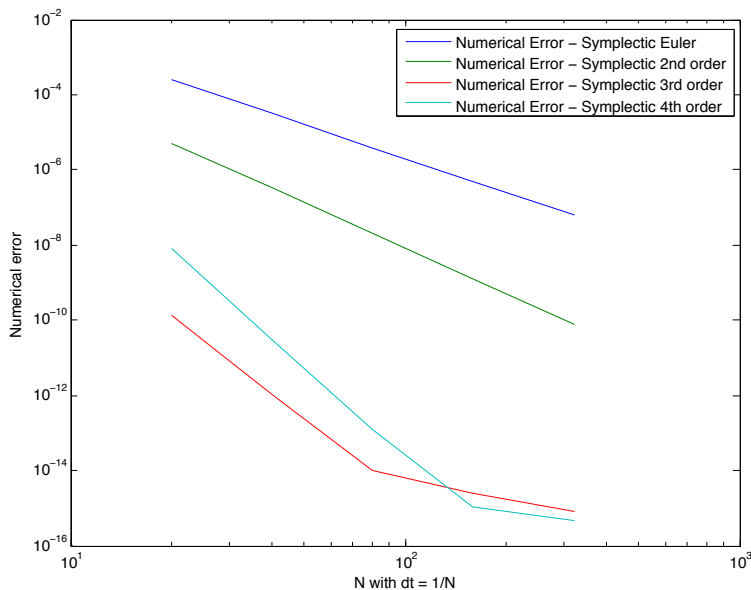
#### 4.4 Empirical Convergence Rates for symplectic time-stepping algorithms

It is a well-known fact that Leapfrog integration provides a second-order convergence rate and is therefore of consistency order  $\mathcal{O}(\Delta t^2)$ . In this section, we will provide empirical convergence rates

for the symplectic time-stepping algorithms introduced in the previous two sections. To that effect, we will apply the symplectic time-stepping algorithms to the model problem for the ODE

$$u''(x) = -u(x) \tag{101}$$

The codes utilized for the empirical convergence analysis are readily available in the Appendix (c.f. section 7.2). Therein, we have applied the symplectic time-stepping algorithms to the above model problem ODE with the **initial value problem**  $u(0) = 1, u'(0) = 0$ . Applying the codes in section 7.2, we arrive at the following empirical convergence rates for  $\Delta t = 0.1 \cdot 2^j$  for  $j \in \{-1, \dots, -5\}$



Empirical Numerical Error of symplectic time-stepping applied to the model problem  $u''(x) = -u(x)$  with initial value problem  $u(0) = 1$  and  $u'(0) = 0$ .

Clearly in the above, we may observe the expected behavior. The curve of least descent (blue) belongs to that of the symplectic Euler algorithm, which provides the convergence order  $\mathcal{O}(\Delta t)$ . The green, red and cyan curves each have slope greater than that belonging to the symplectic Euler algorithm, which clearly indicates a higher convergence order. Clearly, the green curve belonging to that of the second-order symplectic time-stepping algorithm has the greatest slope of the aforementioned three, which clearly indicates the convergence order  $\mathcal{O}(\Delta t^2)$ . Correspondingly, the red curve belonging to the third-order symplectic time-stepping algorithm exhibits a greater slope than the cyan curve and therefore leads to convergence order  $\mathcal{O}(\Delta t^3)$ . Finally, the third curve belonging to the fourth order symplectic time-stepping algorithm has the steepest descent and therefore provides convergence order  $\mathcal{O}(\Delta t^4)$ .

## 5 Dispersion Relation of the fully discrete evolution problem

### 5.1 Leapfrog integration

Our first analysis will involve Lagrangian Finite Element spatial discretization of Linear, Quadratic and Cubic order coupled with a simple Leapfrog integration. To that extent, let us determine the generalized eigenvalue problem for Leapfrog integration.

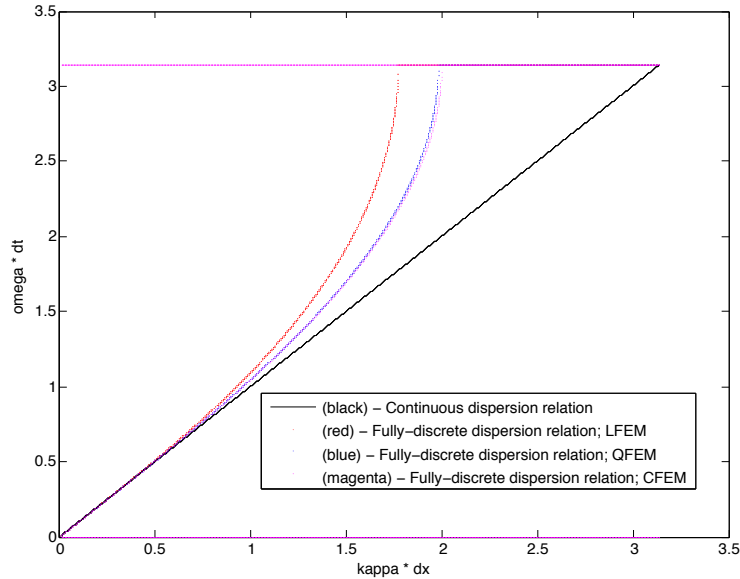
**Definition 18** (Leapfrog and Lagrangian Finite Element discrete evolution problem). *Consider the Lagrangian Finite Element spatial discretization of arbitrary degree  $p$  applied to the PDE (3) on an equidistant spatial grid of width  $\Delta x > 0$ . Let  $\mathbf{M}$  and  $\mathbf{A}$  the corresponding mass and stiffness matrix, recall the corresponding linear ODE (45). Let  $\Delta t > 0$  and  $\mathfrak{G} : V_x^{\mathbb{Z}} \rightarrow V_x^{\mathbb{Z}}$  be the Galerkin Matrix with the property that for all  $n \in \mathbb{N}$  it holds that for all  $n \in \mathbb{N}$  it holds that  $\vec{\mu}^{(n+1)} = \mathfrak{G}\vec{\mu}^{(n)}$ . Then*

$$\mathbf{M} \frac{\vec{\mu}^{(n+1)} - 2\vec{\mu}^{(n)} + \vec{\mu}^{(n-1)}}{(\Delta t)^2} = \mathbf{A}\vec{\mu}^{(n)} \quad (102)$$

**Proposition 19.** *Let  $\Delta x, \Delta t > 0$  and let  $\mathbf{A}, \mathbf{M} : \mathbb{R}^{\mathbb{Z}} \rightarrow \mathbb{R}^{\mathbb{Z}}$  be the mass and stiffness matrix corresponding to the Lagrangian Finite Element spatial discretization of the PDE (3) on an equidistant spatial grid of width  $\Delta x > 0$ . Applying Leapfrog integration to the ODE (45), we arrive at the numerical dispersion relation for Leapfrog time-stepping as the mapping  $\omega : \kappa \in \mathbb{R} \rightarrow \omega(\kappa) \in \mathcal{P}(\mathbb{R})$  s.t. for all  $\lambda : \kappa \in \mathbb{R} \rightarrow \lambda(\kappa) \in \mathbb{R}$  with property (36) and for all  $\kappa \in \mathbb{R}$  there exists  $\omega \in \omega(\kappa) \subseteq \mathbb{R}$ , s.t.*

$$2 \cos(\omega \Delta t) - 1 = \lambda(\kappa) \quad (103)$$

Finally, we are able to provide the first result for the numerical dispersion relation of LFEM, QFEM and CFEM coupled with Leapfrog integration.



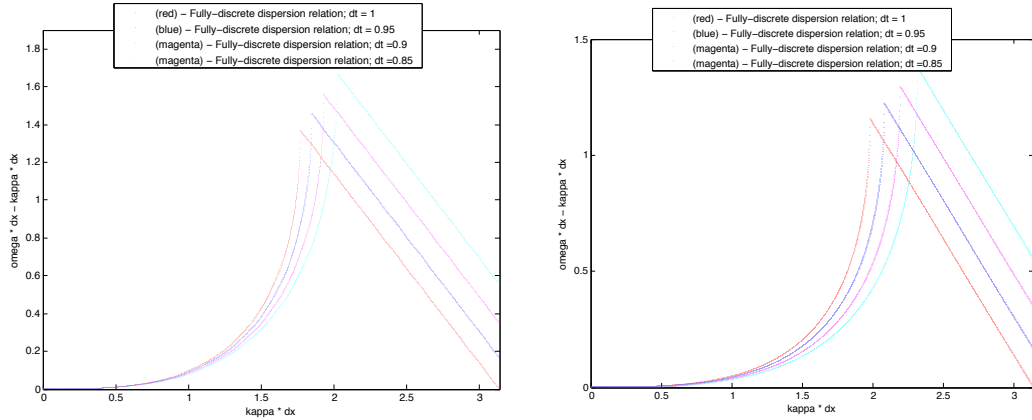
Continuous dispersion relation (black) and fully-discrete dispersion relation LFEM (red) , QFEM (blue), CFEM (magenta) together with Leapfrog time-stepping. The x-axis is scaled by  $\Delta x > 0$  and the y-axis is scaled by  $\Delta t > 0$ .

Now that we have determined the dispersion relation for fixed  $\Delta x, \Delta t > 0$ , we would like to discuss the asymptotics as  $\Delta x, \Delta t \rightarrow 0$ . To that extent, it is a poignant remark to note that in the figure above the x-axis is scaled by  $\Delta x > 0$  and the y-axis is scaled by  $\Delta t > 0$ . Furthermore, let us define  $\alpha = \frac{\Delta t}{\Delta x}$ , and consider fixed  $\Delta x > 0$ .

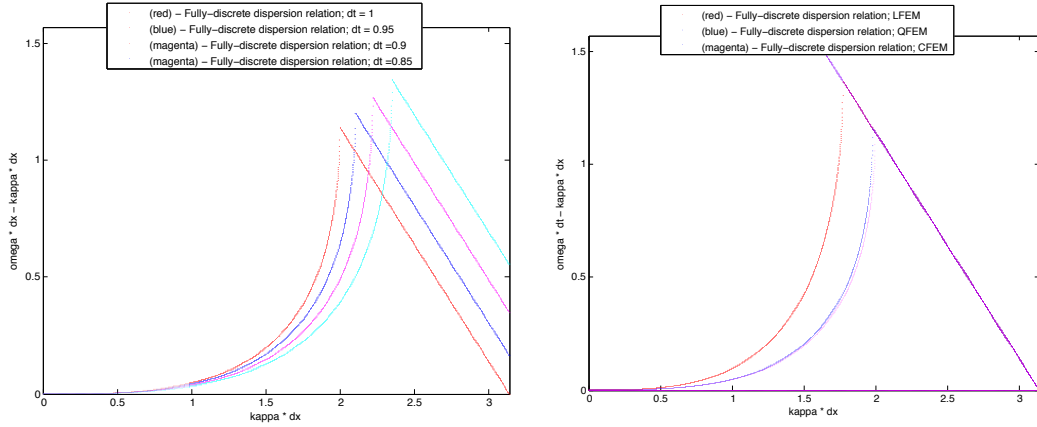
**$\alpha$  constant:** In the case where  $\alpha$  is held constant, the dispersion relation remains self-similar on the interval  $[0, \frac{\pi}{\Delta x}]$ . This has been proven rigorously in a separate paper however can be deduced readily by substituting  $\tilde{\kappa} \cong \kappa \Delta x$  and  $\tilde{\omega} \cong \omega \Delta t$  into the relevant equations when computing the dispersion relation. By means of this substitution, the dispersion relation translates into a simple equation that relates an arbitrary  $\kappa \in \mathbb{R}$  with a corresponding set of  $\omega(\kappa) \subseteq \mathbb{R}$

$\alpha \rightarrow 0$ : Let  $\Delta x > 0$  and consider  $\alpha \rightarrow 0$ , or equivalently  $\Delta t \rightarrow 0$ . Numerical experiments have validated that as  $\Delta t \rightarrow 0$ , the numerical dispersion relation of the fully-discrete scheme converges to the semi-discrete dispersion relation determined in section 3.

**Remark 20.** Observe in the figure above that as we increase the order of the Lagrangian Finite Element Method, we may observe a significant reduction in the numerical dispersion introduced by the fully-discrete evolution problem. In order to gauge the quantitative reduction in numerical dispersion, consider the following error estimates

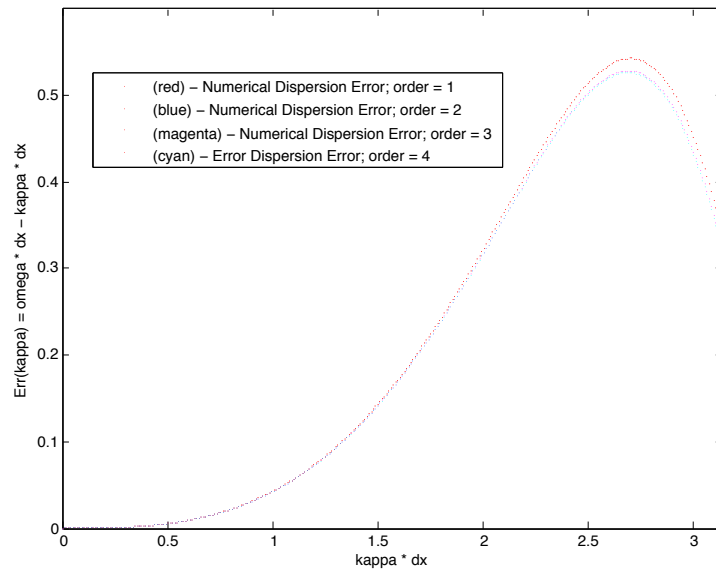


Numerical Dispersion Relation Error  $[Err(\kappa) = \omega(\kappa) - \kappa]$  for LFEM (left) and QFEM (right).



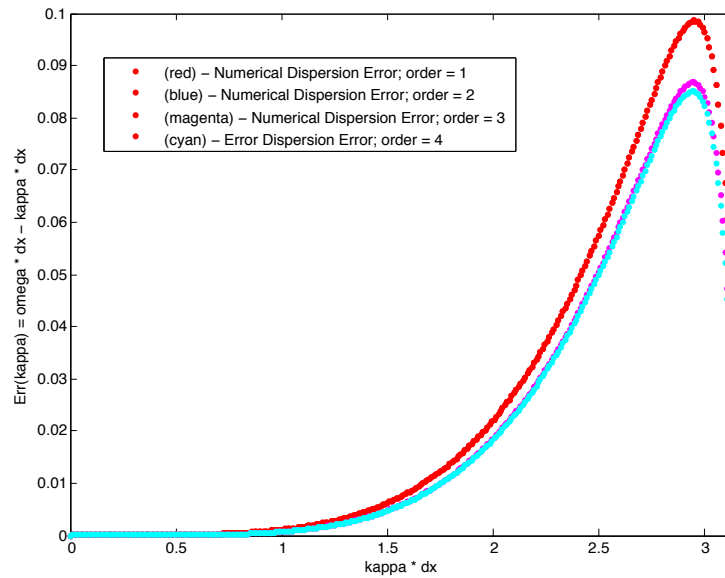
Numerical Dispersion Relation Error [ $Err(\kappa) = \omega(\kappa) - \kappa$ ] for CFEM (left) and combined LFEM, QFEM and CFEM for a single  $\Delta t > 0$  (right).

## 5.2 Comparison - LFEM with symplectic integrators



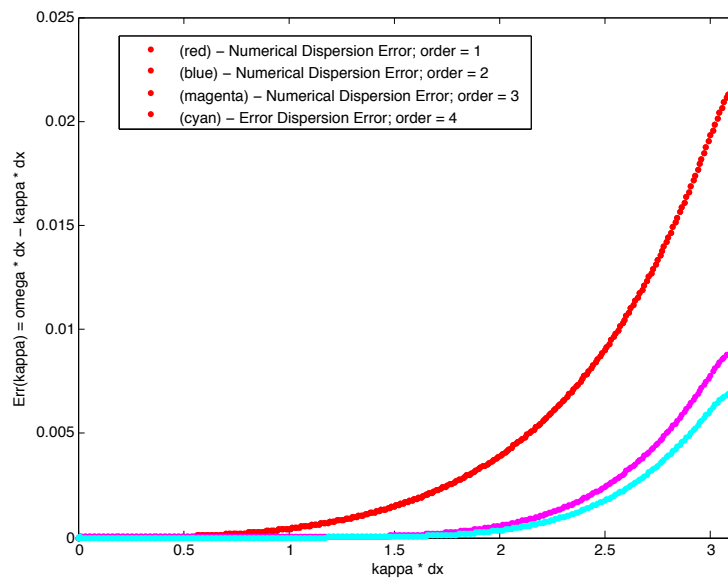
Fully-discrete Numerical Dispersion Relation LFEM with symplectic time-stepping of order 1 (red), 2 (blue), 3 (magenta) and 4 (cyan). The x-axis is scaled by  $\Delta x > 0$  and the y-axis is scaled by  $\Delta t > 0$ .

### 5.3 Comparison - QFEM with symplectic integrators



Continuous dispersion relation (black) and fully-discrete dispersion relation LFEM (red) , QFEM (blue), CFEM (magenta) together with Leapfrog time-stepping. The x-axis is scaled by  $\Delta x > 0$  and the y-axis is scaled by  $\Delta t > 0$ .

### 5.4 Comparison - CFEM with symplectic integrators

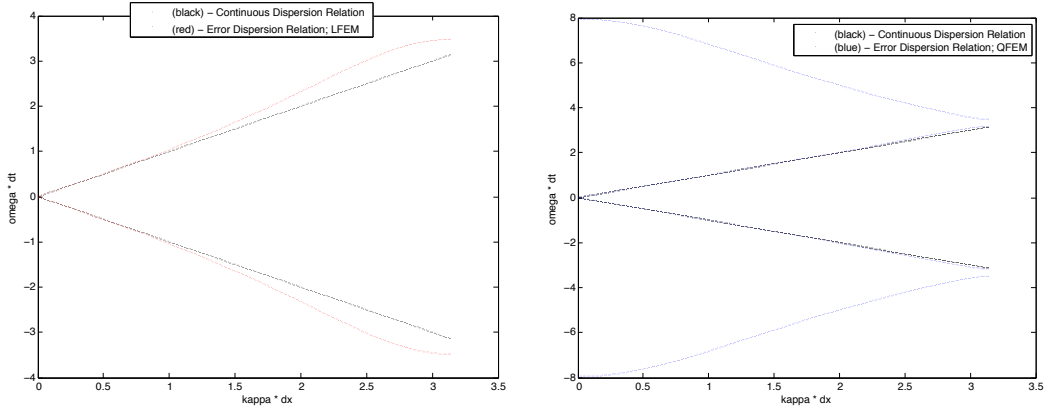


Continuous dispersion relation (black) and fully-discrete dispersion relation LFEM (red) , QFEM (blue), CFEM (magenta) together with Leapfrog time-stepping. The x-axis is scaled by  $\Delta x > 0$  and the y-axis is scaled by  $\Delta t > 0$ .

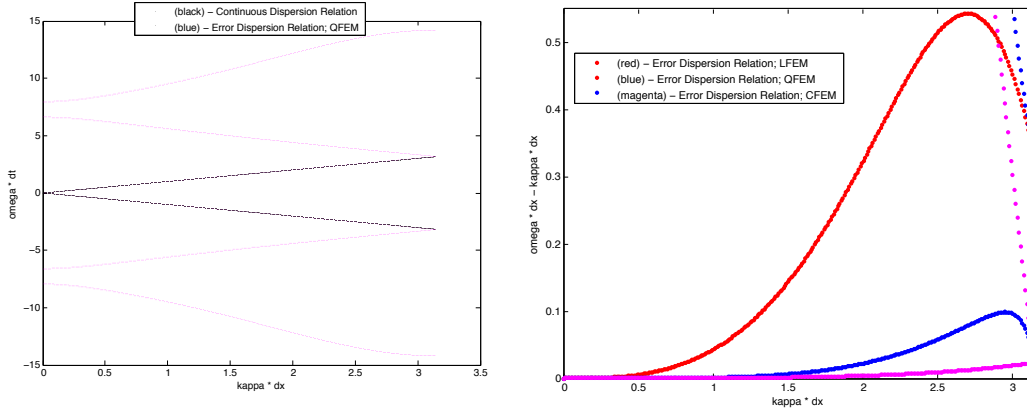


(blue), CFEM (magenta) together with Leapfrog time-stepping. The x-axis is scaled by  $\Delta x > 0$  and the y-axis is scaled by  $\Delta t > 0$ .

### 5.5 Semi-implicit Euler time-stepping

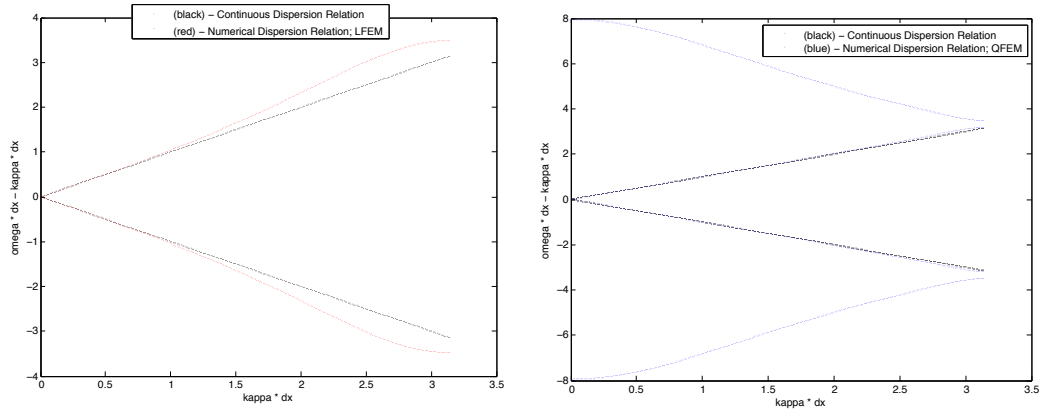


Numerical Dispersion Relation for LFEM (left) and QFEM (right). Note that the x-axis is scaled by  $\Delta x > 0$  and the y-axis is scaled by  $\Delta t > 0$

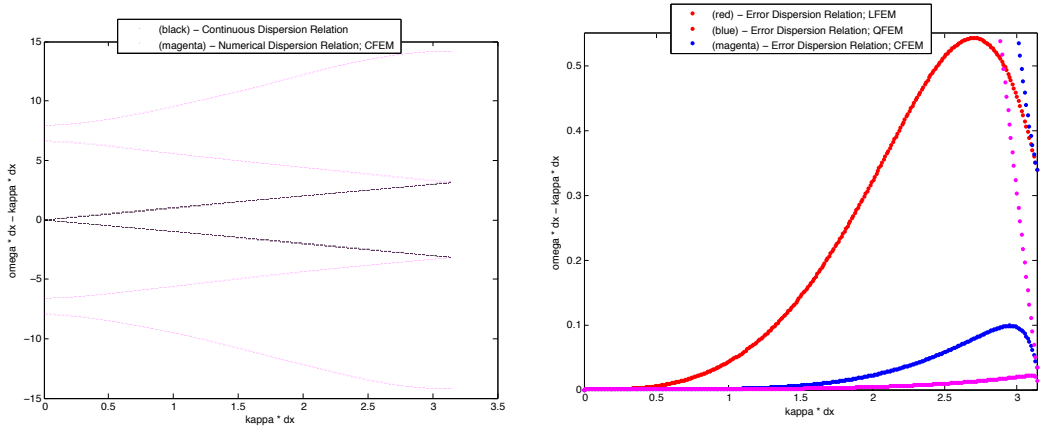


Numerical Dispersion Relation for CFEM (left) and combined Numerical Dispersion Relation Error [ $Err(\kappa) = \omega(\kappa) - \kappa$ ] for LFEM, QFEM and CFEM for a single  $\Delta t > 0$  (right).

## 5.6 Symplectic $2^{nd}$ -order time-stepping

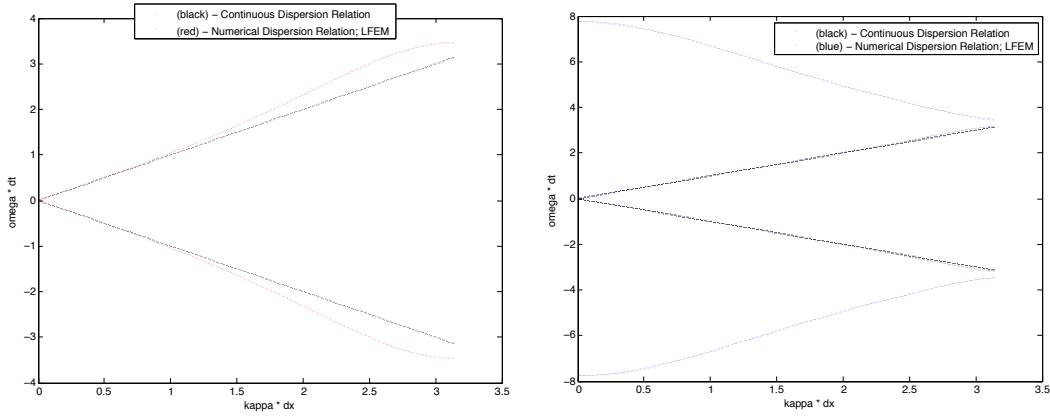


Numerical Dispersion Relation for LFEM (left) and QFEM (right). Note that the x-axis is scaled by  $\Delta x > 0$  and the y-axis is scaled by  $\Delta t > 0$

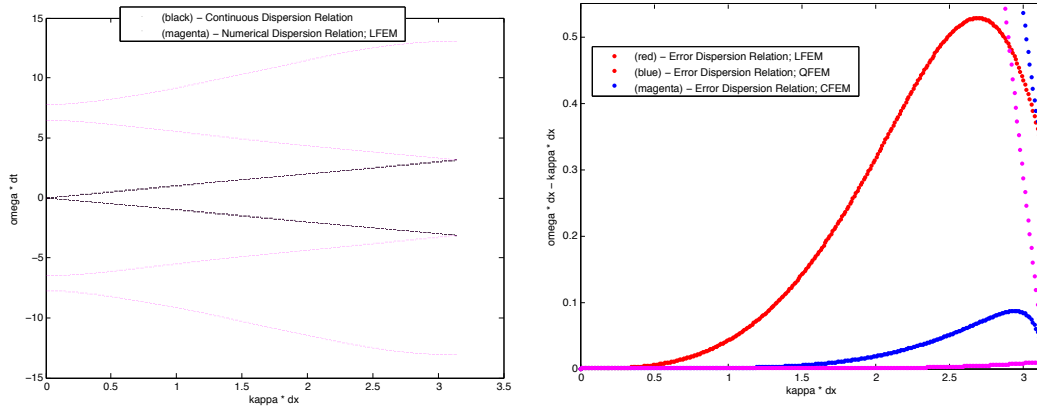


Numerical Dispersion Relation for CFEM (left) and combined Numerical Dispersion Relation Error  $[Err(\kappa) = \omega(\kappa) - \kappa]$  for LFEM, QFEM and CFEM for a single  $\Delta t > 0$  (right).

## 5.7 Symplectic 3<sup>rd</sup>-order time-stepping

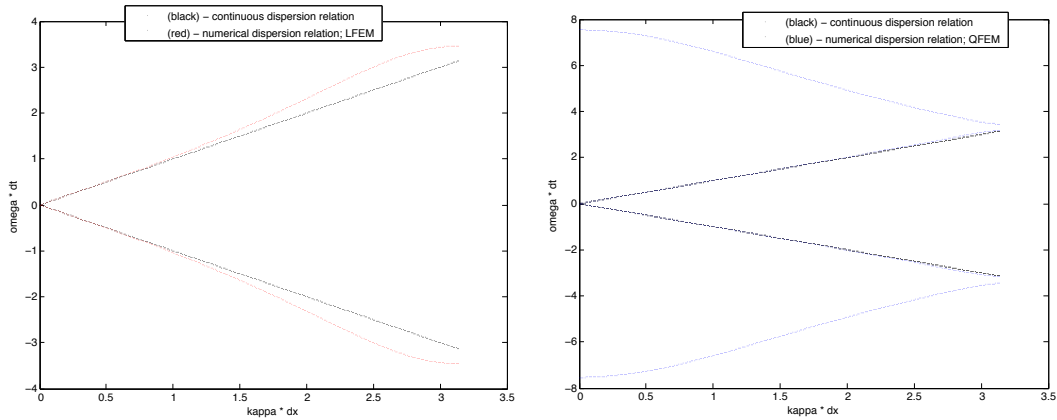


Numerical Dispersion Relation for LFEM (left) and QFEM (right). Note that the x-axis is scaled by  $\Delta x > 0$  and the y-axis is scaled by  $\Delta t > 0$

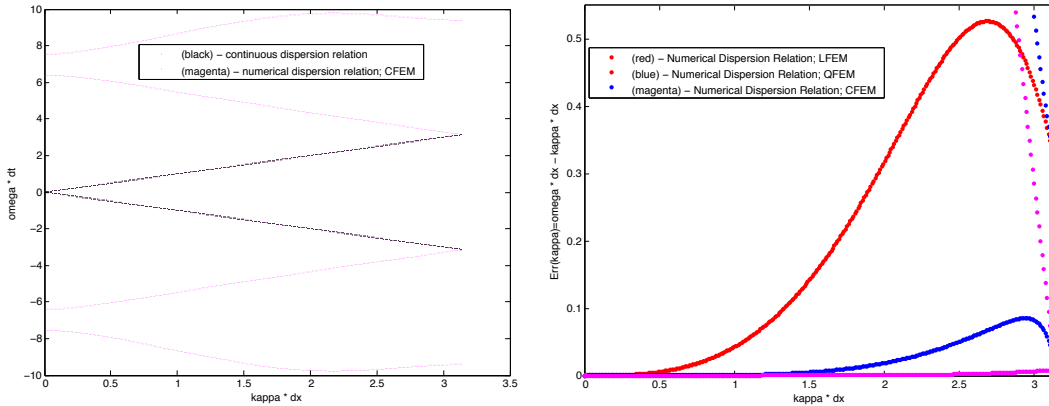


Numerical Dispersion Relation for CFEM (left) and combined Numerical Dispersion Relation Error  $[Err(\kappa) = \omega(\kappa) - \kappa]$  for LFEM, QFEM and CFEM for a single  $\Delta t > 0$  (right).

## 5.8 Symplectic 4<sup>th</sup>-order time-stepping



Numerical Dispersion Relation for LFEM (left) and QFEM (right). Note that the x-axis is scaled by  $\Delta x > 0$  and the y-axis is scaled by  $\Delta t > 0$



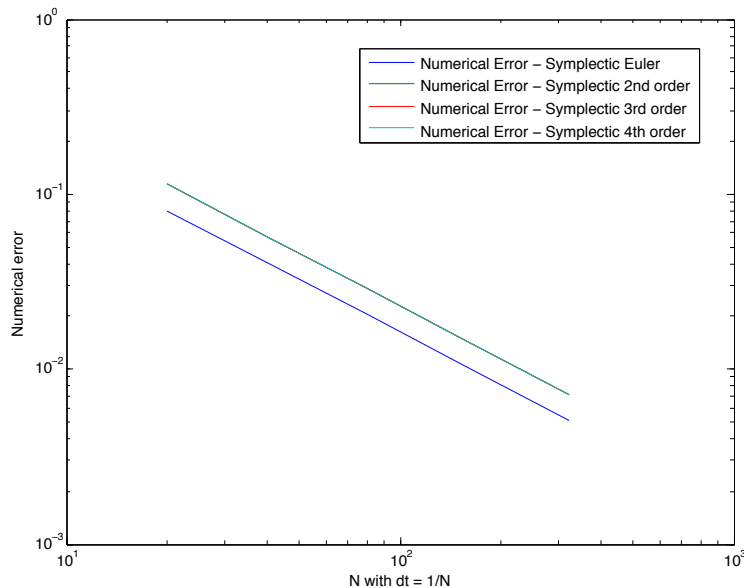
Numerical Dispersion Relation for CFEM (left) and combined Numerical Dispersion Relation Error [ $Err(\kappa) = \omega(\kappa) - \kappa$ ] for LFEM, QFEM and CFEM for a single  $\Delta t > 0$  (right).

## 6 Conclusion

The observed dispersion relations for spatial discretization of linear, quadratic and cubic finite elements provides the expected results. If we compare the numerical dispersion relation for the same time-stepping algorithm, we may observe that as the degree of the spatial discretization increases, the error in the numerical dispersion relation (compared to the continuous dispersion relation). This phenomenon was already observed in the case of the semi-discrete dispersion relation in section 3.

On the other hand, if we fix the degree of spatial discretization and compare the numerical dispersion relation for each of the symplectic time-stepping algorithms, we may not observe an improvement in the error of the numerical dispersion relation in all cases. For one, the numerical dispersion relation for symplectic Euler and second-order symplectic time-stepping are identical, regardless of

the degree of spatial discretization. However, if we compare the numerical dispersion error between symplectic Euler, third-order symplectic time-stepping and finally the fourth-order symplectic time-stepping, we may in fact observe the desired result. This is most likely a coincidence, since the empirical convergence rates of all symplectic time-stepping algorithms were observed to be the same. There seems to be a fundamental difference between the symplectic time-stepping algorithms applied to ODEs such as that of the model problem  $u'' = -u$  and the spatial discretization via Finite Elements of arbitrary order. When the empirical convergence rates of the symplectic time-stepping algorithms were analyzed for the Finite Element problem, the result was somewhat unexpected. It turns out that when the symplectic time-stepping algorithms are applied to the Finite Element problem, the order of the algorithm becomes irrelevant. That is, all symplectic time-stepping algorithms converge with the same order as the symplectic Euler algorithm. This is illustrated in the below figure



Empirical Numerical Error of symplectic time-stepping applied to the Finite Element problem in (45) with initial value problem  $u_0 = \sin(x)^2 \mathbb{1}_{[0, \pi]}(x)$  and  $v_0 = -2 \sin(x) \cos(x) \mathbb{1}_{[0, \pi]}(x)$ .

## 7 Appendix

### 7.1 Code - Leapfrog time-stepping (model problem)

```
function Reference = SymplecticModelProblemLeapfrog(N)

dt = 8*pi/N;

u = zeros(1,N+1);
v = zeros(1,N+1);
```

```

u(1) = 1;

v(1) = 0;

%First time step (initial step)

u(:,2) = u(1) + v(1)*dt - 0.5*dt^2*u(1);

%Leapfrog timestepping
for i=2:N

u(:,i+1) = ( 2*u(i) - u(i-1) - dt^2*u(i) );

end;

plot((1:N+1),u,'r');

Reference = u(end);

```

## 7.2 Code - Symplectic time-stepping (model problem)

```

function X = SymplecticModelProblem( N , order )

if order == 1

a = [1]; b = [1];

elseif order == 2

a = [0.5 0.5]; b = [0 1];

elseif order == 3

a = [2/3; -2/3; 1];
b = [7/24; 3/4; -1/24];

else

a = [ (2 + 2^(1/3) + 2^(-1/3))/6; (1 - 2^(1/3) - 2^(-1/3))/6;
      (1 - 2^(1/3) - 2^(-1/3))/6; (2 + 2^(1/3) + 2^(-1/3))/6];

b = [0; 1/(2 - 2^(1/3)); 1/(1 - 2^(2/3)); 1/(2 - 2^(1/3))];

end

```

```

dt = pi/N;

u = zeros(1,N+1);
v = zeros(1,N+1);

u(1) = 1;

v(1) = 0;

for j = 1 : N

v_in = v(j);

u_in = u(j);

    for s = 1 : order

v_out = v_in + b(s)*dt*u_in;

u_out = u_in - a(s)*dt*v_out;

v_in = v_out;

u_in = u_out;

    end

v(j+1) = v_out;

u(j+1) = u_out;

end

plot( (1:N+1) , u )

X = u(end);

```

### 7.3 Code - (Animation) Linear Finite Element for Leapfrog time-stepping

```

function Reference = LFEMSymplecticForErrorLeapfrog(N)

dt = 1/N; %time-step
n=10; %number of elements on the interval [0,pi]
dx = pi/n; %spatial-step width
T = 1; %Unit time interval
k = T/dt; %Number of time steps on unite time interval

```

```

alpha = dt/dx;

%Create Mass Matrix and Stiffness Matrix
v = ones(1,2*n-1);

%Stiffness Matrix
A = 2*diag([v 1]) - (diag(v,1) + diag(v,-1));

A(1,end) = -1;
A(end,1) = -1;

%Mass Matrix
M = (1/6)*( 4*diag([v 1]) + (diag(v,1) + diag(v,-1)) );

M(1,end) = 1/6;
M(end,1) = 1/6;

%Initial solution
x_int = linspace(0,pi,n); %Intermediate variable
x_0 = linspace(-pi, pi, 2*n); %X axis for plotting
y_0 = [zeros(1,n) sin(x_int).^2]; %Initial value
v_0 = [zeros(1,n) -2*sin(x_int).*cos(x_int)]; %Initial velocity

%Solution Matrix
sol = zeros(2*n,k);

%Initial value problem
sol(:,1) = y_0';

%First time step (initial step)
temp = A*sol(:,1);
temp = M\temp;

sol(:,2) = ( sol(:,1) + v_0'*dt + 0.5*alpha^2*temp);

%Leapfrog timestepping
for i=2:k-1

temp = A*sol(:,i);
temp = M\temp;

sol(:,i+1) = ( 2*sol(:,i) - sol(:,i-1) - alpha^2*temp );

clf
plot(x_0,sol(:,i),'r');

```



```

hold on
pause(0.1);
axis([-pi pi 0 1]);
grid on;

```

```
end;
```

```
Reference = sol(:,end);
```

## 7.4 Code - (Animation) Linear Finite Element for symplectic time-stepping

```
function Reference = LFEMSymplecticForError(N,order)
```

```

dt = 1/N; %time-step
n=10; %number of elements on the interval [0,pi]
dx = pi/n; %spatial-step width
T = 1; %Unit time interval
k = T/dt; %Number of time steps on unite time interval

```

```

%Determine Coefficients for time-stepping algorithm based on the input "order"
if order == 1

```

```
    a = [1]; b = [1];
```

```
elseif order == 2
```

```
    a = [0.5 0.5]; b = [0 1];
```

```
elseif order == 3
```

```
    a = [2/3; -2/3; 1];
    b = [7/24; 3/4; -1/24];
```

```
else
```

```
    a = [ (2 + 2^(1/3) + 2^(-1/3))/6; (1 - 2^(1/3) - 2^(-1/3))/6;
          (1 - 2^(1/3) - 2^(-1/3))/6; (2 + 2^(1/3) + 2^(-1/3))/6];
```

```
    b = [0; 1/(2 - 2^(1/3)); 1/(1 - 2^(2/3)); 1/(2 - 2^(1/3))];
```

```
end
```

```
%Create Mass Matrix and Stiffness Matrix
```

```
v = ones(1,2*n-1);
```

```

%Stiffness Matrix
A = 2*diag([v 1]) - (diag(v,1) + diag(v,-1));

A(1,end) = -1;
A(end,1) = -1;

A = A/dx;

%Mass Matrix
M = (1/6)*( 4*diag([v 1]) + (diag(v,1) + diag(v,-1)) );

M(1,end) = 1/6;
M(end,1) = 1/6;

M = M*dx;

%Initial solution
x_int = linspace(0,pi,n); %Intermediate variable
x_0 = linspace(-pi, pi, 2*n); %X axis for plotting
y_0 = [zeros(1,n) sin(x_int).^2]; %Initial value
v_0 = [zeros(1,n) 2*sin(x_int).*cos(x_int)]; %Initial velocity

%Solution Matrix
p = zeros(2*n,k); %initialize matrix for inertia
q = zeros(2*n,k); %initialize matrix for displacement

%Initial value problem
q(:,1) = y_0';
p(:,1) = v_0';

%Symplectic time-stepping
for i=1:k-1

    p_in = p(:,i);
    q_in = q(:,i);

    for j=1:order

        temp = b(j)*dt*A*q_in;

        temp = M\temp;

        p_out = p_in - temp;

        q_out = q_in + a(j)*dt*p_out;

```

```

        p_in = p_out;
        q_in = q_out;

    end

%Save inertia and displacement at next time-step
p(:,i+1) = p_out;
q(:,i+1) = q_out;

%Plot solution at i-th time-step
clf
plot(x_0,q(:,i),'r');
hold on
pause(0.1);
axis([-pi pi 0 1]);
    grid on;

end

Reference = q(:,end);

```

## 7.5 Code - Dispersion relation for semi-discrete scheme

```

function SymplecticGeneralizedSemi(dx,dim)
%%The following function determines the dispersion relation for Linear,
%%Quadratic and Cubic Lagrangian Finite Elements

%%1. Input: "dx" determines the spatial step-size

%%2. Input: "dim" determines the dimension of spatial discretization

m = 300; %Number of partitions in [0,pi]

kappa = linspace(0,pi,m);    %Kappa on [0,pi]

omega = zeros(dim,m);    %Solution vectors for plotting in each row

%%Start for-loop that determines the dispersion relation. The for
%%loop fixes kappa in [-pi,pi].
%%Then omega(kappa) is determined for that particular time-step and value of
%%kappa according to the Generalized Eigenvalue Problem (GEVP) determined
%%in the paper.

    for j=1:m

```

```

A = zeros(dim);      %Declare local stiffness matrix
B = zeros(dim);      %Declare local mass matrix
AA = eye(mult*dim);  %Declare AA for the GEVP: AA - lambda BB = 0
BB = eye(mult*dim);  %Declare BB for the GEVP: AA - lambda BB = 0

if dim == 1
    B = ( 2 + cos(kappa(j)) )/3;      %local mass matrix
    A = 4*sin(kappa(j)/2).^2;        %local stiffness
elseif dim == 2
    m1 = ( 2 + cos(kappa(j)) )/3;
    a1 = 4*sin(kappa(j)/2).^2;
    c1 = (1 + exp(-1i*kappa(j)))/3;
    c2 = (1 + exp(1i*kappa(j)))/3;
    B = [ m1 c1; c2 8/15 ];          %local mass matrix
    A = [ a1 0; 0 16/3];            %local stiffness
elseif dim == 3
    m1 = ( 2 + cos(kappa(j)) )/3;
    a1 = 4*sin(kappa(j)/2).^2;
    c1 = (1 + exp(-1i*kappa(j)))/3;
    c2 = (1 + exp(1i*kappa(j)))/3;
    d1 = 8/45*( 1 - exp(-1i*kappa(j)) );
    d2 = 8/45*( 1 - exp(1i*kappa(j)) );
    B = [ m1 c1 d1; c2 8/15 0; d2 0 512/945 ]; %local mass matrix
    A = [ a1 0 0; 0 16/3 0; 0 0 1024/45 ]; %local stiffness

```

```

        end

        E = eig(A,B);      %Solve the GEVP

        temp = E + 1;

        temp = acos(temp)\dt(t);

        omega(:,j) = real(temp);

    end

for j = 1:dim
plot(kappa,omega(j,:), 'r', kappa, kappa, 'b' )
legend( 'fully-discrete dispersion ', 'continuous dispersion relation ')
hold on;
drawnow;
end
end
end

```

## 7.6 Code - Dispersion relation for Leapfrog time-stepping

```

function LeapfrogGeneralized(dim,dx,dt)
%%The following function determines the dispersion relation for Linear,
%%Quadratic and Cubic Lagrangian Finite Elements together with Leapfrog time-stepping.
%%The inputs dim and order determine the spatial discretization and time-stepping order
%%respectively

m = 300;

kappa = linspace(0,pi,m);      %Kappa on [-pi,pi]

alpha = dt./dx;                %alpha determined for each timestep

omega = zeros(dim,m);

%%Start for-loop that determines the dispersion relation. The outer loop
%%fixes the time-step size and the inner loop fixes kappa in [-pi,pi]. Then
%%omega(kappa) is determined for that particular time-step and value of
%%kappa according to the Generalized Eigenvalue Problem (GEVP) determined
%%in the paper.
for k=1:dim
for j=1:m
    m1 = ( 2 + cos(kappa(j)) )/3;

```

```

a1 = 4*sin(kappa(j)/2).^2;
c1 = (1 + exp(-1i*kappa(j)))/3;
c2 = (1 + exp(1i*kappa(j)))/3;
d1 = 8/45*( 1 - exp(-1i*kappa(j)) );
d2 = 8/45*( 1 - exp(1i*kappa(j)) );

if dim == 1
    B = -2*m1 ;           %local mass matrix
    A = alpha^2*a1;      %local stiffness
elseif dim ==2
    B = -2*[ m1 c1; c2 8/15 ];   %local mass matrix
    A = alpha^2*[ a1 0; 0 16/3]; %local stiffness
elseif dim == 3
    B = -2*[ m1 c1 d1; c2 8/15 0; d2 0 512/945 ];   %local mass matrix
    A = alpha^2*[ a1 0 0; 0 16/3 0; 0 0 1024/45 ]; %local stiffness
end

%%In the following , we determine the matrices
%%necessary for determining the dispersion relation

E = eig(A,B);

temp = E(k) + 1;

temp = acos(temp);

omega(k,j) = real(temp)/dt;

end

plot( kappa ,omega(k,:) , 'r.' , kappa ,kappa , 'b.' , 'Markersize' ,5 )

```

```

xlabel('kappa * dx')
ylabel('omega * dx')
legend('Fully-discrete dispersion relation', 'Continuous dispersion relation')
hold on;
drawnow;
end
end

```

## 7.7 Code - Dispersion relation for symplectic time-stepping algorithms

```

function SymplecticGeneralizedStability(dim, order, dx, dt)
%%The following function determines the dispersion relation for Linear,
%%Quadratic and Cubic Lagrangian Finite Elements together with symplectic
%%time-stepping algorithms of arbitrary order 1 - 4. The inputs dim and
%%order determine the spatial discretization and time-stepping order
%%respectively

%Determine Coefficients for time-stepping algorithm based on the input "order"
if order == 1

    a = [1]; b = [1];

elseif order == 2

    a = [0.5 0.5]; b = [0 1];

elseif order == 3

    a = [2/3; -2/3; 1];
    b = [7/24; 3/4; -1/24];

else

    a = [ (2 + 2^(1/3) + 2^(-1/3))/6; (1 - 2^(1/3) - 2^(-1/3))/6;
          (1 - 2^(1/3) - 2^(-1/3))/6; (2 + 2^(1/3) + 2^(-1/3))/6];

    b = [0; 1/(2 - 2^(1/3)); 1/(1 - 2^(2/3)); 1/(2 - 2^(1/3))];

end

m = 300; %Number of partitions in [0,pi]

alpha = dt./dx; %alpha determined for each timestep

kappa = linspace(0,pi,m); %Kappa on [0,pi]

```

```

omega = zeros(2*dim,m);    %Solution vectors for plotting in each row

%%Start for-loop that determines the dispersion relation. The outer loop
%%fixes the time-step size and the inner loop fixes kappa in [-pi,pi]. Then
%%omega(kappa) is determined for that particular time-step and value of
%%kappa according to the Generalized Eigenvalue Problem (GEVP) determined
%%in the paper.

for j=1:m

    A = zeros(dim);        %Declare local stiffness matrix

    B = zeros(dim);        %Declare local mass matrix

    AA = eye(2*dim);       %Declare AA for the GEVP: AA - lambda BB = 0

    BB = eye(2*dim);       %Declare BB for the GEVP: AA - lambda BB = 0

    E = ones(2*dim,1);

    if dim == 1

        B = ( 2 + cos(kappa(j)) )/3;    %local mass matrix

        A = 4*sin(kappa(j)/2).^2;        %local stiffness

    elseif dim == 2

        m1 = ( 2 + cos(kappa(j)) )/3;

        a1 = 4*sin(kappa(j)/2).^2;

        c1 = (1 + exp(-1i*kappa(j)))/3;

        c2 = (1 + exp(1i*kappa(j)))/3;

        B = [ m1 c1; c2 8/15 ];          %local mass matrix

        A = [ a1 0; 0 16/3];            %local stiffness

    elseif dim == 3

        m1 = ( 2 + cos(kappa(j)) )/3;

        a1 = 4*sin(kappa(j)/2).^2;

```



```

    c1 = (1 + exp(-1i*kappa(j)))/3;
    c2 = (1 + exp(1i*kappa(j)))/3;
    d1 = 8/45*( 1 - exp(-1i*kappa(j)) );
    d2 = 8/45*( 1 - exp(1i*kappa(j)) );
    B = [ m1 c1 d1; c2 8/15 0; d2 0 512/945 ]; %local mass matrix
    A = [ a1 0 0; 0 16/3 0; 0 0 1024/45 ]; %local stiffness
end

%%In the following loop, we determine the matrices
%%AA and BB for the Generalized Eigenvalue Problem

tildeM = [ eye(size(B)) zeros(size(B)); zeros(size(B)) B ];
UpperRight = [ zeros(size(A)) eye(size(A)); zeros(size(A)) zeros(size(A))];
LowerLeft = [ zeros(size(A)) zeros(size(A)); -A zeros(size(A))];

for k =1:order
    Left = (tildeM - dt*a(k)*UpperRight);
    Right = (tildeM + dt*b(k)*LowerLeft);
    BB = inv(Right)*Left*BB;
end

E = eig( AA , BB );

temp = E;

temp = log(temp)/(dt*1i); %Determine omega( kappa )

omega(:,j) = real(temp); %Ensure that omega( kappa ) is real-valued

end

for t = 1:2*dim
    plot( kappa,omega(t,:), 'r.', kappa, kappa , 'b.', 'MarkerSize',2)
    xlabel('kappa * dx')

```

```
ylabel('Err(kappa)=omega * dx - kappa * dx')
legend('(red) - Numerical Dispersion Relation ',
       '(blue) - Continuous Dispersion Relation ')
hold on
```

end

end

## References

- [1] R. Rieben, D. White and G. Rodrigue, "High-Order Symplectic Integration Methods for Finite Element Solution to Time Dependent Maxwell Equations", *IEEE Trans. Antennas Propagat.*, vol. 52, No. 8, pp. 2190 - 2195, August 2004
- [2] Ruth, Ronald D., "A Canonical Integration Technique", *Nuclear Science, IEEE Trans. on NS*, vol. 30, No. 4, pp. 2669 - 2671, August 1983
- [3] Forest, E.; Ruth, Ronald D., "Fourth-order symplectic integrator", *Physica D*, vol. 43, p. 105, 1990