



Theory and Implementation of Coupled Port-Hamiltonian Continuum and Lumped Parameter Models

Finbar J. Argus¹ · Chris P. Bradley¹ · Peter J. Hunter¹

Received: 30 September 2020 / Accepted: 10 June 2021 / Published online: 13 July 2021
© The Author(s) 2021, corrected publication 2021

Abstract

A continuous Galerkin finite element method that allows mixed boundary conditions without the need for Lagrange multipliers or user-defined parameters is developed. A mixed coupling of Lagrange and Raviart-Thomas basis functions are used. The method is proven to have a Hamiltonian-conserving spatial discretisation and a symplectic time discretisation. The energy residual is therefore guaranteed to be bounded for general problems and exactly conserved for linear problems. The linear 2D wave equation is discretised and modelled by making use of a port-Hamiltonian framework. This model is verified against an analytic solution and shown to have standard order of convergence for the temporal and spatial discretisation. The error growth over time is shown to grow linearly for this symplectic method, which agrees with theoretical results. A modal analysis is performed which verifies that the eigenvalues of the model accurately converge to the exact eigenvalues, as the mesh is refined. The port-Hamiltonian framework allows boundary coupling with bond-graph or, more generally, lumped parameter models, therefore unifying the two fields of lumped parameter modelling and continuum modelling of Hamiltonian systems. The wave domain discretisation is shown to be equivalent to a coupling of canonical port-Hamiltonian forms. This feature allows the model to have mixed boundary conditions as well as to have mixed causality interconnections with other port-Hamiltonian models. A model of the 2D wave equation is coupled, in a monolithic manner, with a lumped parameter model of an electromechanical linear actuator. The combined model is also verified to conserve energy exactly.

Keywords Port-Hamiltonian · Modelling · PDE · FEniCS · Hamiltonian · Finite element · Galerkin · Symplectic · Lumped parameter · Continuum · Monolithic · Bond graph

Mathematics Subject Classification Primary 65P10 · Secondary 35L05, 74F99, 65M60

1 Introduction

As computational power increases and the desire for ever more complex models grows, the need to have an energy-conserving mathematical framework for multiphysics, multidomain

✉ F.J. Argus
farg967@aucklanduni.ac.nz

¹ Auckland Bioengineering Institute, University of Auckland, Auckland, New Zealand

problems is becoming apparent. Many fields require complex couplings between continuum and lumped parameter models (LPMs) including physiology, aerospace, vehicle dynamics, robotics, and many more. The coupling of these models is typically done in an iterative manner, which has multiple disadvantages. Two major downsides are the failure of iterative couplings to conserve energy [35] and the difficulty in finding a stable combination of timesteps for the coupled models [45]. Another equally large disadvantage of the iterative approach is the difficulty of implementing control algorithms without a monolithic state-space matrix. A fully coupled, monolithic approach that conserves the structural properties of the lumped parameter and continuum models is desirable.

In this paper we discuss the energy conservation of a continuum system described by a partial differential equation (PDE) coupled with a LPM. Specifically, a model is derived for the linear 2D wave equation with an electromechanical (EM) linear actuator driving one of the boundaries. This example also shows that the port-Hamiltonian (PH) formalism can be used to interconnect various types of models while conserving energy flow through ports/boundaries. PH is a proven method for the modelling and control of complex multiphysics systems. Over the last 20 years, a significant amount of work has been done on infinite-dimensional PH methods for the modelling of continuum systems [17, 42]. Some PDEs that have been modelled using the PH framework are transmission line equations, shallow water equations and Timoshenko beam equations [14]. More recently, both 2D and 3D models have been implemented using the PH framework [38, 41, 44].

In Sect. 2 the spatial discretisation is proven to conserve the Hamiltonian in the same manner as the exact equations. Section 3 formulates the discretisation of the temporal domain with well known symplectic integrators: the symplectic Euler (SE) method, the symplectic (implicit) midpoint (SM) method and the Störmer-Verlet (SV) (leapfrog) method. A symplectic method conserves volume in phase space, which results in bounded conservation of the Hamiltonian [20]. The combined spatial-temporal discretisation, therefore, conserves the Hamiltonian structure of the governing equations. A similar class of methods that conserves the Hamiltonian structure is the class of multi-symplectic methods. Multiple groups have applied multi-symplectic methods to both the linear and non-linear wave equations. Reich used Runge-Kutta finite difference schemes in both space and time [34], McLachlin compared multiple methods, including spectral methods [28], and Brugnano used a Hamilton Boundary Value Method [5]. McDonald shows that multi-symplecticity preserves the travelling waves of hyperbolic equations [27]. The Partitioned Finite Element Method (PFEM) in the paper by Cardoso-Ribeiro et al. [11] combines a finite element spatial discretisation with an SV time integration scheme in a way that conserves energy but requires Lagrange multipliers to implement mixed boundary conditions [21]. Brugnoli et al. successfully apply this approach to Mindlin and Kirchhoff plate models [7, 8]. Brugnoli et al. [9] have also introduced another method for applying mixed boundary conditions to PFEM. Their method requires discretisation of the spatial domain into separate sections, each section having one type of boundary condition. The Stokes-Dirac structure [43] and interconnection property of PH methods is then used to combine the multiple sections, creating a full system with mixed boundary conditions. Similar to our method, Kotyczka uses a finite element discretisation and symplectic time integration in a way that conserves the Hamiltonian structure and allows for mixed boundary conditions [22], however, user-defined parameters in the method must be tuned for accurate results. The method introduced in this paper combines desirable attributes of Cardoso-Ribeiro's, Brugnoli's, and Kotyczka's methods and provides a Hamiltonian-conserving, symplectic method that allows for easily implemented mixed boundary conditions, port-based boundary coupling, and does not require tuning of user-defined parameters.

To ensure conservation of energy flow through the boundaries, a weak boundary condition implementation is used for the Dirichlet conditions, similar to the way Neumann conditions are typically implemented in finite element methods. Weak boundary conditions are implemented in the variational form and provide many benefits for a finite element formulation. One of these benefits is that it simplifies the implementation by not having to directly prescribe the degrees of freedom (DOFs) at the boundary. This can have benefits in applying multiple types of boundary conditions, including no-slip conditions for the Navier-Stokes equations [3]. Also, no manipulation of the solution matrix is required to prescribe the DOFs. The ability to have mixed boundary conditions is also extended to allow mixed causality interconnections with other PH models. By showing that our spatial discretisation is equivalent to a coupling of canonical PH models with either Neumann or Dirichlet boundaries and by following the interconnection methods of Brugnoli [6], we calculate the causal boundary connections with a LPM. The ability to use the canonical forms to calculate the power conserving interconnection allows mixed causality boundary interconnection between other PH models.

Continuous Galerkin and hybridizable discontinuous Galerkin methods couple well with PH methods due to the ease of ensuring structural properties of the models and the simplicity of coupling different models at the boundary. Brezzi and Fortin give a good overview of Galerkin methods [4]. Cockburn developed a unifying framework for Galerkin methods [12] that multiple authors have extended. McLachlan extends Cockburn's discontinuous Galerkin (DG) methods into a formulation that allows proof of multi-symplecticity for elliptic equations [29]. Sanchez uses a hybridizable discontinuous Galerkin approach to obtain a wave equation discretisation similar to our method, that is proven to be Hamiltonian-conserving and symplectic in time [36]. We prove the same qualities for our method, a continuous Galerkin approach that has the important added novelty of being able to be coupled with arbitrary PH LPMs. The conservativity and modular approach of this method is thus ideal for a wide range of real-world problems.

Energy conservation in the Galerkin method is extremely important for real-world applications that require long-time simulations, such as in geophysical fluid dynamics. Bauer uses a Poisson bracket approach to prove conservation of energy for a Galerkin discretisation of the rotating shallow water equations [2]. Extending on Bauer's work, Eldred uses the Galerkin method coupled with a Poisson time integrator to conserve energy when modelling the thermal shallow water equations [15]. Both of these models take advantage of Hamiltonian-conservation to give good long-time prediction for geophysical fluid dynamics applications.

Modelling of wave propagation also has applications in multiple biological fields. In this paper, the wave equation models homogeneous, linear-elastic media. For extensions with heterogeneous materials, typical of most biological materials, see the paper by Serhani [38]. Elastography [31] is an interesting application that requires a type of inverse modelling to identify the elasticity of heart tissue. A coupled LPM-continuum model could improve the current elastography methods. There has also been work done on non-linear models for wave propagation through biological tissue [13].

The model in this paper is implemented with the software FEniCS [1], which is a tool for automated scientific computing that focuses on solving PDEs. FEniCS allows for a very efficient implementation of finite element methods specified in a weak form. Another strong attribute of FEniCS is its automatic differentiation, which is valuable for inverse problems and for control.

The section outline of the paper is as follows. Section 2 details the PH form of the wave equation, proves that the discretisation conserves energy, and proves that the discretisation

retains the Stokes-Dirac structure. The FEniCS implementation and the results showing energy conservation are shown in Sect. 3. Section 4 validates the wave equation against an analytical solution by showing spatial, temporal, and eigenvalue convergence. An EM model is introduced in Sect. 5. In Sect. 6 the wave equation is coupled with the EM model in a monolithic approach that conserves the canonical PH structure. Finally, in Sect. 7, the results of the combined model are shown and the energy conservation of the model is discussed. Appendices A and C detail the Python code for Sects. 3 and 6, respectively.

2 The Wave Equation

This section details the weak form PH discretisation of the wave equation with constant propagation speed, the conservation of energy proof, and the proof that the discretisation ensures a Stokes-Dirac structure. The 2D wave equation in Cartesian coordinates is used throughout this paper, however, all results in this section naturally extend to the 3D wave equation. The basic linear wave equation is

$$\partial_t^2 w(\mathbf{x}, t) = c^2 \Delta w(\mathbf{x}, t), \tag{1}$$

where w is the wave amplitude, \mathbf{x} denotes the spatial coordinates, and t is the time. To model the simplified wave propagation in an elastic membrane we define $c^2 = k_w / \rho_w$ as the wave speed squared, which is a constant function of the material density (ρ_w) and stiffness (k_w). To transform Equation (1) into PH form, state variables, \tilde{p} , the momentum, and $\tilde{\mathbf{q}}$, the strain, are chosen as

$$\tilde{p} = -\rho_w \partial_t w, \quad \tilde{\mathbf{q}} = \nabla w. \tag{2}$$

This transforms the second-order equation into a system of $(n + 1)$ first-order equations, where n is the number of spatial dimensions. Note that the tilde overscript is used to denote exact variables, to distinguish them from the approximate functions used in subsequent sections. Using the PH notation of flow and effort variables, the time derivative of the state variables, \tilde{f}_p and \tilde{f}_q , are defined as flows

$$\tilde{f}_p = -\partial_t \tilde{p}, \quad \tilde{f}_q = -\partial_t \tilde{\mathbf{q}}. \tag{3}$$

The Hamiltonian functional and the Hamiltonian density are respectively given by

$$H(\tilde{p}, \tilde{\mathbf{q}}) = \int_{\Omega} \mathcal{H}(\tilde{p}, \tilde{\mathbf{q}}) dV, \tag{4}$$

$$\mathcal{H}(\tilde{p}, \tilde{\mathbf{q}}) = \frac{1}{2\rho_w} \tilde{p}^2 + \frac{1}{2} k_w \tilde{\mathbf{q}} \cdot \tilde{\mathbf{q}}, \tag{5}$$

where Ω is an open, bounded spatial domain with a Lipschitz-continuous boundary, $\partial\Omega$. The effort variables \tilde{e}_p , the velocity, and $\tilde{\mathbf{e}}_q$, the stress, are defined as the variational derivatives of the Hamiltonian density,

$$\tilde{e}_p = \partial_{\tilde{p}} \mathcal{H} = \frac{\tilde{p}}{\rho_w}, \quad \tilde{\mathbf{e}}_q = \partial_{\tilde{\mathbf{q}}} \mathcal{H} = k_w \tilde{\mathbf{q}}. \tag{6}$$

For a functional that only depends on its states, not on their spatial derivatives, the variational derivatives are equal to the partial derivatives of the integrand. Transforming Equations (1) and (3) to (6) into a PH structure gives

$$\begin{bmatrix} \tilde{f}_p \\ \tilde{f}_q \end{bmatrix} = \begin{bmatrix} 0 & \text{div} \\ \text{grad} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{e}_p \\ \tilde{e}_q \end{bmatrix} = \mathbf{J} \begin{bmatrix} \tilde{e}_p \\ \tilde{e}_q \end{bmatrix}, \tag{7}$$

where the div and grad operators make up the formally skew-adjoint \mathbf{J} operator. For a proof of the skew-adjointness of \mathbf{J} see the work by Trenchant et al. [40].

Theorem 2.0.1 Equation (7) is energy-conserving. i.e., the rate of change of the Hamiltonian is equal to the energy flow through the domain boundary.

Proof

$$\begin{aligned} \dot{H} &= \int_{\Omega} \left(\frac{\partial H}{\partial \tilde{p}} \frac{\partial \tilde{p}}{\partial t} + \frac{\partial H}{\partial \tilde{q}} \cdot \frac{\partial \tilde{q}}{\partial t} \right) dV \\ &= - \int_{\Omega} \left(\tilde{e}_p \tilde{f}_p + \tilde{e}_q \cdot \tilde{f}_q \right) dV. \end{aligned} \tag{8}$$

Equation (8) satisfies the well known bond-graph and PH condition, that the product of the effort and flow variables equals the power [42]. Substituting Equation (7) into Equation (8) and using integration by parts gives

$$\begin{aligned} \dot{H} &= - \int_{\Omega} \left(\tilde{e}_p (\nabla \cdot \tilde{e}_q) + \tilde{e}_q \cdot \nabla \tilde{e}_p \right) dV \\ &= - \int_{\Omega} \left(\tilde{e}_p (\nabla \cdot \tilde{e}_q) - (\nabla \cdot \tilde{e}_q) \tilde{e}_p \right) dV - \int_{\partial\Omega} (\tilde{e}_q \cdot \mathbf{n}) \tilde{e}_p ds \\ &= - \int_{\partial\Omega} (\tilde{e}_q \cdot \mathbf{n}) \tilde{e}_p ds. \quad \square \end{aligned}$$

Therefore, the rate of change of the Hamiltonian is only dependent on the boundary terms and thus the Hamiltonian is conserved within the internal domain.

Corollary 2.0.2 The canonical inputs and outputs of the system are either

$$\tilde{u}_q = -(\tilde{e}_q \cdot \mathbf{n})|_{\partial\Omega} = -k_w(\tilde{q} \cdot \mathbf{n})|_{\partial\Omega}, \quad \tilde{y}_p = \tilde{e}_p|_{\partial\Omega} = \frac{\tilde{p}}{\rho_w} \Big|_{\partial\Omega}, \tag{9}$$

respectively, or for the opposite causality the inputs and outputs are respectively,

$$\tilde{y}_q = (\tilde{e}_q \cdot \mathbf{n})|_{\partial\Omega} = k_w(\tilde{q} \cdot \mathbf{n})|_{\partial\Omega}, \quad \tilde{u}_p = -\tilde{e}_p|_{\partial\Omega} = -\frac{\tilde{p}}{\rho_w} \Big|_{\partial\Omega}. \tag{10}$$

2.1 Weak Form

In this section, the discretised weak form of the wave equation is derived. First, we define the L^2 inner product over the domain, Ω , and the boundary, $\partial\Omega$, as

$$\langle \mathbf{a} | \mathbf{b} \rangle_{\Omega} = \int_{\Omega} \mathbf{a} \cdot \mathbf{b} dV, \quad \langle \mathbf{a} | \mathbf{b} \rangle_{\partial\Omega} = \int_{\partial\Omega} \mathbf{a} \cdot \mathbf{b} ds. \tag{11}$$

Using the same function spaces as in Cardoso-Ribeiro’s work [11], approximate flow and effort functions are introduced,

$$\begin{aligned} f_p, e_p &\in H^1(\Omega), \\ \mathbf{f}_q, \mathbf{e}_q &\in H^{\text{div}}(\Omega). \end{aligned} \tag{12}$$

In the succeeding sections, we show that this choice of function spaces allows us to combine important features of Cardoso-Ribeiro’s and Kotyczka’s [22] methods. Cardoso-Ribeiro’s methods ease of implementation is combined with Kotyczka’s methods ability to implement mixed boundary conditions without Lagrange multipliers. Substituting these approximate functions for the exact flows and efforts in Equation (7) and taking the inner product with the test functions (v_p, \mathbf{v}_q) gives

$$\langle v_p | f_p \rangle_\Omega = \langle v_p | \nabla \cdot \mathbf{e}_q \rangle_\Omega, \tag{13a}$$

$$\langle \mathbf{v}_q | \mathbf{f}_q \rangle_\Omega = \langle \mathbf{v}_q | \nabla e_p \rangle_\Omega. \tag{13b}$$

The right-hand side of Equations (13a) and (13b) are then integrated by parts to give

$$\langle v_p | f_p \rangle_\Omega = -\langle \nabla v_p | \mathbf{e}_q \rangle_\Omega + \langle v_p | (\mathbf{e}_q \cdot \mathbf{n}) \rangle_{\partial\Omega}, \tag{14a}$$

$$\langle \mathbf{v}_q | \mathbf{f}_q \rangle_\Omega = -\langle (\nabla \cdot \mathbf{v}_q) | e_p \rangle_\Omega + \langle (\mathbf{v}_q \cdot \mathbf{n}) | e_p \rangle_{\partial\Omega}. \tag{14b}$$

These equations are now in a form where the Galerkin method can be applied. To do this, the approximate flow and effort functions in Equation (12) are defined from the discrete flow $(\hat{\mathbf{f}}_p \in \mathbb{R}^{N_p}, \hat{\mathbf{f}}_q \in \mathbb{R}^{N_q})$ and effort $(\hat{\mathbf{e}}_p \in \mathbb{R}^{N_p}, \hat{\mathbf{e}}_q \in \mathbb{R}^{N_q})$ vectors and the vectors of globally defined basis functions $(\boldsymbol{\varphi}_p, \boldsymbol{\varphi}_q)$ as shown here,

$$f_p(\mathbf{x}) = \sum_{i=1}^{N_p} \hat{f}_{pi} \varphi_{pi}(\mathbf{x}) = \langle \hat{\mathbf{f}}_p, \boldsymbol{\varphi}_p(\mathbf{x}) \rangle, \tag{15}$$

$$\mathbf{f}_q(\mathbf{x}) = \sum_{j=1}^{N_q} \hat{f}_{qj} \boldsymbol{\varphi}_{qj}(\mathbf{x}) = (\boldsymbol{\varphi}_q(\mathbf{x}))^T \hat{\mathbf{f}}_q,$$

$$e_p(\mathbf{x}) = \sum_{i=1}^{N_p} \hat{e}_{pi} \varphi_{pi}(\mathbf{x}) = \langle \hat{\mathbf{e}}_p, \boldsymbol{\varphi}_p(\mathbf{x}) \rangle, \tag{16}$$

$$\mathbf{e}_q(\mathbf{x}) = \sum_{j=1}^{N_q} \hat{e}_{qj} \boldsymbol{\varphi}_{qj}(\mathbf{x}) = (\boldsymbol{\varphi}_q(\mathbf{x}))^T \hat{\mathbf{e}}_q,$$

where N_p and N_q are the number of DOFs stored by the discrete vectors. A hat over a variable is used to denote the vector of discrete values, i.e., $\hat{\mathbf{e}}_p$ is the column vector of discrete DOF values for the scalar field e_p . To make the method Galerkin, v_p and \mathbf{v}_q are discretised with the same basis functions as f_p and \mathbf{f}_q , respectively. The basis function families that we use are Lagrange for $\boldsymbol{\varphi}_p$ and Raviart-Thomas [33] for $\boldsymbol{\varphi}_q$, however, any basis functions that satisfy the function spaces in Equation (12) are suitable. In Equations (15) and (16), the sizes of $\boldsymbol{\varphi}_p$ and $\boldsymbol{\varphi}_q$ are $N_p \times 1$ and $N_q \times 2$, respectively. The notation, $\langle \cdot, \cdot \rangle$ is used for the standard inner product on \mathbb{R} , as opposed to $\langle \cdot | \cdot \rangle_\Omega$, the L^2 inner product. For the

lowest order Lagrange and Raviart-Thomas elements, \hat{e}_p and \hat{f}_p are stored at nodes and \hat{e}_q and \hat{f}_q are stored at edges. For details on higher order elements see the FEniCS book [26].

Substituting the approximate functions from Equations (15) and (16) as well as the corresponding test functions into Equation (14a) gives

$$\begin{aligned} & \langle v_p | f_p \rangle_\Omega = -\langle \nabla v_p | e_q \rangle_\Omega + \langle v_p | (e_q \cdot n) \rangle_{\partial\Omega}, \\ \rightarrow \hat{v}_p^T \langle \varphi_p | \varphi_p \rangle_\Omega \hat{f}_p &= -\hat{v}_p^T \langle \nabla \varphi_p | \varphi_q \rangle_\Omega \hat{e}_q + \hat{v}_p^T \langle \psi_p | \psi_q \rangle_{\partial\Omega} \hat{e}_q, \\ \rightarrow \hat{v}_p^T \mathbf{M}_p \hat{f}_p &= \hat{v}_p^T \mathbf{K}_p \hat{e}_q + \hat{v}_p^T \mathbf{L}_p \hat{e}_q, \end{aligned} \tag{17}$$

where $\psi_p = \varphi_p|_{\partial\Omega}$ represents φ_p evaluated at the boundary and $\psi_q = \varphi_q \cdot n|_{\partial\Omega}$ represents φ_q evaluated in the normal direction at the boundary. More clearly, the basis functions for the boundary terms satisfy the following relations,

$$e_p|_{\partial\Omega} = \langle \hat{e}_p, \psi_p \rangle, \quad e_q \cdot n|_{\partial\Omega} = \langle \hat{e}_q, \psi_q \rangle. \tag{18}$$

Substituting the approximate functions from Equations (15), (16) and (18), as well as the corresponding test functions into Equation (14b) gives

$$\begin{aligned} & \langle v_q | f_q \rangle_\Omega = -\langle (\nabla \cdot v_q) | e_p \rangle_\Omega + \langle (v_q \cdot n) | e_p \rangle_{\partial\Omega}, \\ \rightarrow \hat{v}_q^T \langle \varphi_q | \varphi_q \rangle_\Omega \hat{f}_q &= -\hat{v}_q^T \langle (\nabla \cdot \varphi_q) | \varphi_p \rangle_\Omega \hat{e}_p + \hat{v}_q^T \langle \psi_q | \psi_p \rangle_{\partial\Omega} \hat{e}_p, \\ \rightarrow \hat{v}_q^T \mathbf{M}_q \hat{f}_q &= \hat{v}_q^T \mathbf{K}_q \hat{e}_p + \hat{v}_q^T \mathbf{L}_q \hat{e}_p. \end{aligned} \tag{19}$$

The matrices in Equations (17) and (19) are given as

$$\begin{aligned} \mathbf{M}_p &= \langle \varphi_p | \varphi_p \rangle_\Omega, \\ \mathbf{M}_q &= \langle \varphi_q | \varphi_q \rangle_\Omega, \\ \mathbf{K}_p &= -\langle \nabla \varphi_p | \varphi_q \rangle_\Omega, \\ \mathbf{K}_q &= -\langle \nabla \cdot \varphi_q | \varphi_p \rangle_\Omega, \\ \mathbf{L}_p &= \langle \psi_p | \psi_q \rangle_{\partial\Omega}, \\ \mathbf{L}_q &= \langle \psi_q | \psi_p \rangle_{\partial\Omega}, \end{aligned} \tag{20}$$

where the inner product acts elementwise for the matrix as

$$\langle a | b \rangle_\Omega^{ij} = \langle a^i | b^j \rangle_\Omega. \tag{21}$$

Applying a typical weak form approach, Equations (17) and (19) must hold for any $\hat{v}_p \in \mathbb{R}^{N_p}$ and $\hat{v}_q \in \mathbb{R}^{N_q}$. Therefore, according to the fundamental theorem of variational calculus, the following matrix system of equations holds,

$$\begin{aligned} \mathbf{M}_p \hat{f}_p &= \mathbf{K}_p \hat{e}_q + \mathbf{L}_p \hat{e}_q, \\ \mathbf{M}_q \hat{f}_q &= \mathbf{K}_q \hat{e}_p + \mathbf{L}_q \hat{e}_p. \end{aligned} \tag{22}$$

When implementing Equation (22), it is essential to include the discrete dynamic and constitutive laws which coincide with Equations (3) and (6), respectively. The dynamic and

constitutive laws are respectively,

$$\frac{\partial}{\partial t} \begin{bmatrix} \hat{p} \\ \hat{q} \end{bmatrix} = \begin{bmatrix} -\hat{f}_p \\ -\hat{f}_q \end{bmatrix}, \tag{23}$$

$$\begin{bmatrix} \hat{e}_p \\ \hat{e}_q \end{bmatrix} = \begin{bmatrix} \frac{1}{\rho_w} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & k_w \mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{p} \\ \hat{q} \end{bmatrix} = \mathbf{Q} \begin{bmatrix} \hat{p} \\ \hat{q} \end{bmatrix}, \tag{24}$$

where \mathbf{I} is the identity matrix. Also, \hat{p} and \hat{q} are the discrete vectors of momentum DOFs and strain DOFs, respectively. This formulation assumes constant material properties and therefore a constant \mathbf{Q} matrix. Finally, substituting Equations (23) and (24) into Equation (22) we get the matrix system of equations,

$$\begin{bmatrix} -\mathbf{M}_p & \mathbf{0} \\ \mathbf{0} & -\mathbf{M}_q \end{bmatrix} \frac{\partial}{\partial t} \begin{bmatrix} \hat{p} \\ \hat{q} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{K}_p + \mathbf{L}_p \\ \mathbf{K}_q + \mathbf{L}_q & \mathbf{0} \end{bmatrix} \begin{bmatrix} \frac{1}{\rho_w} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & k_w \mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{p} \\ \hat{q} \end{bmatrix}. \tag{25}$$

In Sect. 2.3 we prove that Equation (25) can be written in two different (one for Dirichlet and one for Neumann boundary conditions) canonical PH forms and that it represents a non-degenerate Stokes-Dirac structure.

2.2 Discrete Conservation of Power Proof

To proceed with a conservation of power proof we first define the approximate Hamiltonian, \hat{H} , and Hamiltonian density, $\hat{\mathcal{H}}$ in the same way as the exact functions that were defined in Equations (4) and (5),

$$\hat{H}(p, \mathbf{q}) = \int_{\Omega} \hat{\mathcal{H}}(p, \mathbf{q}) dV = \int_{\Omega} \left(\frac{1}{2\rho_w} p^2 + \frac{1}{2} k_w \mathbf{q} \cdot \mathbf{q} \right) dV. \tag{26}$$

The variables p and \mathbf{q} have the same basis functions as e_p and e_q respectively. Substituting in the discrete vectors, \hat{p} and \hat{q} and their corresponding basis functions gives

$$\begin{aligned} \hat{H}(\hat{p}, \hat{q}) &= \int_{\Omega} \left(\frac{1}{2\rho_w} \hat{p}^T \boldsymbol{\varphi}_p \boldsymbol{\varphi}_p^T \hat{p} + \frac{1}{2} k_w \hat{q}^T \boldsymbol{\varphi}_q \boldsymbol{\varphi}_q^T \hat{q} \right) dV \\ &= \frac{1}{2\rho_w} \hat{p}^T \mathbf{M}_p \hat{p} + \frac{1}{2} k_w \hat{q}^T \mathbf{M}_q \hat{q}. \end{aligned} \tag{27}$$

The relationship between the approximate state variable functions and the approximate effort functions from Equation (16) can be found by taking the partial derivative of $\hat{\mathcal{H}}(p, \mathbf{q})$ with respect to p and \mathbf{q} .

$$e_p = \partial_p \hat{\mathcal{H}} = \frac{p}{\rho_w}, \quad e_q = \partial_q \hat{\mathcal{H}} = k_w \mathbf{q}. \tag{28}$$

The corresponding relation between discrete effort variables and state vectors is

$$\hat{e}_p = \frac{\hat{p}}{\rho_w}, \quad \hat{e}_q = k_w \hat{q}. \tag{29}$$

To retain the structure of the continuous system, the discretised equations must have the same energy-conserving structure as the continuous equations. Therefore, the rate of change

of the Hamiltonian must only depend on the boundary variables, as in Theorem 2.0.1. The following conservation of energy proof is influenced by the thesis of Kotyczka [22] and the paper by Cardoso-Ribeiro [11].

Theorem 2.2.1 *The rate of change of the Hamiltonian for the discrete system (\hat{H}) is a dual product of the boundary efforts (\hat{e}_{pb} , \hat{e}_{qb}) and boundary flows (\hat{f}_{pb} , \hat{f}_{qb}) in either p or q . Put more simply, the change of energy in the domain equals the energy flow through the domain boundary.*

$$\dot{\hat{H}} = -\hat{e}_{pb}^T \hat{f}_{pb} = -\hat{e}_{qb}^T \hat{f}_{qb}.$$

To prove Theorem 2.2.1 a mapping between general variables and boundary variables must be formulated. Following the work of [22], we decompose L_p into T_p , S_q and L_q into T_q , S_p , as shown,

$$L_p = T_p^T S_q, \quad L_q = T_q^T S_p. \tag{30}$$

The matrix T_p is simply a mapping from all \hat{e}_p DOFs of the mesh to the DOFs that have \hat{e}_p defined at the boundary. Similarly, T_q is a mapping from all \hat{e}_q DOFs of the mesh to the DOFs that have \hat{e}_q defined at the boundary. Both T_p and T_q are made of zeroes and ones and are semi-orthogonal, therefore $T_p T_p^T = I_{N_p}$ and $T_q T_q^T = I_{N_q}$. For first order Lagrange and Raviart-Thomas elements, the identity matrix I_{N_p} has a number of rows and columns equal to the number of nodes (N_p) and I_{N_q} has a number of rows and columns equal to the number of edges (N_q). Calculation of S_p and S_q is done trivially by using the semi-orthogonality of $T_{p/q}$ and Equation (30), giving $S_q = T_p L_p$ and $S_p = T_q L_q$. The matrix S_p is a mapping from momentum efforts \hat{e}_p to boundary strain flows \hat{f}_{qb} . Similarly, S_q is a mapping from stress efforts \hat{e}_q to boundary velocity flows \hat{f}_{pb} . The matrix mappings of $T_{p/q}$ and $S_{p/q}$ are summarised as

$$\begin{aligned} \hat{e}_{pb} &= T_p \hat{e}_p, & \hat{e}_{qb} &= T_q \hat{e}_q, \\ \hat{f}_{pb} &= S_q \hat{e}_q, & \hat{f}_{qb} &= S_p \hat{e}_p. \end{aligned} \tag{31}$$

To prove Theorem 2.2.1 we need a small lemma.

Lemma 2.2.2 (Lemma 1 [24])

$$(K_p + L_p) + (K_q + L_q)^T = L_p.$$

Proof

$$\begin{aligned} (K_q + L_q)^T &= \left(-\langle \nabla \cdot \varphi_q | \varphi_p \rangle_\Omega + \langle \psi_q | \psi_p \rangle_{\partial\Omega} \right)^T, \\ (K_q + L_q)^T &= \left(-\langle \nabla \cdot \varphi_q | \varphi_p \rangle_\Omega + \langle (\varphi_q \cdot \mathbf{n}) | \psi_p \rangle_{\partial\Omega} \right)^T. \end{aligned}$$

Applying integration by parts in reverse then gives

$$\begin{aligned}
 (\mathbf{K}_q + \mathbf{L}_q)^T &= \left(\langle \boldsymbol{\varphi}_q | \nabla \boldsymbol{\varphi}_p \rangle_\Omega \right)^T \\
 &= \langle \nabla \boldsymbol{\varphi}_p | \boldsymbol{\varphi}_q \rangle_\Omega \\
 &= -\mathbf{K}_p,
 \end{aligned}
 \tag{32}$$

therefore,

$$(\mathbf{K}_p + \mathbf{L}_p) + (\mathbf{K}_q + \mathbf{L}_q)^T = (\mathbf{K}_p + \mathbf{L}_p) - \mathbf{K}_p = \mathbf{L}_p. \quad \square$$

Proof of Theorem 2.2.1 We start by taking the time differential of the Hamiltonian in the usual way.

$$\begin{aligned}
 \dot{H} &= \int_\Omega \left(\frac{\partial H}{\partial p} \frac{\partial p}{\partial t} + \frac{\partial H}{\partial \mathbf{q}} \cdot \frac{\partial \mathbf{q}}{\partial t} \right) dV \\
 &= -\langle \mathbf{e}_p | \mathbf{f}_p \rangle_\Omega - \langle \mathbf{e}_q | \mathbf{f}_q \rangle_\Omega.
 \end{aligned}$$

The approximate efforts and flows from Equations (15) and (16) are then substituted for the effort and flow functions to give the discrete Hamiltonian rate of change

$$\begin{aligned}
 \dot{H} &= -\hat{\mathbf{e}}_p^T \langle \boldsymbol{\varphi}_p | \boldsymbol{\varphi}_p \rangle_\Omega \hat{\mathbf{f}}_p - \hat{\mathbf{e}}_q^T \langle \boldsymbol{\varphi}_q | \boldsymbol{\varphi}_q \rangle_\Omega \hat{\mathbf{f}}_q \\
 &= -\hat{\mathbf{e}}_p^T \mathbf{M}_p \hat{\mathbf{f}}_p - \hat{\mathbf{e}}_q^T \mathbf{M}_q \hat{\mathbf{f}}_q.
 \end{aligned}$$

Equation (22) is then used to give

$$\begin{aligned}
 \dot{H} &= -\hat{\mathbf{e}}_p^T (\mathbf{K}_p + \mathbf{L}_p) \hat{\mathbf{e}}_q - \hat{\mathbf{e}}_q^T (\mathbf{K}_q + \mathbf{L}_q) \hat{\mathbf{e}}_p \\
 &= -\hat{\mathbf{e}}_p^T (\mathbf{K}_p + \mathbf{L}_p) \hat{\mathbf{e}}_q - \hat{\mathbf{e}}_p^T (\mathbf{K}_q + \mathbf{L}_q)^T \hat{\mathbf{e}}_q.
 \end{aligned}$$

Lemma 2.2.2 can now be used. Following that, the definitions of the boundary flows and efforts in Equation (31) can be used to complete the first half of the proof,

$$\begin{aligned}
 \dot{H} &= -\hat{\mathbf{e}}_p^T \mathbf{L}_p \hat{\mathbf{e}}_q \\
 &= -\hat{\mathbf{e}}_p^T \mathbf{T}_p^T \mathbf{S}_q \hat{\mathbf{e}}_q \\
 &= -(\mathbf{T}_p \hat{\mathbf{e}}_p)^T (\mathbf{S}_q \hat{\mathbf{e}}_q) \\
 &= -\hat{\mathbf{e}}_{pb}^T \hat{\mathbf{f}}_{pb},
 \end{aligned}
 \tag{33}$$

and identically for the second half of the proof,

$$\begin{aligned}
 \dot{H} &= -\hat{e}_p^T L_p \hat{e}_q \\
 &= -\hat{e}_p^T L_q^T \hat{e}_q \\
 &= -\hat{e}_q^T L_q \hat{e}_p \\
 &= -\hat{e}_q^T T_q^T S_p \hat{e}_p \\
 &= -(T_q \hat{e}_q)^T (S_p \hat{e}_p) \\
 &= -\hat{e}_{qb}^T \hat{f}_{qb}. \quad \square
 \end{aligned}$$

Corollary 2.2.3 *The rate of change of the Hamiltonian of the discrete system conserves energy in the same way as in Theorem 2.0.1, for the continuous system.*

Proof Beginning with Equation (33) and substituting in L_p from Equation (20) gives

$$\begin{aligned}
 \dot{H} &= -\hat{e}_p^T L_p \hat{e}_q \\
 &= -\int_{\partial\Omega} \hat{e}_p^T \psi_p \psi_q^T \hat{e}_q ds
 \end{aligned}$$

Substituting Equation (18) gives

$$\dot{H} = -\int_{\partial\Omega} e_p (e_q \cdot n) ds,$$

which confirms that the rate of change of the discrete system Hamiltonian is a function of the efforts at the boundary. This correctly coincides with Theorem 2.0.1. \square

The approximate system inputs and outputs are thus defined in the same way as Equations (9) and (10), either

$$u_q = -e_q \cdot n|_{\partial\Omega}, \quad y_p = e_p|_{\partial\Omega}, \tag{34}$$

or

$$y_q = e_q \cdot n|_{\partial\Omega}, \quad u_p = -e_p|_{\partial\Omega}. \tag{35}$$

2.3 Stokes-Dirac Structure and the Canonical Port-Hamiltonian Form

In Equation (22) we introduce the model in a non-canonical PH form, however, in this section we prove that Equation (22) is equivalent to a structure-preserving coupling of canonical Port-Hamiltonian forms. This also proves that the discretisation ensures a Stokes-Dirac structure [43], therefore, conserving the structure of the continuous equations, Equation (7). To formulate the system in an input-state-output PH form, we first define the input and output functions at the boundary corresponding to Equations (34) and (35) in terms of their discrete vectors $(\hat{u}_q, \hat{y}_p, \hat{u}_p, \hat{y}_q)$ and basis functions (θ_q, θ_p) .

$$\begin{aligned}
 u_q &= \langle \hat{e}_q, \psi_q \rangle = \langle \hat{u}_q, \theta_q \rangle, \\
 y_p &= \langle \hat{e}_p, \psi_p \rangle = \langle \hat{y}_p, \theta_p \rangle,
 \end{aligned} \tag{36}$$

$$\begin{aligned}
 u_p &= \langle \hat{e}_p, \psi_p \rangle = \langle \hat{u}_p, \theta_p \rangle, \\
 y_q &= \langle \hat{e}_q, \psi_q \rangle = \langle \hat{y}_q, \theta_q \rangle,
 \end{aligned}
 \tag{37}$$

where θ_p and θ_q contain the entries of ψ_p and ψ_q corresponding to the boundary DOFs. The first canonical PH form for Neumann boundary conditions can be set up by using

$$(\mathbf{K}_q + \mathbf{L}_q)^T = -\mathbf{K}_p,
 \tag{38}$$

which was formulated in Equation (32). Inserting the transpose of Equation (38) into Equation (22) gives

$$\begin{bmatrix} \mathbf{M}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_q \end{bmatrix} \begin{bmatrix} \hat{f}_p \\ \hat{f}_q \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{K}_p + \mathbf{L}_p \\ -\mathbf{K}_p^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{e}_p \\ \hat{e}_q \end{bmatrix}.
 \tag{39}$$

We then multiply the output equation in Equation (34) by the $v_{yq} = \langle \hat{v}_{yq} | \theta_q \rangle_{\partial\Omega_N}$ trial function, where a subscript N denotes a Neumann boundary. Following this, we integrate over the boundary then use the fundamental theorem of variational calculus to get,

$$\begin{aligned}
 \hat{v}_{yq} \int_{\partial\Omega_N} \theta_q y_p dS_N &= \hat{v}_{yq} \int_{\partial\Omega_N} \theta_q e_p |_{\partial\Omega_N} dS_N \\
 \rightarrow \hat{v}_{yq} \int_{\partial\Omega_N} \theta_q \theta_p^T \hat{y}_p dS_N &= \hat{v}_{yq} \int_{\partial\Omega_N} \theta_q \psi_p^T \hat{e}_p dS_N \\
 \rightarrow \mathbf{M}_{yp} \hat{y}_p &= \mathbf{B}_{qb}^T \hat{e}_p
 \end{aligned}
 \tag{40}$$

where $\mathbf{M}_{yp} = \langle \theta_q | \theta_p \rangle_{\partial\Omega_N}$. Also, \mathbf{B}_{qb} is defined as,

$$\mathbf{B}_{qb} \hat{u}_q = -\mathbf{L}_p \hat{e}_q, \quad \mathbf{B}_{qb} = \langle \psi_p | \theta_q \rangle_{\partial\Omega_N}.
 \tag{41}$$

Combining Equations (39) to (41) gives the PH canonical form,

$$\begin{bmatrix} \mathbf{M}_p & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_q & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_{yp} \end{bmatrix} \begin{bmatrix} \hat{f}_p \\ \hat{f}_q \\ \hat{y}_p \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{K}_p & -\mathbf{B}_{qb} \\ -\mathbf{K}_p^T & \mathbf{0} & \mathbf{0} \\ \mathbf{B}_{qb}^T & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{e}_p \\ \hat{e}_q \\ \hat{u}_q \end{bmatrix}.
 \tag{42}$$

Including the dynamic and constitutive laws from Equations (23) and (24) gives the input-state-output PH form,

$$\begin{bmatrix} -\mathbf{M}_p & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{M}_q & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_{yp} \end{bmatrix} \begin{bmatrix} \partial_t \hat{p} \\ \partial_t \hat{q} \\ \hat{y}_p \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{K}_p & -\mathbf{B}_{qb} \\ -\mathbf{K}_p^T & \mathbf{0} & \mathbf{0} \\ \mathbf{B}_{qb}^T & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \frac{1}{\rho_w} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & k_w \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{p} \\ \hat{q} \\ \hat{u}_q \end{bmatrix},$$

$$\mathbf{M}_{qb} \begin{bmatrix} \partial_t \hat{p} \\ \partial_t \hat{q} \\ \hat{y}_p \end{bmatrix} = \mathbf{J}_{qb} \mathbf{Q}_{qb} \begin{bmatrix} \hat{p} \\ \hat{q} \\ \hat{u}_q \end{bmatrix},
 \tag{43}$$

where \mathbf{M}_{qb} is the mass matrix, \mathbf{J}_{qb} is the skew-symmetric matrix, and \mathbf{Q}_{qb} is the constitutive law matrix. The infinite-dimensional canonical form corresponding to the discretised

canonical form in Equation (42) is

$$\mathcal{M}_{qb} \begin{bmatrix} f_p \\ f_q \\ y_p \end{bmatrix} = \mathcal{J}_{qb} \begin{bmatrix} e_p \\ e_q \\ u_q \end{bmatrix}, \tag{44}$$

where, \mathcal{M}_{qb} is the mass operator and \mathcal{J}_{qb} is the skew-symmetric operator.

Technically, Equations (43) and (44) are canonical forms for domains with Neumann conditions only. In the following we formulate the canonical form for Dirichlet conditions. We take the transpose of Equation (38) and use the fact that $L_q = L_p^T$ to give

$$(K_p + L_p)^T = -K_q. \tag{45}$$

Equation (45) can be substituted into Equation (22) and the discretisation of y_q can be done in the same way as for y_p in Equation (40) to give the Dirichlet boundary condition equivalent of Equation (42). This discrete canonical form is

$$\begin{bmatrix} M_p & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & M_q & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & M_{yq} \end{bmatrix} \begin{bmatrix} \hat{f}_p \\ \hat{f}_q \\ \hat{y}_q \end{bmatrix} = \begin{bmatrix} \mathbf{0} & -K_q^T & \mathbf{0} \\ K_q & \mathbf{0} & -B_{pb} \\ \mathbf{0} & B_{pb}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{e}_p \\ \hat{e}_q \\ \hat{u}_p \end{bmatrix}, \tag{46}$$

and the input-state-output PH form is

$$\begin{bmatrix} -M_p & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -M_q & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & M_{yq} \end{bmatrix} \begin{bmatrix} \partial_t \hat{p} \\ \partial_t \hat{q} \\ \hat{y}_q \end{bmatrix} = \begin{bmatrix} \mathbf{0} & -K_q^T & \mathbf{0} \\ K_q & \mathbf{0} & -B_{pb} \\ \mathbf{0} & B_{pb}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \frac{1}{\rho_w} I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & k_w I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I \end{bmatrix} \begin{bmatrix} \hat{p} \\ \hat{q} \\ \hat{u}_p \end{bmatrix},$$

$$M_{pb} \begin{bmatrix} \partial_t \hat{p} \\ \partial_t \hat{q} \\ \hat{y}_q \end{bmatrix} = J_{pb} Q_{pb} \begin{bmatrix} \hat{p} \\ \hat{q} \\ \hat{u}_p \end{bmatrix}, \tag{47}$$

where $M_{yq} = \langle \theta_p | \theta_q \rangle_{\partial\Omega_D}$ and a subscript D denotes a Dirichlet boundary. The canonical PH form in Equation (47) has a mass matrix, M_{pb} , a skew-symmetric matrix, J_{pb} , and a constitutive law matrix, Q_{pb} . The interconnection matrix, B_{pb} is defined as

$$B_{pb} \hat{u}_p = -L_q \hat{e}_p, \quad B_{pb} = \langle \psi_q | \theta_p \rangle_{\partial\Omega_D}. \tag{48}$$

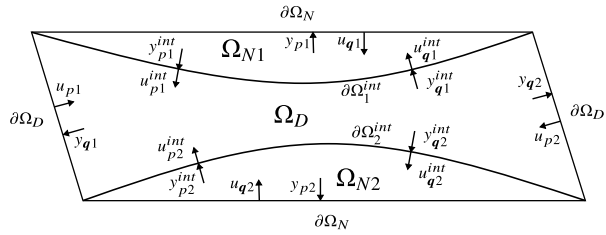
The infinite-dimensional canonical form corresponding to Equation (46) is

$$\mathcal{M}_{pb} \begin{bmatrix} f_p \\ f_q \\ y_q \end{bmatrix} = \mathcal{J}_{pb} \begin{bmatrix} e_p \\ e_q \\ u_p \end{bmatrix}, \tag{49}$$

where, \mathcal{M}_{pb} is the mass operator and \mathcal{J}_{pb} is the skew-symmetric operator.

Equations (47) and (49) are only canonical PH forms for boundaries with Dirichlet conditions. For a domain with mixed boundaries, our system in Equation (25) needs to be equivalent to a canonical PH system, or in our case a combination of canonical PH systems.

Fig. 1 Example of the subdivision of a domain with Neumann boundaries, $\partial\Omega_N$, and Dirichlet boundaries, $\partial\Omega_D$, showing boundary and interconnection inputs/outputs



Any closed domain with mixed Neumann and Dirichlet boundary conditions can be subdivided into subdomains with only Neumann or only Dirichlet boundary conditions. This idea has been taken advantage of in Brugnoli et al.’s work [9], where the authors numerically segment the domain and apply PFEM to each section. Here the idea is only used conceptually to prove that our system in Equation (25) is equivalent to a combination of canonical input-state-output PH formulations and therefore, by the compositionality property, retains the Stokes-Dirac structure of the analytic equations, Equation (7). This means that mixed boundary conditions can be implemented in the weak form, as detailed in Sect. 3, and that the resulting system is a Stokes-Dirac structure. It is also important to note that our formulation is non-degenerate, due to M_p and M_q being full rank. The matrices turn out to be full rank because we use the same basis functions for effort and flow functions. Proposition 1 of Kotyczka’s paper [24], uses a similar compositionality argument to allow mixed boundary conditions and mixed causality at the boundaries. However, in Kotyczka’s work, the non-full-rank M_p and M_q matrices cause the requirement of user-defined parameters in-order to form a non-degenerate Stokes-Dirac structure.

Theorem 2.3.1 Equation (25) over any connected domain with Neumann and Dirichlet boundaries is equivalent to a structure-preserving combination of canonical PH systems in the form of Equations (44) and (49). (Kotyczka makes the same statement in Proposition. 1 of [24] for a different set of function spaces.)

Proof First we subdivide the domain so that each connected Neumann boundary is in a domain denoted Ω_{Ni} that does not connect or overlap with any Ω_{Nj} ($j \neq i$) and is not connected to any other external boundaries. A simple example of a subdivision is shown in Fig. 1. Each of the i subdomains has inputs and outputs split up into u_{qi}, y_{pi} at the external boundary, and $u_{qi}^{int}, y_{pi}^{int}$ at the internal boundary. The remainder of the domain is therefore connected and only has Dirichlet boundary conditions, we denote this subdomain Ω_D .

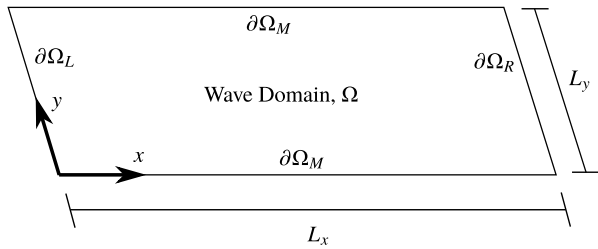
The inputs and outputs of Ω_D are split up into u_{pk}, y_{qk} at the external boundaries and $u_{pi}^{int}, y_{qi}^{int}$ at the internal boundaries that border Ω_{Ni} . The canonical form of Equation (44) is used in the Ω_{Ni} domains and Equation (49) is used in the Ω_D domain. Note that the canonical forms are modified accordingly, to split the inputs and outputs into internal and external parts. The causal interconnection relations [6, 24] at each internal boundary can then be written as

$$u_{pi}^{int} = -y_{pi}^{int}, \quad u_{qi}^{int} = y_{qi}^{int}. \tag{50}$$

This ensures conservation of energy flow between the subdomains due to the power conserving inner product,

$$\langle u_{pi}^{int} | y_{qi}^{int} \rangle_{\partial\Omega_i^{int}} + \langle u_{qi}^{int} | y_{pi}^{int} \rangle_{\partial\Omega_i^{int}} = 0. \tag{51}$$

Fig. 2 Rectangle wave domain diagram



Therefore, since each subdomain has a PH Stokes-Dirac structure, the power conserving interconnection ensures the total system is also a Stokes-Dirac structure by compositionality. Also, each subdomain has the correct canonical form for its Neumann or Dirichlet boundary conditions. Lastly, any connected domain is equivalent to a decomposition in the way we have described. The combination of these results proves that Equation (25) is equivalent to a structure-preserving combination of canonical PH forms for any connected domain with mixed boundary conditions. □

As will be seen in Sect. 6 the infinite-dimensional canonical form in Equation (49) can be used to determine the power conserving interconnection between other canonical PH systems, where the interconnection enforces a Dirichlet condition on the wave domain boundary. Similarly, the infinite-dimensional canonical form of Equation (44) can be used to find the power conserving interconnection for connections that assign a Neumann condition on the wave domain boundary. Due to the conclusion of Theorem 2.3.1, that any domain is equivalent to a subdivision of subdomains with either the canonical form of Equation (49) or Equation (44), both Dirichlet and Neumann interconnections can be implemented on a domain. This means that the system developed in this paper can have mixed boundary conditions as well as mixed causality interconnections between other PH systems.

3 Wave Equation FEniCS Implementation

In this section, the FEniCS implementation of Equations (14a) and (14b) on a rectangle domain is detailed. Appendix A supplements this section by detailing the Python code for the implementation. A schematic of the wave domain is shown in Fig. 2.

An unstructured, triangular mesh was created over the domain with FEniCS meshing software. The boundary conditions on the domain are set as

$$p_L = p(0, y, t) = \begin{cases} 10 \sin(8\pi t) & t < 0.25 \\ 0 & t \geq 0.25 \end{cases}, \tag{52a}$$

$$p_R = p(L_x, y, t) = 0, \tag{52b}$$

$$q_M = (\mathbf{q}(x, 0, t) \cdot \mathbf{n}) = (\mathbf{q}(x, L_y, t) \cdot \mathbf{n}) = 0, \tag{52c}$$

where the inputs in Equations (34) and (35) for the left, right, and middle boundaries are defined respectively as

$$u_{pL} = -\frac{p_L}{\rho_w}, \quad u_{pR} = -\frac{p_R}{\rho_w}, \quad u_{qM} = -k_w q_M. \tag{53}$$

Dirichlet conditions are applied to both the left and the right boundaries. The left boundary is set as an input condition whereas the right boundary is given a fixed zero value. The top and bottom boundaries have a zero-flux Neumann condition applied. Separating the boundary terms in Equations (14a) and (14b) for each boundary condition and reverting to integral rather than inner product notation gives

$$\int_{\Omega} v_p f_p dV = - \int_{\Omega} \nabla v_p \cdot \mathbf{e}_q dV + \int_{\partial\Omega_L} v_p (\mathbf{e}_q \cdot \mathbf{n}) ds_L \tag{54a}$$

$$+ \int_{\partial\Omega_R} v_p (\mathbf{e}_q \cdot \mathbf{n}) ds_R + \int_{\partial\Omega_M} v_p (\mathbf{e}_q \cdot \mathbf{n}) ds_M,$$

$$\int_{\Omega} \mathbf{v}_q \cdot \mathbf{f}_q dV = - \int_{\Omega} (\nabla \cdot \mathbf{v}_q) e_p dV + \int_{\partial\Omega_L} (\mathbf{v}_q \cdot \mathbf{n}) e_p ds_L \tag{54b}$$

$$+ \int_{\partial\Omega_R} (\mathbf{v}_q \cdot \mathbf{n}) e_p ds_R + \int_{\partial\Omega_M} (\mathbf{v}_q \cdot \mathbf{n}) e_p ds_M.$$

In Equations (54a) and (54b), the terms inside the boundary integrals are evaluated at their respective boundary. Substituting the state variables $f_p = -\dot{p}$, $\mathbf{f}_q = -\dot{\mathbf{q}}$, $e_p = p/\rho_w$, $\mathbf{e}_q = k_w \mathbf{q}$ and the boundary terms from Equations (52a)–(52c) gives

$$- \int_{\Omega} v_p \dot{p} dV = - k_w \int_{\Omega} \nabla v_p \cdot \mathbf{q} dV + k_w \int_{\partial\Omega_L} v_p (\mathbf{q} \cdot \mathbf{n}) ds_L \tag{55a}$$

$$+ k_w \int_{\partial\Omega_R} v_p (\mathbf{q} \cdot \mathbf{n}) ds_R + k_w \int_{\partial\Omega_M} v_p q_M ds_M,$$

$$- \int_{\Omega} \mathbf{v}_q \cdot \dot{\mathbf{q}} dV = - \frac{1}{\rho_w} \int_{\Omega} (\nabla \cdot \mathbf{v}_q) p dV + \frac{1}{\rho_w} \int_{\partial\Omega_L} (\mathbf{v}_q \cdot \mathbf{n}) p_L ds_L \tag{55b}$$

$$+ \frac{1}{\rho_w} \int_{\partial\Omega_R} (\mathbf{v}_q \cdot \mathbf{n}) p_R ds_R + \frac{1}{\rho_w} \int_{\partial\Omega_M} (\mathbf{v}_q \cdot \mathbf{n}) p ds_M.$$

These equations can be implemented in FEniCS, which automatically generates a matrix system of equations in the form of Equation (25). However, to solve these equations the system must also be discretised in time. Symplectic time integration schemes conserve the symplectic structure of the continuous equations and approximately conserve the Hamiltonian, therefore, they are the natural choice for the temporal discretisation. The symplectic Euler (SE) time integration scheme is applied to Equations (55a) and (55b) to give

$$- \int_{\Omega} v_p \frac{p - p^m}{\Delta t} dV = - k_w \int_{\Omega} \nabla v_p \cdot \mathbf{q}^m dV + k_w \int_{\partial\Omega_L} v_p (\mathbf{q}^m \cdot \mathbf{n}) ds_L \tag{56a}$$

$$+ k_w \int_{\partial\Omega_R} v_p (\mathbf{q}^m \cdot \mathbf{n}) ds_R + k_w \int_{\partial\Omega_M} v_p q_M ds_M,$$

$$- \int_{\Omega} \mathbf{v}_q \cdot \frac{\mathbf{q} - \mathbf{q}^m}{\Delta t} dV = - \frac{1}{\rho_w} \int_{\Omega} (\nabla \cdot \mathbf{v}_q) p dV + \frac{1}{\rho_w} \int_{\partial\Omega_L} (\mathbf{v}_q \cdot \mathbf{n}) p_L ds_L \tag{56b}$$

$$+ \frac{1}{\rho_w} \int_{\partial\Omega_R} (\mathbf{v}_q \cdot \mathbf{n}) p_R ds_R + \frac{1}{\rho_w} \int_{\partial\Omega_M} (\mathbf{v}_q \cdot \mathbf{n}) p ds_M,$$

where the m superscript denotes the variable at the previous time step. The L , R , and M subscripts denote variables at the left, right, and middle boundaries, respectively. The SE

scheme combines an explicit step for Equation (56a) and an implicit step for Equation (56b). When the Hamiltonian is separable the SE scheme is semi-explicit, meaning Equation (56b) could be solved explicitly after the solution of Equation (56a). However, due to the ease of implementation in FEniCS, the equations are solved in one implicit step. By Theorem 2.2.1, Equations (55a) and (55b) conserve the Hamiltonian. Combining this with SE integration gives a discrete system that retains the Hamiltonian structure of the continuum equations and conserves energy for large times, as further discussed in Sect. 3.1.

The energy bound on symplectic methods is, in general, proportional to $\mathcal{O}((\Delta t)^r)$, where r is the order of the time integration scheme [20]. To compare the energy bound between different order methods, a second-order symplectic scheme, the Störmer-Verlet (SV) method [19] is also implemented,

$$\begin{aligned}
 - \int_{\Omega} \mathbf{v}_q \cdot \frac{\mathbf{q}^{1/2} - \mathbf{q}^m}{0.5\Delta t} dV &= - \frac{1}{\rho_w} \int_{\Omega} (\nabla \cdot \mathbf{v}_q) p^m dV + \frac{1}{\rho_w} \int_{\partial\Omega_L} (\mathbf{v}_q \cdot \mathbf{n}) p_L^m ds_L \quad (57a) \\
 &+ \frac{1}{\rho_w} \int_{\partial\Omega_R} (\mathbf{v}_q \cdot \mathbf{n}) p_R^m ds_R + \frac{1}{\rho_w} \int_{\partial\Omega_M} (\mathbf{v}_q \cdot \mathbf{n}) p^m ds_M,
 \end{aligned}$$

$$\begin{aligned}
 - \int_{\Omega} v_p \frac{p - p^m}{\Delta t} dV &= - k_w \int_{\Omega} \nabla v_p \cdot \mathbf{q}^{1/2} dV + k_w \int_{\partial\Omega_L} v_p (\mathbf{q}^{1/2} \cdot \mathbf{n}) ds_L \quad (57b) \\
 &+ k_w \int_{\partial\Omega_R} v_p (\mathbf{q}^{1/2} \cdot \mathbf{n}) ds_R + k_w \int_{\partial\Omega_M} v_p q_M^{1/2} ds_M,
 \end{aligned}$$

$$\begin{aligned}
 - \int_{\Omega} \mathbf{v}_q \cdot \frac{\mathbf{q} - \mathbf{q}^{1/2}}{0.5\Delta t} dV &= - \frac{1}{\rho_w} \int_{\Omega} (\nabla \cdot \mathbf{v}_q) p dV + \frac{1}{\rho_w} \int_{\partial\Omega_L} (\mathbf{v}_q \cdot \mathbf{n}) p_L ds_L \quad (57c) \\
 &+ \frac{1}{\rho_w} \int_{\partial\Omega_R} (\mathbf{v}_q \cdot \mathbf{n}) p_R ds_R + \frac{1}{\rho_w} \int_{\partial\Omega_M} (\mathbf{v}_q \cdot \mathbf{n}) p ds_M.
 \end{aligned}$$

Solving Equations (57a)–(57c) requires two system of equation solves per time step, Equation (57a) is solved to get $\mathbf{q}^{1/2}$ then Equations (57b) and (57c) are solved for \mathbf{q} and p . Although in the general case the energy residual is bounded for symplectic methods, since the example we are modelling is linear we can improve this result and get exact energy conservation. We do this with the symplectic midpoint (SM) method, which conserves all quadratic invariants for linear systems [20]. The variational form of the SM scheme is

$$\begin{aligned}
 - \int_{\Omega} v_p \frac{p - p^m}{\Delta t} dV &= - k_w \int_{\Omega} \nabla v_p \cdot \frac{(\mathbf{q}^m + \mathbf{q})}{2} dV + k_w \int_{\partial\Omega_L} v_p \left(\frac{(\mathbf{q}^m + \mathbf{q})}{2} \cdot \mathbf{n} \right) ds_L \quad (58a) \\
 &+ k_w \int_{\partial\Omega_R} v_p \left(\frac{(\mathbf{q}^m + \mathbf{q})}{2} \cdot \mathbf{n} \right) ds_R + k_w \int_{\partial\Omega_M} v_p \frac{(q_M^m + q_M)}{2} ds_M,
 \end{aligned}$$

$$\begin{aligned}
 - \int_{\Omega} \mathbf{v}_q \cdot \frac{\mathbf{q} - \mathbf{q}^m}{\Delta t} dV &= - \frac{1}{\rho_w} \int_{\Omega} (\nabla \cdot \mathbf{v}_q) \frac{(p^m + p)}{2} dV + \frac{1}{\rho_w} \int_{\partial\Omega_L} (\mathbf{v}_q \cdot \mathbf{n}) \frac{(p_L^m + p_L)}{2} ds_L \quad (58b) \\
 &+ \frac{1}{\rho_w} \int_{\partial\Omega_R} (\mathbf{v}_q \cdot \mathbf{n}) \frac{(p_R^m + p_R)}{2} ds_R + \frac{1}{\rho_w} \int_{\partial\Omega_M} (\mathbf{v}_q \cdot \mathbf{n}) \frac{(p^m + p)}{2} ds_M.
 \end{aligned}$$

3.1 Wave Results

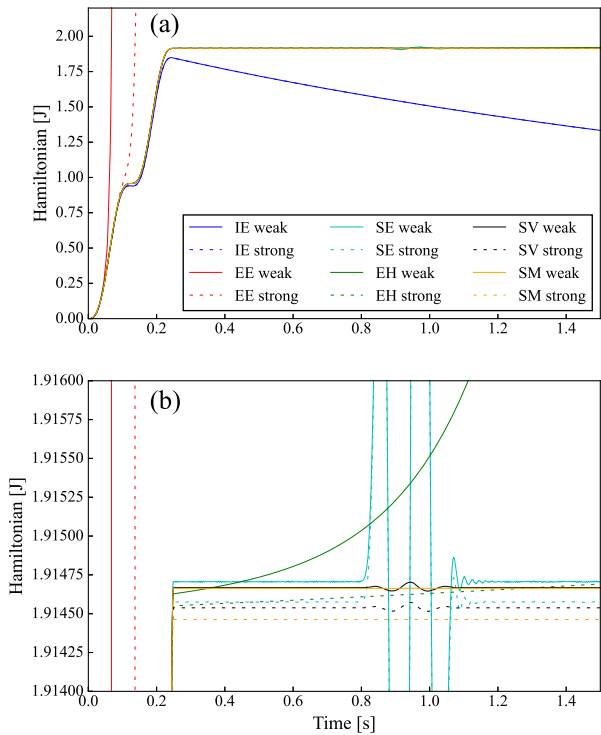
In this section, the model that is detailed in Sect. 3 and proven to conserve energy by Theorem 2.2.1 has been implemented with a range of different time integration schemes. Our method, which has a weak Dirichlet boundary condition implementation is compared to an implementation with Dirichlet boundary conditions implemented in the typical strong form. This is done to show that the naive setting of boundary conditions in a strong manner is detrimental to energy conservation. The strong Dirichlet implementation modifies the matrix system of equations generated by FEniCS to directly enforce the boundary condition values. This differs from our weak boundary Dirichlet implementation, where we apply the boundary conditions by specifying the boundary integral terms in Equations (55a) and (55b). A detailed example of the different boundary condition implementations in FEniCS is shown in Appendix A.

For the wave equation in PH form, setting p at the boundary is a Dirichlet condition, equivalent to setting $\rho_w e_p$, and setting $(\mathbf{q} \cdot \mathbf{n})$ is a Neumann condition, equivalent to setting $\frac{1}{k_w}(\mathbf{e}_q \cdot \mathbf{n})$. The reason that we can implement Dirichlet and Neumann conditions in the weak form is because we integrate both Equations (13a) and (13b) by parts, giving boundary terms for Dirichlet and Neumann conditions in Equations (55a) and (55b). This approach differs from the PFEM of Cardoso-Ribeiro [10], where integration by parts is only used on a subset of the governing equations. Our approach thus has the advantage of allowing mixed boundary conditions without the need for Lagrange multipliers. Therefore, our method results in a matrix system of equations that can be solved as an ordinary differential equation (ODE), rather than a differential algebraic equation (DAE), which is easier to solve in general. All methods in this section use a time step of $\Delta t = 5 \times 10^{-4}$. Figure 3 shows the resulting Hamiltonian, \hat{H} , for the input in Equations (52a)–(52c). The Hamiltonian is expected to be constant after 0.25 s, because the input boundary condition is set to zero, therefore, no energy flows into or out of the domain.

In Fig. 3(a) we show that the implicit Euler (IE) scheme incorrectly dissipates energy and the explicit Euler (EE) scheme has a fictitious increase in energy, resulting in instability. At $t \approx 0.9$ s the SE integration schemes for strong and weak boundary conditions show an undesirable non-constant Hamiltonian. A zoomed-in image of the non-constant behaviour is also shown in Fig. 3(b). The bump at $t \approx 0.9$ s is likely due to there being high-order dynamics when the wavefront approaches the right boundary that are not accounted for in the first-order SE scheme. To remedy this situation, three second-order integration schemes are implemented, Explicit Heun's (EH) (also called improved Euler [39]), SV and SM. All of these methods drastically decrease the bump at $t \approx 0.9$ s. It should be noted that decreasing the time step of the first-order methods also has the same effect of decreasing the bump (not shown). Interestingly, EH removes the bump completely, however, SV does not. SM also removes the bump completely because the SM scheme conserves quadratic invariants exactly for linear systems [20].

Figure 3(b) shows that for SE and SV the Hamiltonian oscillates about a conserved average Hamiltonian. This plot also shows that SV has an approximately constant Hamiltonian (apart from the aforementioned bump at $t \approx 0.9$ s), whereas EH has a gradually increasing Hamiltonian, indicating that energy is not conserved for long times. For SV, however, the Hamiltonian oscillates about a conserved value and therefore, is conserved for long times. The bound on the oscillations also converges towards the conserved value as the time step is decreased [20]. The energy bound for SE and SV is likely higher at the bump due to the reflecting boundary. This would hint that the non-perfect energy conservation of SE and SV is heavily influenced by the boundary and not so much by the internal domain. The SM

Fig. 3 (a) Hamiltonian for different time integration schemes and for strong and weak boundary condition implementations. (b) Zoomed in view. IE=Implicit Euler, EE=Explicit Euler, SE=Symplectic Euler, EH=Explicit Heun, SV=Störmer-Verlet, SM=Symplectic (Implicit) Midpoint



scheme has a perfectly flat Hamiltonian after 0.25 s, which is the expected result, again due to the conservation of quadratic invariants.

One problem of the first-order SE weak method in Fig. 3 is its oscillatory behaviour. These oscillations may be due to applying an essential boundary condition (EBC) weakly, without a penalty method. Typically EBC's are applied with a penalty method such as Nitsche [25, 30], however, according to Scovazzi, penalty methods are not required due to the hyperbolic nature of the wave equation [37]. This could mean that the oscillations may simply be the expected oscillations of low order symplectic time integration schemes. As anticipated by the proof of boundedness in [18], the oscillations are reduced when the time step is decreased (not shown) or the order of the symplectic method is increased, as seen with the SV method.

To compare the energy conservation of weak and strong boundary condition implementations, we define the energy residual at time t_f as

$$\begin{aligned}
 E_{res} &= \hat{H} + \int_{t_0}^{t_f} \int_{\partial\Omega_L} (\mathbf{e}_q^* \cdot \mathbf{n}) e_p^* ds_L dt \\
 &= \hat{H} + c^2 \int_{t_0}^{t_f} \int_{\partial\Omega_L} (\mathbf{q}^* \cdot \mathbf{n}) p_L^* ds_L dt,
 \end{aligned}
 \tag{59}$$

where \hat{H} , which is equal to the internal energy in the domain at time t_f , is calculated from Equation (27) with $\hat{\mathbf{q}}$ and $\hat{\mathbf{p}}$ variables at time t_f . Kotyczka et al. [23] shows that implicit Gauss-Legendre schemes such as SM conserve the discrete energy exactly for linear PH systems. However, for schemes such as SV and SE that do not conserve energy perfectly,

we expect an energy error from both the non-conservativity within the domain and a non-exact energy transferred through the boundary. Therefore, the second term in Equation (59) is simply the energy that we expect to have transferred through the left boundary out of the domain (the term is negative for flow into the domain) between the initial time t_0 and time t_f . Since we want to calculate the numerical energy flow out of the boundary, care must be taken to evaluate the output energy calculation in the same way as the numerical time integration scheme. This ensures that there is no discrepancy between the accuracy in which the two terms in Equation (59) are calculated. To ensure this consistent numerical accuracy, variables with a \star overscript are the effort or state variables evaluated at time steps that correspond with the chosen time integration scheme. For example, the expected energy contribution of the SV scheme at the boundary can be calculated by decomposing the SV method, as shown in a concise form here,

$$\frac{\mathbf{q}^{1/2} - \mathbf{q}^m}{0.5\Delta t} = \mathcal{F}(p^m), \tag{60a}$$

$$\frac{p - p^m}{\Delta t} = \mathcal{F}(\mathbf{q}^{1/2}), \tag{60b}$$

$$\frac{\mathbf{q} - \mathbf{q}^{1/2}}{0.5\Delta t} = \mathcal{F}(p), \tag{60c}$$

into two adjoint SE method steps with half timestep each [20], this gives

$$\frac{\mathbf{q}^{1/2} - \mathbf{q}^m}{0.5\Delta t} = \mathcal{F}(p^m), \quad \frac{p^{1/2} - p^m}{0.5\Delta t} = \mathcal{F}(\mathbf{q}^{1/2}), \tag{61a}$$

$$\frac{\mathbf{q} - \mathbf{q}^{1/2}}{0.5\Delta t} = \mathcal{F}(p), \quad \frac{p - p^{1/2}}{0.5\Delta t} = \mathcal{F}(\mathbf{q}^{1/2}), \tag{61b}$$

where $\mathcal{F}()$ denotes a function of the entries in the bracket. Then, by knowing that the superscripts of \mathbf{q}^\star and p_L^\star are the same as the superscripts of \mathbf{q} and p in Equation (61a) for the first half time step, and the same as in Equation (61b) for the second half time step. We can calculate the residual in two half time steps, as follows from Equation (59),

$$E_{res} = H^\star + c^2 \sum_{numSteps} \left(\int_{t^m}^t \int_{\partial\Omega_L} (\mathbf{q}^\star \cdot \mathbf{n}) p_L^\star ds_L dt \right),$$

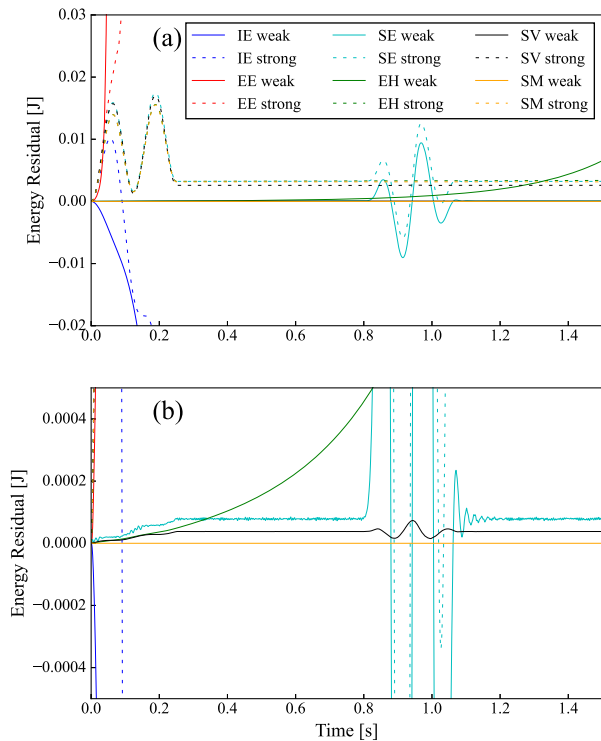
$$E_{res} = H^\star + c^2 \sum_{numSteps} \left(\int_{t^m}^{t^{1/2}} \int_{\partial\Omega_L} (\mathbf{q}^{1/2} \cdot \mathbf{n}) p_L^m ds_L dt + \int_{t^{1/2}}^t \int_{\partial\Omega_L} (\mathbf{q}^{1/2} \cdot \mathbf{n}) p_L ds_L dt \right),$$

$$E_{res} = H^\star + c^2 \sum_{numSteps} \left(0.5\Delta t \int_{\partial\Omega_L} (\mathbf{q}^{1/2} \cdot \mathbf{n}) p_L^m ds_L + 0.5\Delta t \int_{\partial\Omega_L} (\mathbf{q}^{1/2} \cdot \mathbf{n}) p_L ds_L \right). \tag{62}$$

The energy residual in Equation (59) is plotted for multiple time integration schemes in Fig. 4.

It is clear from the bounded energy residual of the SE and SV schemes and the exactly conserved energy of the SM scheme in Fig. 4 that the weak boundary condition implementation conserves energy for long times. This agrees with our proof of Theorem 2.2.1 and the expected bounded energy residual of symplectic time integrations schemes. Again, for the SE and SV schemes, the energy is conserved in an average sense and oscillations about the conserved energy do occur. As expected, the SM scheme conserves energy exactly, with an

Fig. 4 (a) Energy residual for different time integration schemes and for strong and weak boundary condition implementations. (b) Zoomed-in view. IE=Implicit Euler, EE=Explicit Euler, SE=Symplectic Euler, EH=Explicit Heun, SV=Störmer-Verlet, SM=Symplectic (Implicit) Midpoint



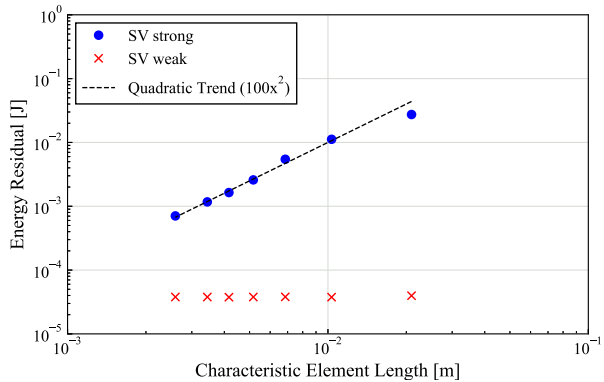
energy residual of $< 10^{-12}$, which is round-off error. The strong implementation of the input Dirichlet boundary condition does not conserve energy. More exactly, the strong implementation's energy residual is dependent on the refinement of the spatial mesh. To display this effect, the energy residual of the wave equation with SV time integration is analysed for varying element characteristic length ($\sqrt{\text{Mean Element Area}}$) and is shown in Fig. 5. A quadratic trend is plotted to show that the energy residual when using a strong boundary condition implementation has a quadratic dependence on the element characteristic length. This differs from the weak boundary condition implementation, which conserves energy independently of the mesh refinement. This agrees with our proof of Theorem 2.2.1, i.e., that our spatial discretisation is perfectly energy-conserving. Since the energy error is bounded for symplectic time integration schemes [19], the energy error for the weak boundary implementation is only dependent on the step size of the symplectic time integration scheme.

Temporal integration schemes cannot, in general, conserve both the exact energy and the symplectic structure of the system [20]. However, a general result that applies to non-linear systems, as well as our linear system, is that conserving the symplectic structure results in a bounded energy error which decreases as the time step is reduced. In subsequent chapters, we focus on the results of the SV scheme rather than the SM scheme to show energy conservation results that resemble what we expect for general non-linear problems.

4 Wave Equation Comparison with Analytical Solution

In this section, our numerical model with a weak boundary condition implementation is compared against an analytical solution to ensure stable spatial and temporal convergence

Fig. 5 Energy residual vs characteristic element length for strong and weak boundary condition implementations with an SV integration scheme. A quadratic trend of $y = 100x^2$ is plotted for comparison. SV=Störmer-Verlet



as well as good long-time behaviour and accurate eigenvalues. The analytic solution of the linear 2D wave equation was derived from one of the particular solutions in Sect. 7.1.1 of Polyanin and Nazaikinskii’s book [32]. The initial and boundary conditions are given as

$$p(x, y, 0) = \rho c \sqrt{\frac{\pi^2}{4L_x^2} + \frac{4\pi^2}{L_y^2}} \sin\left(\frac{\pi x}{2L_x} + \frac{\pi}{2}\right) \sin\left(\frac{2\pi y}{L_y} + \frac{\pi}{2}\right), \tag{63a}$$

$$p(0, y, t) = \rho c \sqrt{\frac{\pi^2}{4L_x^2} + \frac{4\pi^2}{L_y^2}} \sin\left(\frac{2\pi y}{L_y} + \frac{\pi}{2}\right) \cos\left(tc \sqrt{\frac{\pi^2}{4L_x^2} + \frac{4\pi^2}{L_y^2}}\right), \tag{63b}$$

$$p(L_x, y, t) = 0, \tag{63c}$$

$$\mathbf{q}(x, 0, t) \cdot \mathbf{n} = 0, \tag{63d}$$

$$\mathbf{q}(x, L_y, t) \cdot \mathbf{n} = 0. \tag{63e}$$

The analytical solution is then given by

$$p^a(x, y, t) = \rho c \sqrt{\frac{\pi^2}{4L_x^2} + \frac{4\pi^2}{L_y^2}} \sin\left(\frac{\pi x}{2L_x} + \frac{\pi}{2}\right) \sin\left(\frac{2\pi y}{L_y} + \frac{\pi}{2}\right) \cos\left(tc \sqrt{\frac{\pi^2}{4L_x^2} + \frac{4\pi^2}{L_y^2}}\right), \tag{64}$$

where all constants are given in Appendix D.

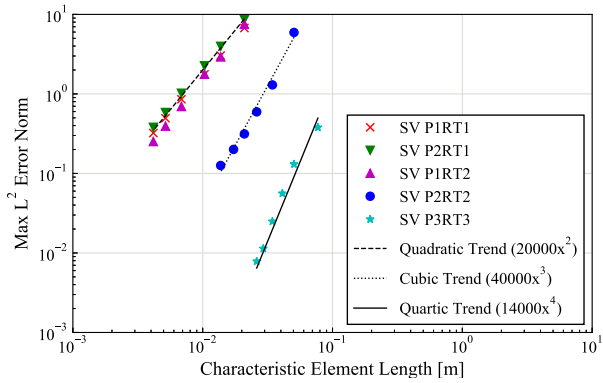
4.1 Spatial Convergence

The error between the numerical model and the analytic solution for a range of characteristic element lengths is evaluated to show the spatial convergence of the model. A table of the number of elements, with corresponding error and convergence details for each refinement level, is shown in Appendix E. The SV time integration scheme was used with a time step of 5×10^{-4} s. The L^2 error norm for each step n is defined as

$$E^n = \sqrt{\int_{\Omega} (p^n - p^a(t_n))^2 dV}, \tag{65}$$

where $p^a(t_n)$ is the exact solution evaluated at the step n . This error is calculated accurately with the ‘errornorm’ function in FEniCS. The maximum L^2 error norm over 1.5 s of

Fig. 6 Log-log plot of the maximum L^2 error norm of the numerical method for a 1.5 s simulation of the wave equation for varied characteristic element length and element order. SV=Störmer-Verlet



simulation is plotted in Fig. 6 for each characteristic element length. In the figure legend, the numbers following P and RT denote the order of the Lagrange and Raviart-Thomas elements, respectively, i.e., P1RT2 uses first order Lagrange elements and second order Raviart-Thomas elements. As can be seen, the L^2 error norm for all element order combinations shows standard convergence against characteristic element length. Here we define standard spatial convergence as convergence of order $\mathcal{O}((\Delta x_c)^{k+1})$, where k is the order of the methods lowest order basis function and Δx_c is the characteristic element length.

4.2 Modal Analysis

To ensure the correct handling of mixed boundary conditions, the eigenvalues of the model were verified to be accurate by performing a modal analysis. The analytic eigenvalues were calculated by separation of variables of Equation (1) into

$$w(x, y, t) = T(t)X(x)Y(y). \tag{66}$$

This method can simply be shown to give the following three first order eigenvalue problems,

$$\partial_t^2 T + c^2(\mu + \lambda)T = 0, \tag{67a}$$

$$\partial_x^2 X + \lambda X = 0, \tag{67b}$$

$$\partial_y^2 Y + \mu Y = 0, \tag{67c}$$

where λ and μ are eigenvalues of the first order spatial systems and $\omega^2 = c^2(\mu + \lambda)$ is the eigenvalue or the squared eigenfrequency that we want to predict. For the boundary conditions in Equations (63a)–(63e), the real components of the eigenvalues in Equations (67b) and (67c) are zero and the complex components are

$$\lambda = \left(\frac{n\pi}{L_x}\right)^2, \quad \forall n \in \mathbb{N}^+, \quad \mu = \left(\frac{m\pi}{L_y}\right)^2, \quad \forall m \in \mathbb{N}^0. \tag{68}$$

Therefore, from Equation (67a), the analytic eigenfrequencies we wish to find are given by

$$\omega = c \sqrt{\left(\frac{n\pi}{L_x}\right)^2 + \left(\frac{m\pi}{L_y}\right)^2}, \quad \forall n, m \text{ s.t. } n \in \mathbb{N}^+, m \in \mathbb{N}^0. \tag{69}$$

Fig. 7 Comparison of the complex part of the first 50 model and analytic eigenfrequencies

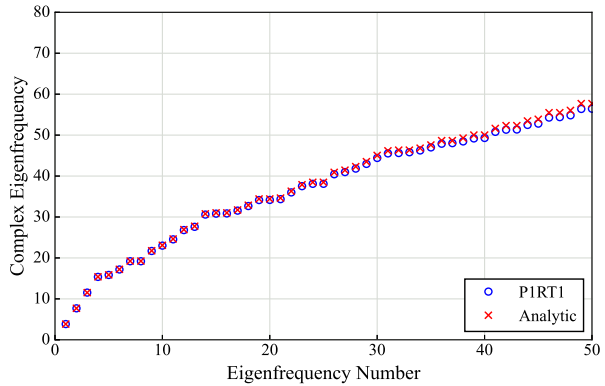
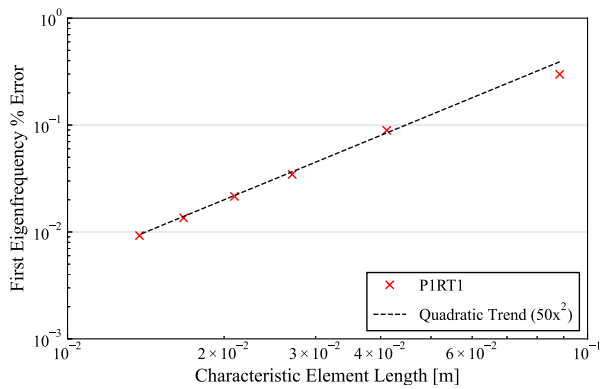


Fig. 8 Log-log plot of the first eigenfrequency percentage error when compared to the analytic eigenfrequency



To predict the eigenfrequencies of our model, Equations (55a) and (55b) were discretised in FEniCS, creating the following system of equations,

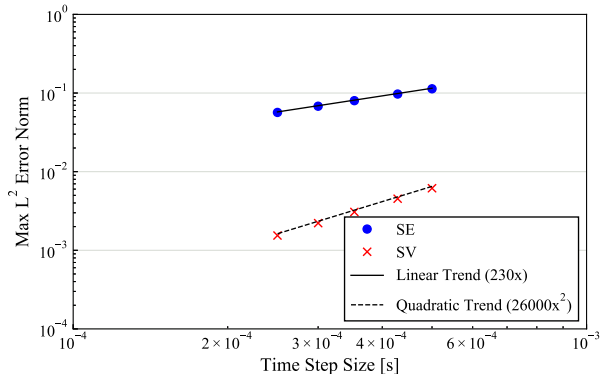
$$M_e \frac{\partial}{\partial t} \begin{bmatrix} \hat{p} \\ \hat{q} \end{bmatrix} = K_e \begin{bmatrix} \hat{p} \\ \hat{q} \end{bmatrix} + L_e \begin{bmatrix} \hat{p} \\ \hat{q} \end{bmatrix}, \tag{70a}$$

$$M_e = \begin{bmatrix} -M_p & \mathbf{0} \\ \mathbf{0} & -M_q \end{bmatrix}, \quad K_e = \begin{bmatrix} \mathbf{0} & k_w K_p \\ \frac{1}{\rho_w} K_q & \mathbf{0} \end{bmatrix}, \quad L_e = \begin{bmatrix} \mathbf{0} & k_w L_p \\ \frac{1}{\rho_w} L_q & \mathbf{0} \end{bmatrix}, \tag{70b}$$

where, M_e , K_e , and L_e are the mass, stiffness, and boundary matrices which determine the eigenfrequencies of the system. The $M_e^{-1}(K_e + L_e)$ matrix was then input into NumPy’s eigensolver to calculate the eigenfrequencies of the system. A plot of the complex part of the first 50 analytic and modelled eigenfrequency pairs for a mesh with 1322 elements is shown in Fig. 7. The real parts of all eigenfrequencies equal zero, as expected for the wave equation with no damping.

As shown in Fig. 7, the eigenfrequencies are predicted accurately with no spurious modes. A plot of the convergence of the eigenfrequencies with respect to the characteristic element length is shown in Fig. 8. This shows standard quadratic convergence for first order Lagrange and Raviart-Thomas elements.

Fig. 9 Log-log plot of the maximum L^2 error norm of the numerical method for a 1.5 s simulation of the wave equation for varied time step and 9358 elements. SE=Symplectic Euler, SV=Störmer-Verlet



4.3 Time Convergence

The numerical error for this model is heavily dominated by the spatial discretisation. Therefore, to show time convergence, third order Lagrange and Raviart-Thomas elements are used. This ensures that the initial spatial error is small, therefore, the error that propagates through the domain is due to the temporal discretisation. A convergence study is done for a 1.5 s simulation and an error growth study is assessed for long-times ($t = 10$ s). The long-time analysis shows the rate at which the error grows through time and displays our models effectiveness for long-time simulations. Although the rate of error growth for different symplectic schemes is well known [18], this section importantly shows that our spatial discretisation does not deteriorate the expected error growth rate.

The SE and SV time integration schemes have been implemented to show the temporal convergence of our method. Linear and quadratic convergence is shown for the SE and SV schemes, respectively in Fig. 9, where the maximum L^2 error over the 1.5 s simulation is plotted for a range of time steps. This shows that our method gives standard temporal convergence, where we define standard temporal convergence as convergence of order $\mathcal{O}((\Delta t)^r)$, where r is the order of the time integration scheme. These simulations were performed on a spatial discretisation with 9358 elements.

The order at which the state variable error of a method grows is a common metric for the accuracy of symplectic and multi-symplectic methods, as assessed extensively in Hairer's book [20]. Hairer shows that symplectic methods have state variable error growth of order $\mathcal{O}(t(\Delta t)^r)$, where r is the order of the time integration scheme. The time step convergence in Fig. 9 for fixed final time $t = 1.5$ s confirms that the error converges with order $\mathcal{O}((\Delta t)^r)$. Therefore, observing an error growth proportional to t when simulated for long times is sufficient to show the correct order of error growth, $\mathcal{O}(t(\Delta t)^r)$. The L^2 error norm for a 10 s simulation of the wave equation is shown in Fig. 10 for various time integration schemes. To give a fair comparison, the schemes have time steps that result in the same number of function evaluations, 0.00025 s for IE and SM and 0.0005 s for SV. To display a result that does not blow up immediately, the time step for EH is decreased even further to 0.000125 s.

In Fig. 10 the EH method blows up with exponential error growth because it does not have a bounded energy residual. This behaviour is typical of fully explicit methods, which can be unstable for long times [19]. The IE method has large error growth due to its inherent energy dissipation. This error increase tapers off (not shown) due to the complete loss of energy in the numerical model, this results in p^n approaching a constant zero throughout the domain. The symplectic methods both show an error that is linearly dependent on time, as

Fig. 10 L^2 error norm over a 10 s simulation for various time integration schemes. EH=Explicit Heun, IE=Implicit Euler, SV=Störmer-Verlet, SM=Symplectic (Implicit) Midpoint

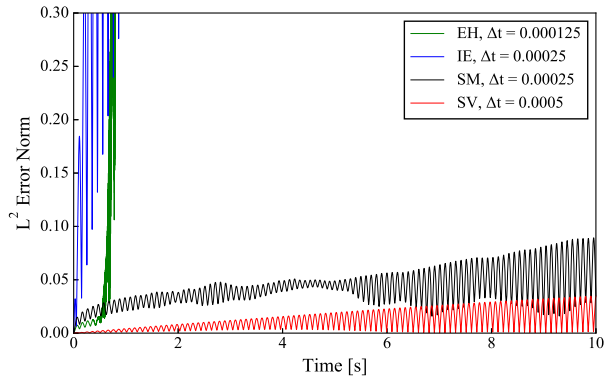
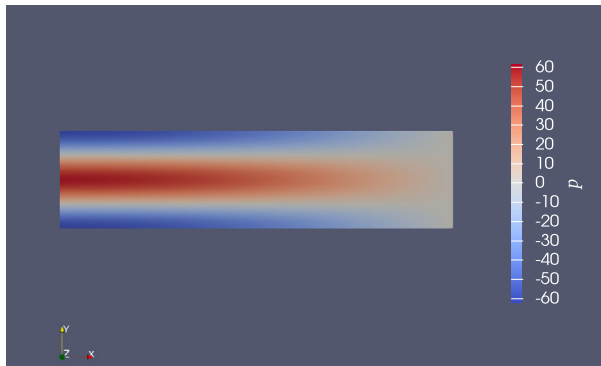


Fig. 11 p variable in wave domain after 0.5 s, for the model used to compare with the analytic solution, using Störmer-Verlet time integration



required to validate that the error growth is proportional to $\mathcal{O}(t(\Delta t)^r)$. The high-frequency oscillations of the L^2 error norm in Fig. 10 are due to p'' varying from being zero throughout the domain to having a maximum, as shown in Fig. 11. Therefore, the oscillation frequency is the frequency that p oscillates from maximum/minimum to zero.

5 Electromechanical Lumped Parameter Model

This section details a simple LPM for a linear actuated electric motor. The system diagram and bond-graph schematic are shown in Fig. 12, with constants defined in Appendix D. The current of the electrical system is i and the displacement of the linear motor is s . The bond-graph methodology is a modular approach for LPMs that ensure conservation of energy within and between models, for a review see the work by Gawthrop [16]. The PH framework extends from bond-graphs to also allow continuum models that conserve energy.

The Hamiltonian for this system is

$$H_{EM} = \frac{h_E^2}{2L_E} + \frac{h_M^2}{2m} + \frac{K_m s^2}{2}, \tag{71}$$

where the canonical momentum for the mechanical subsystem is given by $h_M = m\dot{s}$ and the electrical system equivalent of the canonical momentum is the magnetic flux linkage,

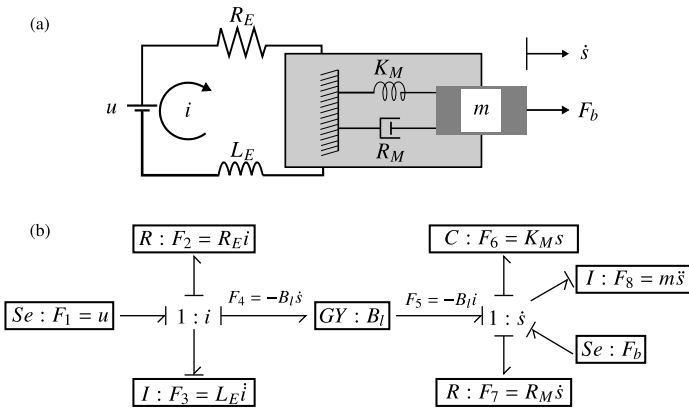


Fig. 12 (a) System diagram and (b) bond-graph diagram for an electromechanical system, where Se, R, I, C, and GY denote effort sources, dissipative components, inductive/mass storage components, capacitive/spring storage components, and gyrator components respectively

denoted by $h_E = L_E i$. The canonical PH form for this system is

$$\frac{\partial}{\partial t} \begin{bmatrix} h_E \\ h_M \\ s \end{bmatrix} = \left(\begin{bmatrix} 0 & B_l & 0 \\ -B_l & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} - \begin{bmatrix} R_E & 0 & 0 \\ 0 & R_M & 0 \\ 0 & 0 & 0 \end{bmatrix} \right) \begin{bmatrix} \frac{\partial H_{EM}}{\partial h_E} \\ \frac{\partial H_{EM}}{\partial h_M} \\ \frac{\partial H_{EM}}{\partial s} \end{bmatrix} + \mathbf{B}_u u + \mathbf{B}_F F_b, \tag{72}$$

$$y_u = \mathbf{B}_u^T \begin{bmatrix} \frac{\partial H_{EM}}{\partial h_E} \\ \frac{\partial H_{EM}}{\partial h_M} \\ \frac{\partial H_{EM}}{\partial s} \end{bmatrix}, \quad y_F = \mathbf{B}_F^T \begin{bmatrix} \frac{\partial H_{EM}}{\partial h_E} \\ \frac{\partial H_{EM}}{\partial h_M} \\ \frac{\partial H_{EM}}{\partial s} \end{bmatrix},$$

where \mathbf{B}_u is the input control matrix with corresponding input, u and output, y_u . Also, \mathbf{B}_F is the boundary port matrix with corresponding boundary force F_b and boundary output y_F . Including the constitutive laws and evaluating the \mathbf{B}_u and \mathbf{B}_F matrices gives

$$\frac{\partial}{\partial t} \begin{bmatrix} h_E \\ h_M \\ s \end{bmatrix} = \left(\begin{bmatrix} 0 & B_l & 0 \\ -B_l & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} - \begin{bmatrix} R_E & 0 & 0 \\ 0 & R_M & 0 \\ 0 & 0 & 0 \end{bmatrix} \right) \begin{bmatrix} \frac{1}{L_E} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & K_M \end{bmatrix} \begin{bmatrix} h_E \\ h_M \\ s \end{bmatrix} + \begin{bmatrix} u \\ F_b \\ 0 \end{bmatrix},$$

$$y_u = \frac{h_E}{L_E}, \quad y_F = \frac{h_M}{m}. \tag{73}$$

In this paper, the resistances, R_E and R_M , are set to zero, as a non-dissipative system is required to display a conserved Hamiltonian. The values of all constants used in the simulations are given in Appendix D. From Sect. 6 onwards F_b will be the reaction force from the coupled wave equation. The first part of Equation (73) can be written as a typical linear system of ODE's with a state vector $\mathbf{y} = [h_E, h_M, s]^T$ and a control/interconnection vector $\mathbf{u} = [u, F_b, 0]^T$, as

$$\dot{\mathbf{y}} = \mathbf{A} \mathbf{y} + \mathbf{u}, \tag{74}$$

$$A = \left(\left[\begin{array}{ccc} 0 & B_l & 0 \\ -B_l & 0 & -1 \\ 0 & 1 & 0 \end{array} \right] - \left[\begin{array}{ccc} R_E & 0 & 0 \\ 0 & R_M & 0 \\ 0 & 0 & 0 \end{array} \right] \right) \left[\begin{array}{ccc} \frac{1}{L_E} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & K_M \end{array} \right].$$

To create a monolithic coupling of this LPM with the wave equation from Sect. 3 the LPM needs to be implemented in FEniCS. The ODE in Equation (74) can be implemented in FEniCS by using the ‘real function space’, which assumes a function has one value over a domain, i.e., it has no spatial dependence. This makes the domain that the equations are implemented on irrelevant to the ODE. Equation (74) can then be implemented by multiplying by a trial function, v_y , and taking the trace against the border of an arbitrary domain. To allow easier coupling in the following section the trace is taken against the left boundary of the wave domain. The remainder of this section details the implementation of the LPM in FEniCS. Although in the following sections we use an SV time integration scheme, the SE scheme is detailed here to provide easier understanding to the reader. For the implementation of the SV method see the GitHub repo.

https://github.com/FinbarArgus/portHamiltonian_FEM.git

For the SE scheme, the ODE can be implemented by solving the matrix systems of equations generated by

$$\int_{\partial\Omega_L} v_y \cdot \frac{y - y^m}{\Delta t} ds_L = \int_{\partial\Omega_L} v_y \cdot (A y^a + u) ds_L, \tag{75}$$

at each time step. This method of implementing a LPM by assigning the variables as real functions over the whole domain is not optimally efficient and thus, future work should look at developing a method specifically for LPMs that is compatible with FEniCS. y^m is a vector of state variables at the previous time step and y^a is a vector of state variables at a combination of current and previous time steps, as shown here,

$$y^m = \begin{bmatrix} h_E^m \\ h_M^m \\ s^m \end{bmatrix}, \quad y^a = \begin{bmatrix} h_E \\ h_M \\ s^m \end{bmatrix}, \quad y = \begin{bmatrix} h_E \\ h_M \\ s \end{bmatrix}. \tag{76}$$

Once real function spaces are created and the vectors in Equation (76) are formed, the variational form in Equation (75) can be expressed and solved with the Python code in Appendix B.

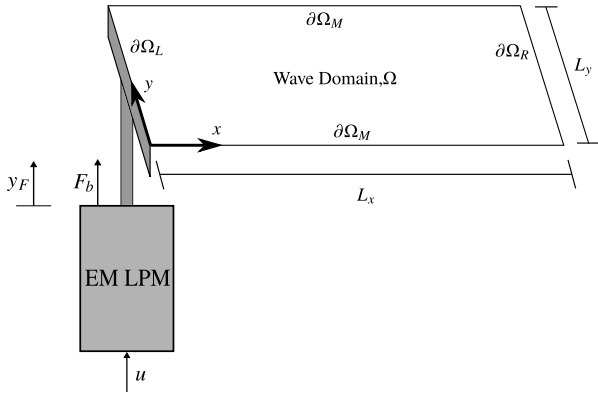
6 Coupling with the Electromechanical Model

In this section, the wave equation from Sect. 3 is coupled with the EM PH model from Sect. 5. A schematic of the combined model is shown in Fig. 13 for a rectangle wave domain.

Following the previous sections, the boundaries denoted $\partial\Omega_M$ and $\partial\Omega_R$ have zero Neumann and zero Dirichlet conditions, respectively. The $\partial\Omega_L$ boundary is the Dirichlet boundary connection with the LPM model. All boundary conditions are implemented in a weak manner. Since the only boundary connection is a Dirichlet boundary, the canonical form of Equation (49) can be used to set up the interconnection with Equation (73). The relationship between inputs and outputs of the wave domain and EM domain can be written as

$$u_p = -\mathcal{W}y_F, \quad F_b = \mathcal{W}^*y_q, \tag{77}$$

Fig. 13 System diagram for the coupled wave-electromechanical system



where \mathcal{W} is a compact operator and \mathcal{W}^* is its adjoint operator. The duality pairings for the inputs and outputs are

$$\langle u_p | y_q \rangle_{\partial\Omega}, \quad \langle F_b, y_F \rangle, \tag{78}$$

where the $\langle \cdot | \cdot \rangle_{\partial\Omega_L}$ duality product is an L^2 inner product that acts over the connection boundary and the $\langle \cdot, \cdot \rangle$ inner product is the standard inner product in \mathbb{R} . We know that the velocity of the left wave boundary is directly set by the output velocity of the EM system, therefore,

$$u_p = -\mathcal{W}y_F = -wy_F, \tag{79}$$

where w transforms a scalar into a constant function over the boundary, with the value of y_F . Substituting the boundary output relation from Equation (73) gives

$$u_p = -w \frac{h_M}{m}, \tag{80}$$

In practice, to turn the scalar value into a function it must be turned into a constant vector and then the vector must be dot producted with a vector of basis functions. Knowing that $u_p = \theta_p^T \hat{u}_p$ we get,

$$\theta_p^T \hat{u}_p = -w \frac{h_M}{m}, \tag{81}$$

Now, we enforce that every point on the left boundary of the wave domain has the same vertical velocity ($\hat{e}_p |_{\partial\Omega_L}$) as the vertical velocity of the output motor ($\frac{h_M}{m}$). Therefore, $\hat{u}_p = -\hat{e}_p |_{\partial\Omega_L}$ has the value of $-\frac{h_M}{m}$ for each of its DOFs, which gives

$$-\theta_p^T \mathbf{1} \frac{h_M}{m} = -w \frac{h_M}{m}, \tag{82}$$

where $\mathbf{1}$ is a column vector of size equal to the number of boundary \hat{u}_p DOFs that transforms the scalar $\frac{h_M}{m}$ into a vector with value $\frac{h_M}{m}$ for each DOF on $\partial\Omega_L$. Equating both sides of Equation (82) gives

$$w = \theta_p^T \mathbf{1} \tag{83}$$

The following energy conserving relation between the L^2 and \mathbb{R} inner products in Equation (78) can be used to determine \mathcal{W}^* ,

$$\langle \mathcal{W}y_F | y_q \rangle_{\partial\Omega_L} = \langle \mathcal{W}^*y_q, y_F \rangle. \tag{84}$$

Evaluating the left-hand side of Equation (84) gives

$$\begin{aligned} \langle \mathcal{W}y_F | y_q \rangle_{\partial\Omega_L} &= \int_{\partial\Omega_L} (wy_F)^T y_q dS_L \\ &= y_F \int_{\partial\Omega_L} w^T y_q dS_L. \end{aligned} \tag{85}$$

Equating to the right-hand side of Equation (84) gives

$$\begin{aligned} y_F \int_{\partial\Omega_L} wy_q dS_L &= \langle \mathcal{W}^*y_q, y_F \rangle \\ &= y_F \mathcal{W}^*y_q, \end{aligned} \tag{86}$$

which results in

$$F_b = \mathcal{W}^*y_q = \int_{\partial\Omega_L} wy_q dS_L. \tag{87}$$

Substituting y_q from Equation (35) and $e_q = k_w q$ gives

$$F_b = k_w \int_{\partial\Omega_L} w(q \cdot n) dS_L. \tag{88}$$

Therefore, Equations (80) and (88) are the energy conserving interconnection relations between the two domains. The total Hamiltonian of the combined system is given by

$$H_{total} = \hat{H} + H_{EM} = \int_{\Omega} \left(\frac{1}{2\rho_w} p^2 + \frac{1}{2} k_w q \cdot q \right) dV + \frac{h_E^2}{2L_E} + \frac{h_M^2}{2m} + \frac{K_M s^2}{2} \tag{89}$$

Finally, the canonical forms of Equations (47) and (73) can be combined to give the input-state-output PH form of the interconnected system,

$$\begin{aligned} &\begin{bmatrix} -M_p & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -M_q & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & 0 & 1 & 0 \\ \mathbf{0} & \mathbf{0} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \partial_t \hat{p} \\ \partial_t \hat{q} \\ \partial_t h_E \\ \partial_t h_M \\ \partial_t s \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0} & -K_q^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ K_q & \mathbf{0} & \mathbf{0} & D_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0 & B_l & 0 \\ \mathbf{0} & D_p^0 & -B_l & 0 & -1 \\ \mathbf{0} & \mathbf{0} & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \rho_w \mathbf{I} & \mathbf{0} & 0 & 0 & 0 \\ \mathbf{0} & k_w \mathbf{I} & 0 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \frac{1}{L_E} & 0 & 0 \\ \mathbf{0} & \mathbf{0} & 0 & \frac{1}{m} & 0 \\ \mathbf{0} & \mathbf{0} & 0 & 0 & K_M \end{bmatrix} \begin{bmatrix} \hat{p} \\ \hat{q} \\ h_E \\ h_M \\ s \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u, \\ &y_u = \frac{h_E}{L_E}, \end{aligned} \tag{90}$$

where D_p and D_p^0 are the interconnection matrices of the skew symmetric system matrix. For skew-symmetry of the system matrix to hold the following must be true

$$D_p^0 = -D_p^T. \tag{91}$$

To verify that this skew symmetry holds we calculate D_p and D_p^0 by discretising Equations (80) and (88). Firstly, Equation (80) is discretised by multiplying on the left by the trial basis function, ψ_q , and integrating over the boundary, as was done in the formulation of the boundary term in Equation (19),

$$\int_{\partial\Omega_L} \psi_q u_p ds_L = - \left(\int_{\partial\Omega_L} \psi_q \theta_p^T ds_L \right) \mathbf{1} \frac{h_M}{m} \tag{92}$$

$$= -\mathbf{B}_{pb} \mathbf{1} \frac{h_M}{m}, \tag{93}$$

where the final step comes from the definition of \mathbf{B}_{pb} in Equation (48). Equating this with the boundary term in the second row of Equation (90) gives

$$\begin{aligned} D_p \frac{h_M}{m} &= -\mathbf{B}_{pb} \mathbf{1} \frac{h_M}{m}, \\ \rightarrow D_p &= -\mathbf{B}_{pb} \mathbf{1}. \end{aligned} \tag{94}$$

To discretise Equation (88), we first substitute Equation (83) and apply $w = w^T$, because w is a constant function,

$$F_b = k_w \int_{\partial\Omega_L} (\theta_p^T \mathbf{1})^T (\mathbf{q} \cdot \mathbf{n}) ds_L. \tag{95}$$

Discretising $\mathbf{q} \cdot \mathbf{n}$ in the same way as $e_q \cdot \mathbf{n}$ was discretised in Equation (18) gives,

$$\begin{aligned} F_b &= \mathbf{1}^T k_w \left(\int_{\partial\Omega_L} \theta_p \psi_q^T ds_L \right) \hat{\mathbf{q}} \\ &= \mathbf{1}^T k_w \mathbf{B}_{pb}^T \hat{\mathbf{q}}, \end{aligned} \tag{96}$$

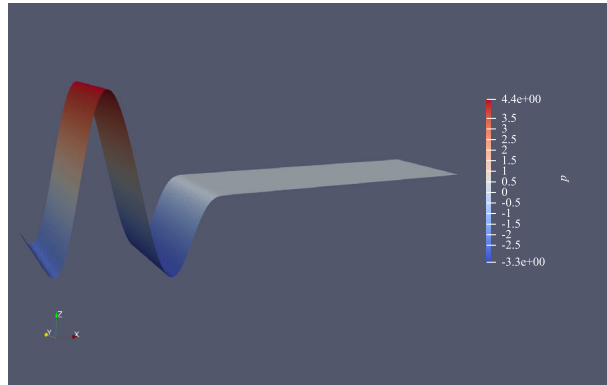
where $\mathbf{1}^T$ simply adds up the force contribution from each boundary DOF to calculate F_b . Equating the boundary term in the fourth row of Equation (90) with the boundary term in the second row of Equation (73) gives

$$\begin{aligned} k_w D_p^0 \hat{\mathbf{q}} &= F_b \\ k_w D_p^0 \hat{\mathbf{q}} &= k_w \mathbf{1}^T \mathbf{B}_{pb}^T \hat{\mathbf{q}}. \\ D_p^0 &= \mathbf{1}^T \mathbf{B}_{pb}^T \\ \rightarrow D_p^0 &= -D_p^T. \end{aligned} \tag{97}$$

Therefore, the discrete interconnected system retains a skew symmetric matrix, as required for a port Hamiltonian system. Equation (90) encompasses the canonical form, the dynamic equations and the constitutive law equations of the interconnected system.

It is important to note that if the right-side boundary condition, p_R , was non-zero then there would be an extra boundary input and output in the canonical form of Equation (90).

Fig. 14 p variable in the wave domain after 0.3 s of a wave-EM simulation, using Störmer-Verlet time integration



Also, if the top and bottom Neumann boundaries were non-zero conditions, the canonical form of Equation (90) would have interconnection inputs u_p^{int} and outputs y_q^{int} with the same definitions as the discrete version of Equation (35). A canonical form for the Neumann conditions would also have to be created with inputs/outputs for the boundary u_q, y_p and for the interconnection u_q^{int}, y_p^{int} with the same definitions as the discrete version of Equation (34). The two canonical forms could then be connected with Equation (50). Although the formulation of the interconnected canonical form seems complicated, it is not necessary for implementation. The full system canonical form is only discussed to reassure the reader that the total system is equivalent to a combination of canonical forms and therefore is a Stokes-Dirac structure and can have mixed causality boundary connections. The usefulness of the canonical forms, Equations (47) and (73), is that they allow calculation of the interconnection relations, which can then be implemented in FEniCS, as shown in Appendix C.

7 Interconnection Model Results

This section displays the results for the coupled wave-EM model for a sinusoidal input voltage. The time step for the simulations is 5×10^{-4} s, using an SV time integration scheme. The problem is solved on both a simple rectangle domain with 21110 elements and a square domain with central input and 13067 elements.

7.1 Rectangle Domain

The input voltage condition for the rectangle domain is given by

$$u = \begin{cases} 5 \sin(8\pi t) \text{ [V]} & t < 0.25 \text{ [s]} \\ 0 \text{ [V]} & t \geq 0.25 \text{ [s]} \end{cases} \quad (98)$$

The p variable of the wave equation is displayed after 0.3 s and 1.1 s in Figs. 14 and 15, respectively. The initial large sinusoidal wave, which is due to the input over the first 0.25 s, flows through the domain as expected. This wave is followed by smaller repeating waves caused by the lingering oscillations of the EM system.

To confirm that the energy error of the model is bounded, the energy residual of each domain and the total energy residual is plotted in Fig. 16. The residual for each domain is

Fig. 15 p variable in the wave domain after 1.1 s of a wave-EM simulation, using Störmer-Verlet time integration

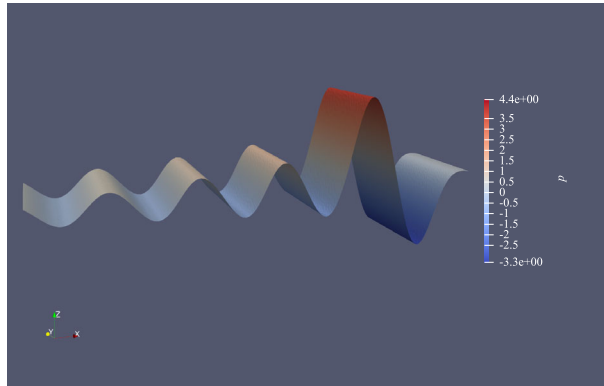
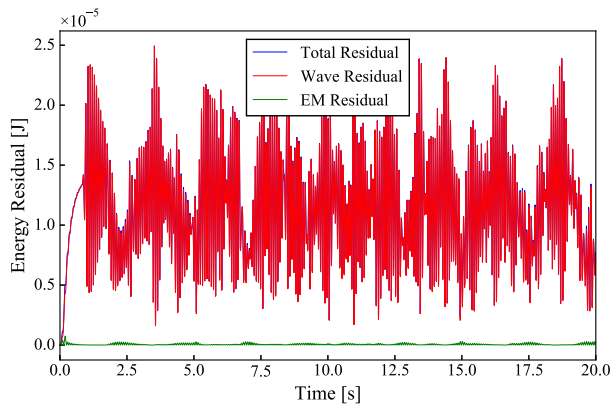


Fig. 16 Energy residual vs time for the interconnected wave-EM simulation, using Störmer-Verlet time integration. Total Residual is the sum of the energy residuals from the wave and EM domains. Wave residual is the difference between the energy in the wave domain and the accumulated energy that has entered the wave domain through its boundaries. EM residual is the difference between the energy in the EM domain and the accumulated energy that has entered the EM domain through its boundaries



the difference between the accumulated energy that has entered the domain and the internal energy in the domain.

As shown, the energy residual is bounded for increasing times, as expected for a symplectic time integration scheme. For a second-order symplectic method such as SV, the energy residual bound should be quadratically dependent on the time step, this relationship is proven in [18] and expressed more generally as

$$H(q(t), p(t)) = H(q(0), p(0)) + \mathcal{O}((\Delta t)^r). \tag{99}$$

To ensure that the bounded energy residual is indeed quadratically dependent on the time step, the energy residual maximum over a 20 s simulation is plotted against the step size in Fig. 17. As discussed in Sect. 4 The SM scheme can also be used to conserve quadratic invariants exactly. As shown in Fig. 17 we achieve an energy residual of $< 10^{-11}$ for a 20 s simulation with SM, confirming that our model, with an SM scheme, conserves energy exactly for linear, coupled LPM-continuum systems.

7.2 Square Domain with Central Input Boundary

A diagram for the domain of this section is shown in Fig. 18 and the input voltage is the same as Equation (98). Following the notation of the previous sections, the boundaries denoted

Fig. 17 Maximum energy residual for the interconnection model with a varied time step size. SV=Störmer-Verlet, SM=Symplectic (Implicit) Midpoint

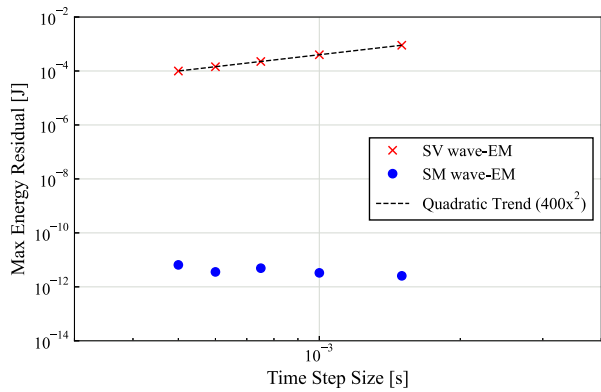
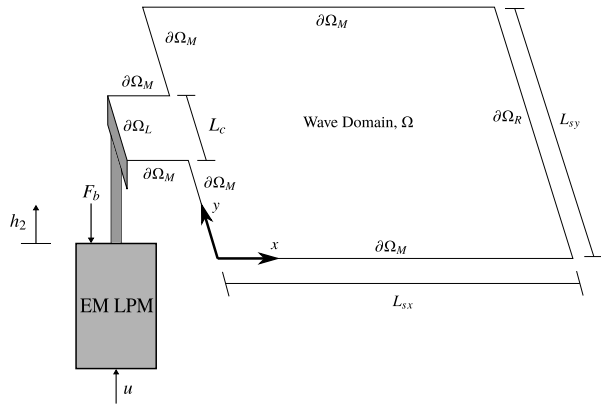


Fig. 18 System diagram for the coupled wave-electromechanical system with a square domain and a central input



$\partial\Omega_M$ and $\partial\Omega_R$ have zero Neumann conditions and zero Dirichlet conditions, respectively. Again, the $\partial\Omega_L$ boundary is the connection boundary with the LPM model and all boundary conditions are implemented in the weak form.

The p variable of the wave equation is displayed after 0.4 s and 0.6 s in Figs. 19 and 20, respectively. As can be seen, this example shows realistic 2D dispersion of a wave entering a square domain.

At long times the uniform circular wave structure breaks down due to waves rebounding off the walls. However, the wave behaviour should still retain some structure, for example, there should be symmetry about the midline parallel to the x axis. Figure 21 shows p^n after 8 s of simulation, once the uniform wavefronts have completely broken down. As can be seen, there is still symmetry about the midline parallel to the x axis, further showing this methods ability to accurately model the physical structure of the system.

The energy residual for an 8 s simulation with the SV scheme is shown in Fig. 22. As in previous sections, the energy residual is oscillatory and bounded, as expected. Also, as in the previous sections, when using the SM method the energy residual is $< 10^{-12}$, therefore, exactly conserved to within round-off error.

Fig. 19 p variable in a square wave domain after 0.4 s of the wave-EM simulation, using Störmer-Verlet time integration

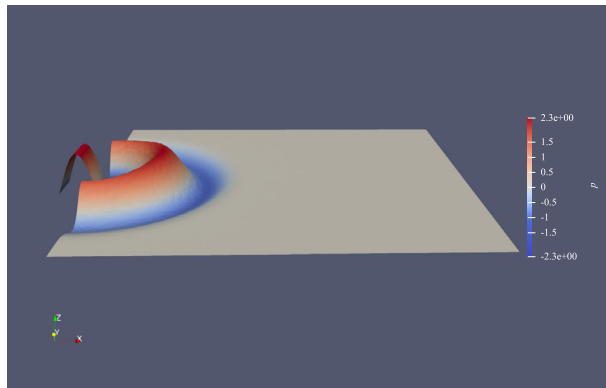


Fig. 20 p variable in a square wave domain after 0.6 s of the wave-EM simulation, using Störmer-Verlet time integration

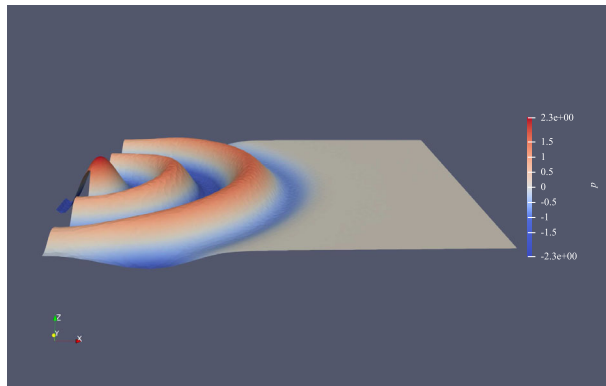
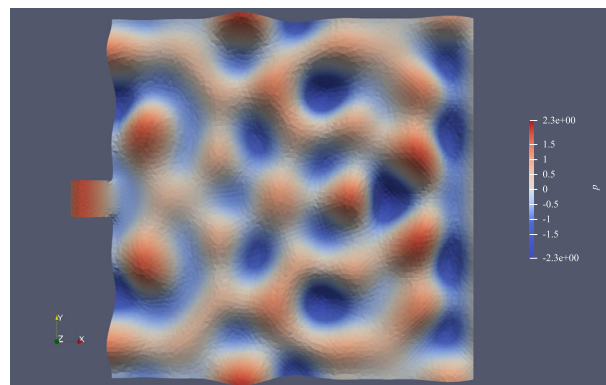


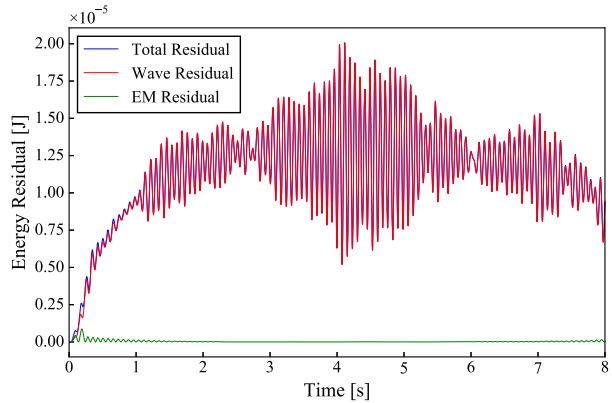
Fig. 21 p variable in a square wave domain after 8 s of the wave-EM simulation, using Störmer-Verlet time integration



8 Conclusion

For the modelling of continuum systems, a port-Hamiltonian, Galerkin finite element method has been developed that, in general, has a bounded energy residual and linear long-time error growth for the state variables. This method allows mixed boundary conditions without the need for Lagrange multipliers or user-defined parameters. The discretisation is shown to be equivalent to a coupling of canonical port-Hamiltonian forms that allows

Fig. 22 Energy residual vs time for the interconnected wave-EM simulation with a square domain, using Störmer-Verlet time integration. Total Residual is the sum of the energy residuals from the wave and EM domains. Wave residual is the difference between the energy in the wave domain and the accumulated energy that has entered the wave domain through its boundaries. EM residual is the difference between the energy in the EM domain and the accumulated energy that has entered the EM domain through its boundaries



mixed interconnections with other canonical port-Hamiltonian models. The discretisation is also shown to be symplectic in both time and space. For our specific 2D linear wave equation system we also show exact energy conservation with a symplectic (implicit) midpoint method that guarantees conservation of quadratic first integrals. We also compare against an analytical solution and show standard order of convergence for the state variables with respect to the temporal and spatial discretisation. A modal analysis is performed and the eigenvalues are verified to be accurate. The boundary conditions are implemented in variational form, in a weak manner, without the need for penalty methods. In addition to this, the method is capable of monolithic coupling with arbitrary LPMs. The coupled model is shown to also have a bounded energy residual with a standard temporal order of convergence for the SV time integration scheme and exact energy conservation for the SM time integration scheme. The example model of a 2D linear wave equation coupled with an EM linear actuator is a good proof of concept for more advanced couplings between Hamiltonian PDEs and LPMs. Future work will be done on implementing control algorithms for coupled models, in order to improve control of multiphysics, multidomain, and non-linear problems.

Appendix A: FEniCS Code Example

To give the reader details on how the variational forms in Sect. 3 are implemented in FEniCS, this section shows code snippets for the SE scheme with weak form and strong form boundary conditions. The full code can be found at

https://github.com/FinbarArgus/portHamiltonian_FEM.git

Firstly FEniCS and the meshing software mshr are imported, then the domain shown in Fig. 2 is created with the following lines of code,

```
from fenics import *
import mshr
mainRectangle = mshr.Rectangle(Point(0.0, 0.0), Point(L_x, L_y))
mesh = mshr.generate_mesh(mainRectangle, res)
```

where `res` is the resolution of the mesh. The function spaces, trial functions, test functions and functions for the DOFs at the previous and current time steps can then be created as follows,

```

# create mixed function space of Lagrange and Raviart–Thomas Elements
P = FiniteElement('P', triangle, lagrange_basis_order)
RT = FiniteElement('RT', triangle, RT_basis_order)
element = MixedElement([P, RT])
U = FunctionSpace(mesh, element)

# Define trial and test functions
p, q = TrialFunctions(U)
v_p, v_q = TestFunctions(U)

# Define functions for solutions at previous and current time steps
u_m = Function(U)
p_m, q_m = split(u_m)

u_ = Function(U)
p_, q_ = split(u_)

```

To implement the Neumann conditions on the top and bottom boundaries, the DOFs on the relevant boundaries are marked as shown,

```

class TopBottomMarker(SubDomain):
    def inside(self, x, on_boundary):
        return on_boundary and (near(x[1], 0) or near(x[1], L_y))

# Initialise mesh function for weak boundaries. Facets are dim()-1
boundaryDomains = MeshFunction('size_t', mesh, mesh.topology().dim()-1)
# initialise subdomains to index an arbitrary index 3
boundaryDomains.set_all(3)

# Mark top and bottom boundaries with index 1
topBottomMark = TopBottomMarker()
topBottomMark.mark(boundaryDomains, 1)

```

For the weak implementation of the Dirichlet left and right boundaries, a similar marking method is done as shown,

```

# Left and right edge boundary condition for marking
class LeftMarker(SubDomain):
    def inside(self, x, on_boundary):
        return on_boundary and near(x[0], 0)

class RightMarker(SubDomain):
    def inside(self, x, on_boundary):
        return on_boundary and near(x[0], L_x)

if dirichletImp == 'weak':
    leftMark = LeftMarker()
    leftMark.mark(boundaryDomains, 0)

    rightMark = RightMarker()
    rightMark.mark(boundaryDomains, 2)

# Redefine ds so that ds(0) is left boundary,
# ds(1) is topBottom (middle) and ds(2) is right boundary
ds = Measure('ds', domain=mesh, subdomain_data=boundaryDomains)

```

The strong implementation of Dirichlet left and right boundaries differs by specifying forced and fixed boundaries, respectively, as shown,

```
# Left edge forced boundary condition for defining strong Dirichlet BC
def forced_boundary(x, on_boundary):
    return on_boundary and near(x[0], 0.0)

# Right edge fixed boundary condition for defining strong Dirichlet BC
def fixed_boundary(x, on_boundary):
    return on_boundary and near(x[0], L_x)
```

After setting an expression for the left boundary condition $p_L = p_L$, a list of the Dirichlet boundary conditions is created as follows,

```
if dirichletImp == 'strong':
    bc_forced = DirichletBC(U.sub(0), p_L, forced_boundary)

    # fixed boundary at right boundary
    bc_fixed = DirichletBC(U.sub(0), Constant(0.0), fixed_boundary)

    # List of strong boundary condition objects
    bcs = [bc_forced, bc_fixed]
```

The final clear difference between the boundary condition implementations can be seen when specifying the variational forms for the SE scheme as shown below. Note the extra terms for the left and right Dirichlet boundaries in the weak implementation.

```
if dirichletImp == 'strong':
    # Variational form of the wave equation with strong Dirichlet BCs
    F = (p - p_m)*v_p*dx + \
        dt*k_w*(-dot(q_m, grad(v_p))*dx + q_M*v_p*ds(1)) + \
        dot(q - q_m, v_q)*dx + \
        dt/rho_w*dot(grad(p), v_q)*dx

elif dirichletImp == 'weak':
    # Variational form of the wave equation with weak Dirichlet BCs
    # This is the implementation of Eq.~(56)
    F = (p - p_m)*v_p*dx + \
        dt*k_w*(-dot(q_m, grad(v_p))*dx +
        q_M*v_p*ds(1) +
        dot(q_m, n)*v_p*ds(0) +
        dot(q_m, n)*v_p*ds(2)) + \
        dot(q - q_m, v_q)*dx -
        dt/rho_w*(div(v_q)*p*dx +
        dot(v_q, n)*p_L*ds(0) +
        dot(v_q, n)*p_R*ds(2) +
        dot(v_q, n)*p*ds(1))
```

Note that q_M , p_L , and p_R are defined as FEniCS expressions equal to their counterpart q_M , p_L , and p_R definitions in Equations (52a)–(52c). To formulate the \mathbf{A} and \mathbf{B} matrices in $\mathbf{Ax} = \mathbf{B}$, where \mathbf{x} is the vector of state variables, we then implement the final lines,

```
# Separate the variational form into LHS and RHS
a = lhs(F)
b = rhs(F)

# Assemble matrices
A = assemble(a)
B = assemble(b)
```

```
# Apply boundary conditions to matrices if we have Dirichlet conditions
if dirichletImp == 'strong':
    [bc.apply(A, B) for bc in bcs]
```

Note that the Dirichlet boundary condition application in the final line modifies the A and B matrices by removing the rows and columns that correspond with the fixed or forced boundary DOFs and specifies the value of the DOF exactly. The system can finally be solved by iterating through the time steps as follows,

```
t = 0
for n in range(number_of_steps)
    # update time
    t += dt

    # Here you must update any time dependent expressions (omitted)

    #assemble B matrix with updated u_m
    B = assemble(b)

    # Apply boundary conditions to matrices
    if dirichletImp = 'strong':
        [bc.apply(A, B) for bc in bcs]

    solve(A, u_.vector(), B)

    # update solution at the previous time step for the next loop
    u_m.assign(u_)
```

Appendix B: Electromechanical Python Implementation

The following block of code shows the variational form used to implement the lumped parameter EM model, Equation (75), in FEniCS.

$$F = (-\text{dot}(v_y, y - y_m)/dt + \text{dot}(v_y, A.\text{dot}(y_a)) + v_y[0]*uInput + v_y[1]*F_b)*ds(0)$$

where A , v_y , y , y_m , and y_a correspond to A , v_y , y , y^m , and y^a , respectively. Also, $uInput$ and F_b are FEniCS expressions that correspond with u and F_b , respectively. The problem can then be solved by iterating through time steps, assembling the variational form, then solving the assembled matrix system for each time step, in the same way as at the end of Appendix A.

Appendix C: FEniCS Implementation of the Interconnection

In this section, since the scalar variables are implemented in FEniCS as constant functions, w , which turns scalar values into constant functions, can be omitted. The interconnection

condition for the wave domain input boundary, Equation (79), is implemented in FEniCS with the boundary variational form in Equation (55b). This variational form is

$$\int_{\partial\Omega_L} (\mathbf{v}_q \cdot \mathbf{n}) \frac{p_L}{\rho_w} dS_L = - \int_{\partial\Omega_L} (\mathbf{v}_q \cdot \mathbf{n}) u_p dS_L = \int_{\partial\Omega_L} (\mathbf{v}_q \cdot \mathbf{n}) y_F dS_L = \int_{\partial\Omega_L} (\mathbf{v}_q \cdot \mathbf{n}) \frac{h_M}{m} dS_L. \tag{100}$$

The variational form in Equation (75) that allows the EM domain input boundary term, Equation (87), to be applied in FEniCS is

$$\int_{\partial\Omega_L} v_{y2} F_b dS_L, \tag{101}$$

where v_{y2} is the 2nd component of \mathbf{v}_y , the ODE test function vector. Substituting Equation (87) into Equation (101) and taking v_{y2} out of the integral, since it is a real function space that is constant over the boundary, gives

$$\int_{\partial\Omega_L} v_{y2} \left(\int_{\partial\Omega_L} y_q dS_L \right) dS_L = v_{y2} \int_{\partial\Omega_L} \left(\int_{\partial\Omega_L} y_q dS_L \right) dS_L. \tag{102}$$

Now, since the inside integral evaluates to a constant over the boundary, the outside integral can be evaluated to give

$$v_{y2} L_y \int_{\partial\Omega_L} y_q dS_L. \tag{103}$$

Finally, substituting y_q from Equation (35) and $\mathbf{e}_q = k_w \mathbf{q}$ gives

$$k_w L_y v_{y2} \int_{\partial\Omega_L} (\mathbf{q} \cdot \mathbf{n}) dS_L. \tag{104}$$

The interconnection with boundary interconnection forms, Equations (100) and (104), can be implemented in FEniCS for the SE scheme with the following code,

```
# This is the variational form of the wave equation
F = (p - p_m)*v_p*dx + \
    dt*k_w*(-dot(q_m, grad(v_p))*dx + \
    q_M*v_p*ds(1) + \
    dot(q_m, n)*v_p*ds(0) + \
    dot(q_m, n)*v_p*ds(2)) + \
    dot(q - q_m, v_q)*dx - \
    dt/rho_w*(div(v_q)*p*dx + \
    dot(v_q, n)*p_R*ds(2) + \
    dot(v_q, n)*p*ds(1)) + \
    dt*dot(v_q, n)*y[1]/m*ds(0) # This line implements Eq.~(100)

# This is the variational form of the electromechanical system
F_em = (-dot(v_y, y - y_m)/dt + \
    dot(v_y, A.dot(y_a)) + \
    v_y[0]*uInput + \
    k_w*L_y*v_y[1]*dot(q_m, n)*ds(0) # This line implements Eq.~(104)

# Combine the two variational forms
F = F + F_em
```


Table 1 Material constants for wave equation and coupled wave-EM simulations

Parameter	Symbol	Value	Units
Wave domain density	ρ_w	2.0	kg m^{-3}
Wave domain stiffness	k_w	3.0	kg m s^{-2}
Wave speed squared	$c^2(k_w/\rho_w)$	1.5	$\text{m}^2 \text{s}^{-2}$
Wave domain x-length	L_x	1.0	m
Wave domain y-length	L_y	0.25	m
Square domain x-length	L_{sx}	1.0	m
Square domain y-length	L_{sy}	1.0	m
Square domain centre-length	L_c	0.1	m
Electrical resistance	R_E	0	$\Omega = \text{kg m}^2 \text{s}^{-1} \text{C}^{-2}$
Electrical inductance	L_E	0.1	$H = \text{kg m}^2 \text{C}^{-2}$
Gyrator constant	B_l	5.0	$\text{kg m s}^{-1} \text{C}^{-1}$
Mechanical damping	R_M	0	kg m s^{-1}
Mechanical stiffness	K_M	5.0	kg m s^{-2}
Mechanical mass	m	0.15	kg

Table 2 Spatial Discretisation properties for each refinement level for P1RT1, P2RT1, and P1RT2. RL=Refinement Level, NoE=Number of Elements, CEL=Characteristic Element Length, $L^2 = L^2$ Error Norm, OoC=Order of convergence from previous refinement

RL	NoE	CEL [m]	P1RT1		P2RT1		P1RT2	
			L^2	OoC	L^2	OoC	L^2	OoC
0	570	0.0209	6.75	0.00	8.62	0.00	7.51	0.00
1	1322	0.0138	3.03	1.90	3.97	1.84	2.92	2.25
2	2344	0.0103	1.76	1.90	2.26	1.97	1.77	1.75
3	5318	0.0069	0.86	1.76	1.02	1.94	0.69	2.28
4	9358	0.0052	0.50	1.94	0.58	1.97	0.39	2.01
5	14404	0.0042	0.32	1.99	0.38	1.99	0.25	2.07

Appendix D: Material Constants

The constants used for simulation of the wave equation and coupled wave-EM model are shown in Table 1.

Appendix E: Spatial Convergence Tables

The element numbers, characteristic element lengths, L^2 error norm and order of convergence from the previous refinement level for the spatial convergence in Sect. 4.1 are shown in Table 2, Table 3, and Table 4.

Table 3 Spatial Discretisation properties for each refinement level for P2RT2. RL=Refinement Level, NoE=Number of Elements, CEL=Characteristic Element Length, $L^2 = L^2$ Error Norm, OoC=Order of convergence from previous refinement

RL	NoE	CEL [m]	L^2	OoC
0	98	0.0505	5.93	0.00
1	211	0.0344	1.30	3.96
2	368	0.0261	0.59	2.81
3	570	0.0209	0.31	2.92
4	837	0.0173	0.20	2.34
5	1322	0.0138	0.13	2.04

Table 4 Spatial Discretisation properties for each refinement level for P3RT3. RL=Refinement Level, NoE=Number of Elements, CEL=Characteristic Element Length, $L^2 = L^2$ Error Norm, OoC=Order of convergence from previous refinement

RL	NoE	CEL [m]	L^2	OoC
0	42	0.0772	0.379	0.000
1	98	0.0505	0.131	2.510
2	148	0.0411	0.056	4.110
3	211	0.0344	0.025	4.576
4	291	0.0293	0.011	4.895
5	368	0.0261	0.008	3.138

Acknowledgements The authors acknowledge the help of Professor Robert McLachlan who advised on symplectic methods and Dr Gonzalo Maso Talou for reviewing the manuscript. This work was supported by the University of Auckland Doctoral Scholarship and the MedTech CoRE: Platform 5 grant.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Alnæs, M., Blechta, J., Hake, J., Johansson, A., Kehlet, B., Logg, A., Richardson, C., Ring, J., Rognes, M.E., Wells, G.N.: The FEniCS Project Version 1.5. Arch. Numer. Softw. **3**(100) (2015). <https://doi.org/10.11588/ans.2015.100.20553>
2. Bauer, W., Cotter, C.J.: Energy-entropy conserving compatible finite element schemes for the rotating shallow water equations with slip boundary conditions. J. Comput. Phys. **373**, 171–187 (2018). <https://doi.org/10.1016/j.jcp.2018.06.071>
3. Bazilevs, Y., Hughes, T.J.: Weak imposition of Dirichlet boundary conditions in fluid mechanics. Comput. Fluids **36**(1), 12–26 (2007). <https://doi.org/10.1016/j.compfluid.2005.07.012>
4. Brezzi, F., Fortin, M.: Mixed and hybrid finite element methods, SIAM Rev. **3** (1993). <https://doi.org/10.1137/1035113>
5. Brugnano, L., Frasca Caccia, G., Iavernaro, F.: Energy conservation issues in the numerical solution of the semilinear wave equation. Appl. Math. Comput. **270**, 842–870 (2015). <https://doi.org/10.1016/j.amc.2015.08.078>
6. Brugnoli, A., Alazard, D., Pommier-Budinger, V., Matignon, D.: Interconnection of the Kirchhoff plate within the port-Hamiltonian framework. In: IEEE Conference on Decision and Control, vol. 58, pp. 6857–6862 (2019). <https://doi.org/10.1109/CDC40024.2019.9029487>
7. Brugnoli, A., Alazard, D., Pommier-Budinger, V., Matignon, D.: Port-Hamiltonian formulation and symplectic discretization of plate models Part I: Mindlin model for thick plates. Appl. Math. Model. **75**, 940–960 (2019). <https://doi.org/10.1016/j.apm.2019.04.035>

8. Brugnoli, A., Alazard, D., Pommier-Budinger, V., Matignon, D.: Port-Hamiltonian formulation and symplectic discretization of plate models Part II: Kirchhoff model for thin plates. *Appl. Math. Model.* **75**, 961–981 (2019). <https://doi.org/10.1016/j.apm.2019.04.036>
9. Brugnoli, A., Cardoso-Ribeiro, F.L., Haine, G., Kotyczka, P.: Partitioned finite element method for power-preserving structured discretization with mixed boundary conditions. In: Proceedings of the 21st IFAC. World Congress, ??? (2020)
10. Cardoso-Ribeiro, F.L., Matignon, D., Lefèvre, L.: A structure-preserving Partitioned Finite Element Method for the 2D wave equation. *IFAC-PapersOnLine* **51**(3), 119–124 (2018). <https://doi.org/10.1016/j.ifacol.2018.06.033>
11. Cardoso-Ribeiro, F.L., Matignon, D., Lefèvre, L.: A partitioned finite element method for power-preserving discretization of open systems of conservation laws. *IMA J. Math. Control Inf.* **38**(2), 493–533 (2020). <https://doi.org/10.1093/imamci/dnaa038>
12. Cockburn, B., Gopalakrishnan, J., Lazarov, R.: Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. *SIAM J. Numer. Anal.* **47**(2), 1319–1365 (2009). <https://doi.org/10.1137/070706616>
13. Duck, F.: Tissue non-linearity. *J. Eng. Med.* **224**(2), 155–170 (2010). <https://doi.org/10.1243/09544119JEIM574>
14. Duindam, V., Macchelli, A., Stramigioli, S., Bruyninckx, H.: Modeling and Control of Complex Physical Systems: The Port-Hamiltonian Approach (2009). <https://doi.org/10.1007/978-3-642-03196-0>
15. Eldred, C., Dubos, T., Kritsikis, E.: A quasi-Hamiltonian discretization of the thermal shallow water equations. *J. Comput. Phys.* **379**, 1–31 (2019). <https://doi.org/10.1016/j.jcp.2018.10.038>
16. Gawthrop, P.J., Bevan, G.P.: Bond-graph modeling. *IEEE Control Syst. Mag.* **27**(2), 24–45 (2007). <https://doi.org/10.1109/MCS.2007.338279>
17. Golo, G., Talasila, V., Van der Schaft, A., Maschke, B.: Hamiltonian discretization of boundary control systems. *Automatica* **40**(5), 757–771 (2004). <https://doi.org/10.1016/j.automatica.2003.12.017>
18. Hairer, E., Lubich, C.: Symmetric multistep methods over long times. *Numer. Math.* **97**(4), 699–723 (2004). <https://doi.org/10.1007/s00211-004-0520-2>
19. Hairer, E., Lubich, C., Wanner, G.: Geometric numerical integration illustrated by the Störmer-Verlet method. *Acta Numer.* **12**, 399–450 (2003). <https://doi.org/10.1017/S0962492902000144>
20. Hairer, E., Lubich, C., Wanner, G.: Geometric Numerical Integration - Structure-Preserving Algorithms for Ordinary Differential Equations (2006). <https://doi.org/10.1007/3-540-30666-8>
21. Joly, P.: Variational methods for time-dependent wave propagation problems. In: Topics in Computational Wave Propagation, pp. 201–264 (2003)
22. Kotyczka, P.: Numerical Methods for Distributed Parameter Port-Hamiltonian Systems (2019). <https://doi.org/10.14459/2019md1510230>
23. Kotyczka, P., Lefèvre, L.: Discrete-time port-Hamiltonian systems: a definition based on symplectic integration. *Syst. Control Lett.* **133**, 104530 (2018)
24. Kotyczka, P., Maschke, B., Lefèvre, L.: Weak form of Stokes-Dirac structures and geometric discretization of port-Hamiltonian systems. *J. Comput. Phys.* **361**, 442–476 (2018). <https://doi.org/10.1016/j.jcp.2018.02.006>
25. Lindell, I., Sihvola, A.: General boundary conditions. *Bound. Conditions Electromagn.* **78**(267), 101–141 (2019). <https://doi.org/10.1002/9781119632429.ch5>
26. Logg, A., Mardal, K.A., Wells, G.: Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book, vol. 84. Springer, ??? (2012). <https://doi.org/10.1007/978-3-642-23099-8>
27. McDonald, F., McLachlan, R.I., Moore, B.E., Quispel, G.R.: Travelling wave solutions of multisymplectic discretizations of semi-linear wave equations. *J. Differ. Equ. Appl.* **22**(7), 913–940 (2016). <https://doi.org/10.1080/10236198.2016.1162161>
28. McLachlan, R.: Symplectic integration of Hamiltonian wave equations. *Numer. Math.* **66**, 465–492 (1993). <https://doi.org/10.1007/BF01385708>
29. McLachlan, R.I., Stern, A.: Multisymplecticity of hybridizable discontinuous Galerkin methods. *Found. Comput. Math.* **20**(1), 35–69 (2020). <https://doi.org/10.1007/s10208-019-09415-1>
30. Nitsche, J.: Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. In: Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg, vol. 36, pp. 9–15 (1971). <https://doi.org/10.1007/BF02995904>
31. Ophir, J., Céspedes, I., Ponnekanti, H., Yazdi, Y., Li, X.: Elastography: a quantitative method for imaging the elasticity of biological tissues. *Ultrason. Imag.* **13**(2), 111–134 (1991). <https://doi.org/10.1177/016173469101300201>
32. Polyanin, A.D.: Handbook of Linear Partial Differential Equations for Engineers and Scientists, vol. 40 (2002). <https://doi.org/10.5860/choice.40-0964>
33. Raviart, P.A., Thomas, J.M.: A mixed finite element method for 2nd order elliptic problems. In: Mathematical Aspects of Finite Element Methods, pp. 292–315. Springer, Berlin (1977). <https://doi.org/10.1007/bfb0064470>

34. Reich, S.: Multi-symplectic Runge-Kutta collocation methods for Hamiltonian wave equations. *J. Comput. Phys.* **157**(2), 473–499 (2000). <https://doi.org/10.1006/jcph.1999.6372>
35. Sadjina, S., Kyllingstad, L.T., Skjong, S., Pedersen, E.: Energy conservation and power bonds in co-simulations: non-iterative adaptive step size control and error estimation. *Eng. Comput.* **33**(3), 607–620 (2017). <https://doi.org/10.1007/s00366-016-0492-8>
36. Sánchez, M.A., Ciuca, C., Nguyen, N.C., Peraire, J., Cockburn, B.: Symplectic Hamiltonian HDG methods for wave propagation phenomena. *J. Comput. Phys.* **350**, 951–973 (2017). <https://doi.org/10.1016/j.jcp.2017.09.010>
37. Scovazzi, G., Carnes, B.: Weak boundary conditions for wave propagation problems in confined domains: formulation and implementation using a variational multiscale method. *Comput. Methods Appl. Mech. Eng.* **221**, 117–131 (2012). <https://doi.org/10.1016/j.cma.2012.01.018>
38. Serhani, A., Matignon, D., Haine, G.: Partitioned finite element method for port-Hamiltonian systems with boundary damping: anisotropic heterogeneous 2D wave equations. *IFAC-PapersOnLine* **52**, 96–101 (2019). <https://doi.org/10.1016/j.ifacol.2019.08.017>
39. Süli, E., Mayers, D.F.: *An Introduction to Numerical Analysis* (2003)
40. Trenchant, V., Fares, Y., Ramirez, H., Gorrec, Y.L., Ouisse, M.: A port-Hamiltonian formulation of a 2D boundary controlled acoustic system. *IFAC-PapersOnLine* **28**(13), 235–240 (2015). <https://doi.org/10.1016/j.ifacol.2015.10.245>
41. Trenchant, V., Ramirez, H., Le Gorrec, Y., Kotyczka, P.: Structure preserving spatial discretization of 2D hyperbolic systems using staggered grids finite difference. In: *Proceedings of the American Control Conference*, pp. 2491–2496 (2017). <https://doi.org/10.23919/ACC.2017.7963327>
42. Van Der Schaft, A., Jeltsema, D.: In: *Port-Hamiltonian Systems Theory: An Introductory Overview*, pp. 2–3 (2014). <https://doi.org/10.1561/2600000002>
43. Van Der Schaft, A.J., Maschke, B.M.: Hamiltonian formulation of distributed-parameter systems with boundary energy flow. *J. Geom. Phys.* **42**(1–2), 166–194 (2002). [https://doi.org/10.1016/S0393-0440\(01\)00083-3](https://doi.org/10.1016/S0393-0440(01)00083-3)
44. Vu, N.M.T., Lefèvre, L., Nouailletas, R., Brémond, S.: Symplectic spatial integration schemes for systems of balance equations. *J. Process Control* **51**, 1–17 (2017). <https://doi.org/10.1016/j.jprocont.2016.12.005>
45. Zhai, Z., Chen, Q.: Solution characters of iterative coupling between energy simulation and CFD programs. *Energy Build.* **35**(5), 493–505 (2003). [https://doi.org/10.1016/S0378-7788\(02\)00156-1](https://doi.org/10.1016/S0378-7788(02)00156-1)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.