eng

MDPI

Article

A Computationally Time-Efficient Method for Implementing Pressure Load to FE Models with Lagrangian Elements

Adnan Shahriar [1,*], Arsalan Majlesi [2] and Arturo Montoya [2]

[1] Department of Mechanical Engineering, The University of Texas at San Antonio, San Antonio, TX 78249, USA
[2] Department of Civil Engineering, The University of Texas at San Antonio, San Antonio, TX 78249, USA; arsalan.majlesi@my.utsa.edu (A.M.); arturo.montoya@utsa.edu (A.M.)
* Correspondence: adnan.shahriar@my.utsa.edu; Tel.: +1-210-8035-230

Abstract: A computationally time-efficient method is introduced to implement pressure load to a Finite element model. Hexahedron elements of the Lagrangian family with Gauss–Lobatto nodes and integration quadrature are utilized, where the integration points follow the same sequence as the nodes. This method calculates the equivalent nodal force due to pressure load using a single Hadamard multiplication. The arithmetic operations of this method are determined, which affirms its computational efficiency. Finally, the method is tested with finite element implementation and observed to increase the runtime ratio compared to the conventional method by over 20 times. This method can benefit the implementation of finite element models in fields where computational time is crucial, such as real-time and cyber–physical testbed implementation.

Keywords: equivalent pressure load; Lagrangian; Gauss–Lobatto nodes; computational time efficient; real-time FEM

Citation: Shahriar, A.; Majlesi, A.; Montoya, A. A Computationally Time-Efficient Method for Implementing Pressure Load to FE Models with Lagrangian Elements. *Eng* **2024**, *5*, 2379–2394. https://doi.org/10.3390/eng5030124

Academic Editor: Antonio Gil Bravo

Received: 15 July 2024
Revised: 16 September 2024
Accepted: 19 September 2024
Published: 22 September 2024

1. Introduction

Surface traction and body forces are the two main forms of external loadings that cause solids to deform. Body forces act on the inner, dispersed mass of the solid, whereas surface tractions act by applying normal and shearing stresses to the surface of the solid [1–3]. In the context of the finite element method (FEM), these two loading conditions are used to solve for the displacement of solids [3]. Other properties, such as stress and strain, can be computed from the displacement to analyze and design components of solid objects, i.e., structures. If multiple solid bodies are involved, contact models are crucial for enforcing continuity and compatibility conditions between the interacting surfaces [3–5]. There are several contact modeling techniques, such as surface-to-surface or node-to-surface models, where surface tractions are applied to model interfaces for coupling. However, the surface tractions cannot be applied to a FEM directly; instead, an equivalent nodal force vector is computed and applied. The computation of the equivalent nodal force consists of several steps: determination of (1) traction force, (2) equivalent nodal force for each of the integration points on the surface, and (3) assembly. This procedure involves several matrix–matrix multiplications. The total FE simulation consists of two phases: (1) precomputation and (2) runtime. For a dynamic case, during the runtime phase, if there is varying pressure loading or the model is in contact with another model, force vector assembly, followed by the computation of an equivalent nodal force vector for surface traction, needs to be carried out at each time step. Such computation requires substantial computational effort.

In fields where real-time computation is involved, computational time becomes very crucial. Such fields include cyber–physical testing, which involves real-time communication between an experimental and computational model [6]. Also, in a system of systems (SoS) [7,8] framework, there is a growing need to incorporate multiple solid or structural bodies to achieve accurate and holistic modeling in real time [9]. Recently, such testing has

been implemented to design resilient space habitats [10,11] that require high-fidelity two- and three-dimensional dynamic FE models. These models can be subjected to pressure loading, and the interaction between the cyber and physical components (cyber–physical testing [12]) can also employ such loading that needs to be computed in real time [13]. Hence, implementing a computational time-efficient method to calculate equivalent nodal force due to pressure load during runtime is necessary.

Efforts have been made to reduce computational time, specifically for dynamic applications. For example, diagonal mass matrices [13] have been implemented in the field of spectral element methods [14–17]. Because it is efficient to invert a mass matrix, this approach can expedite the explicit dynamic simulation. Here, the diagonalization of the mass matrices has resulted from the hexahedral elements of the Lagrangian family with Gauss–Lobatto nodes [18] and integration quadrature, which is different from the Gauss–Legendre quadrature [19]. However, for FE application, equivalent nodal force vector calculation from pressure load is computationally demanding, and minimizing the computational time associated with it has not been approached.

In this paper, a method is developed to reduce the computational cost associated with the implementation of the pressure load to a three-dimensional FEM during the runtime phase of a dynamic simulation. According to this method, the computational overhead for the total procedure to compute equivalent nodal force can be split into two sub-procedures, where the majority of the computation is performed during the precomputation phase. For this purpose, the hexahedral elements of the Lagrangian family with Gauss–Lobatto nodes and integration quadrature are implemented for surface integration. For such an integration quadrature, the integration points overlap with the nodal coordinates. As a result, the value of a shape function evaluated at the integration points is one at one integration point and zero at all others. This property is then utilized to reduce the equivalent nodal force vector computations to a single Hadamard multiplication [20] for the runtime phase. Such multiplication is also known as elementwise multiplication and is implemented in many languages, such as PYTHON (3.12.6) and MATLAB®(2023b). This method will come with a limitation as it cannot be implemented for large deformation when the surface area or surface normal vector changes.

The remainder of this paper is organized as follows. The methodology section first describes the implementation of the Lagrangian elements of the hexahedral family and integration quadrature. Then, the properties of such elements are utilized to develop a computationally time-efficient algorithm that utilizes a single Hadamard multiplication to compute the equivalent nodal force vector from the pressure load. Next, arithmetic operations are determined for the new algorithm and compared against the existing one. Then, the algorithms are implemented into FE models, and the CPU time is computed, which shows the efficiency of the current method compared to the conventional method. Next, a convergence study is conducted, and a dynamic analysis is carried out using an FE model of a space habitat subjected to a dynamic pressure load. Then, the limitations of the current method are discussed in detail. Finally, the conclusion section concludes this current work with some remarks.

2. Materials and Methods

This section first discusses the computation of the equivalent nodal force vector due to surface pressure. Next, the element that will be incorporated into this method is illustrated. Finally, the procedure to implement Hadamard multiplication to compute equivalent nodal force based on the framework discussed priorly is presented.

2.1. Conventional Method to Compute the Equivalent Nodal Force from Pressure Load

For an element, the equation of motion of the model in the coordinate $\boldsymbol{x}$ (x, y, z) is as follows:

$$[m]\{\ddot{u}\}^T + [c]\{\dot{u}\}^T + [k]\{u\}^T = \{r_e\}^T \quad (1)$$

where $[m]$, $[c]$, $[k]$ are the mass, damping, and stiffness matrix; $\{u\}$, $\{\dot{u}\}$, and $\{\ddot{u}\}$ are the displacement, velocity, and acceleration, respectively, for an element. On the right-hand side of Equation (1), $\{r_e\}$ is the equivalent nodal force due to external forces on element e. The matrices $[m]$, $[c]$, $[k]$, and $\{r_e\}$ are computed following the isoparametric formulation that can be found in the literature [4]. The vector of shape functions for an element in the isoparametric coordinate $\boldsymbol{\xi}(\xi, \eta, \zeta)$ is expressed as follows:

$$\{\chi\} = \begin{Bmatrix} N_1 & N_2 & \dots & N_{n_e} \end{Bmatrix} \tag{2}$$

where n_e is the number of nodes of an element and N_i is the value of i shape function at the isoparametric coordinate $\boldsymbol{\xi}$. Following Equation (3), a shape function matrix is defined as

$$[N] = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & \dots & N_{n_e} & 0 & 0 \\ 0 & N_1 & 0 & 0 & N_2 & 0 & \dots & 0 & N_{n_e} & 0 \\ 0 & 0 & N_1 & 0 & 0 & N_2 & \dots & 0 & 0 & N_{n_e} \end{bmatrix} \tag{3}$$

Different types of forces, such as surface traction, $\{f_{trac}\}$ and body force, $\{f_{body}\}$, contribute to the equivalent nodal force vector. Considering the normal surface traction $\{\varphi\}$, the $\{f_{trac}\}$ can be determined following isoparametric formulation as follows:

$$\{f_{trac}\}^T = \int_{-1}^{1}\int_{-1}^{1} [N]_{\cdot}^T P\{d\Gamma\}^T d\eta d\xi = \sum_{i=1}^{i=n_g} [N]_i^T P_i \{d\Gamma\}_i^T w_i \tag{4}$$

Here, the normal surface traction $\{\varphi\} = P\{\hat{n}\}$, where P is the traction magnitude and $\{\hat{n}\}$ is the normal unit vector on the surface of an element pointing outward from the element. P_i is the normal traction magnitude pointing outward from the element surface, w_i is the weight at integration point i, and n_g is the number of integration points of the element on the surface of the element subjected to the traction load. Note: pressure is the negative of the P_i. $\{d\Gamma\}$ is a vector in the global coordinate perpendicular to the surface the pressure is acting on, with $\|d\Gamma\|$ being the ratio of a differential area of the element between global and local coordinates. There are six surfaces for hexahedral elements. For demonstration purposes in this paper, considering $[J]$ as the Jacobian matrix, only the case of the $\zeta = 1$ plane of a hexahedral element is presented, as follows:

$$\{d\Gamma\}^T = \begin{Bmatrix} J_{22}J_{33} - J_{23}J_{32} \\ J_{23}J_{31} - J_{21}J_{33} \\ J_{21}J_{32} - J_{22}J_{31} \end{Bmatrix} \tag{5}$$

Evaluation of the traction value at the integration point i can be simplified by assigning nodal traction value. If the tractions of all nodes of the element e are $\{P\}_e$, following the isoparametric formulation

$$P_i = \{\chi\}_i \{p\}_e^T \tag{6}$$

The $\{f_{trac}\}$ is the equivalent nodal force vector for a single surface of an element. A single element can have multiple surfaces, i.e., hexahedral elements have six surfaces. Hence, $\{f_{trac}\}$ needs to be computed for all the surfaces of an element. This paper considers just one surface of each element subjected to pressure loading. Hence, the total number of surfaces of the whole model equals the number of elements. Considering n_E as the number of elements of the model, the computation of the global force vector can be determined by summing through elements. The equation of motion (EOM) for the assembled system is as follows:

$$[M]\left\{\ddot{U}\right\}^T + [C]\left\{\ddot{U}\right\}^T + [K]\{U\}^T = \{F_{trac}\}^T + \left\{F_{body}\right\}^T \tag{7}$$

where $\left\{\ddot{U}\right\}$, $\left\{\ddot{U}\right\}$, and $\{U\}$ are the acceleration, velocity, and displacement vectors, $[M]$, $[C]$, and $[K]$ are the mass, damping, and stiffness matrices, and $\left\{F_{body}\right\}$ is the body force vector

for the assembled model. Several time-stepping algorithms exist to solve the EOM. For this study, the implicit Newmark-Beta method [4] is implemented, which is an implicit time-stepping method for solving the displacement at discrete time $t + \Delta t$ as follows:

$$\left[\overline{K}\right]\left\{{}^{t+\Delta t}_{\cdot}U\right\}^T = \left\{{}^{t}_{\cdot}\overline{R}\right\}^T \tag{8}$$

where $\left[\overline{K}\right]$ is the equivalent stiffness matrix of the form

$$\left[\overline{K}\right] = [K] + a_0[M] + a_1[C] \tag{9}$$

And $\left\{{}^{t}_{\cdot}\overline{R}\right\}$ is the equivalent nodal force vector

$$\left\{{}^{t}_{\cdot}\overline{R}\right\}^T = [M]\left(a_0\left\{{}^{t}_{\cdot}U\right\}^T + a_2\left\{{}^{t}_{\cdot}\dot{U}\right\}^T + a_3\left\{{}^{t}_{\cdot}\ddot{U}\right\}^T\right) + [C]\left(a_1\left\{{}^{t}_{\cdot}U\right\}^T + a_4\left\{{}^{t}_{\cdot}\dot{U}\right\}^T + a_5\left\{{}^{t}_{\cdot}\ddot{U}\right\}^T\right) + \{F_{trac}\}^T + \left\{F_{body}\right\}^T \tag{10}$$

Upon obtaining the displacement vector, the acceleration and velocity can be obtained, respectively, as follows:

$$\left\{{}^{t+\Delta t}_{\cdot}\ddot{U}\right\} = a_0\left(\left\{{}^{t+\Delta t}_{\cdot}U\right\} - \left\{{}^{t}_{\cdot}U\right\}\right) - a_2\left\{{}^{t}_{\cdot}\dot{U}\right\} - a_3\left\{{}^{t}_{\cdot}\ddot{U}\right\} \tag{11}$$

$$\left\{{}^{t+\Delta t}_{\cdot}\dot{U}\right\} = \left\{{}^{t}_{\cdot}\dot{U}\right\} + a_7\left\{{}^{t+\Delta t}_{\cdot}\ddot{U}\right\} + a_6\left\{{}^{t}_{\cdot}\ddot{U}\right\} \tag{12}$$

where the definition of $a_0 - a_7$ in Equations (8)–(12) can be found in [4].

2.2. Lagrangian Element with Legendre-Gauss-Lobatto Nodes and Integration Quadrature

The hexahedron elements used for FE analysis can be divided into two families: (1) serendipity, i.e., a 20-node brick quadratic widely used in commercial software such as Abaqus 6.9 [21] and (2) Lagrangian, i.e., a 27-node brick quadratic [22]. The Lagrangian element consists of Legendre–Gauss–Lobatto (LGL) nodes and integration quadrature. For a one-dimensional n-point LGL quadrature between -1 and 1, along the ξ axis, the nodal coordinates are the solution of

$$\left(1 - \xi^2\right)\frac{d}{d\xi}L_{n-1} = 0 \tag{13}$$

where L_{n-1} indicates the Legendre polynomial of the degree $n - 1$. The weight corresponding to a node i, with coordinate ξ_i, can be calculated as follows:

$$w_i = \frac{2}{n(n-1)[L_n(\xi_i)]^2} \tag{14}$$

This formulation can be extended to 3D elements with a different order along the ξ, η, ζ axis, as n_ξ, n_η, and n_ζ, respectively. A schematic of such 3D elements with $n_\xi = 6$, $n_\eta = 6$, and $n_\zeta = 3$ utilized by [23] to simulate wave propagation is presented in Figure 1. One of the shape functions is plotted, and the 0 value is presented as void. The integration points of the element overlap with the nodal coordinates; hence, the shape function is 1 at one integration point and 0 at all others.

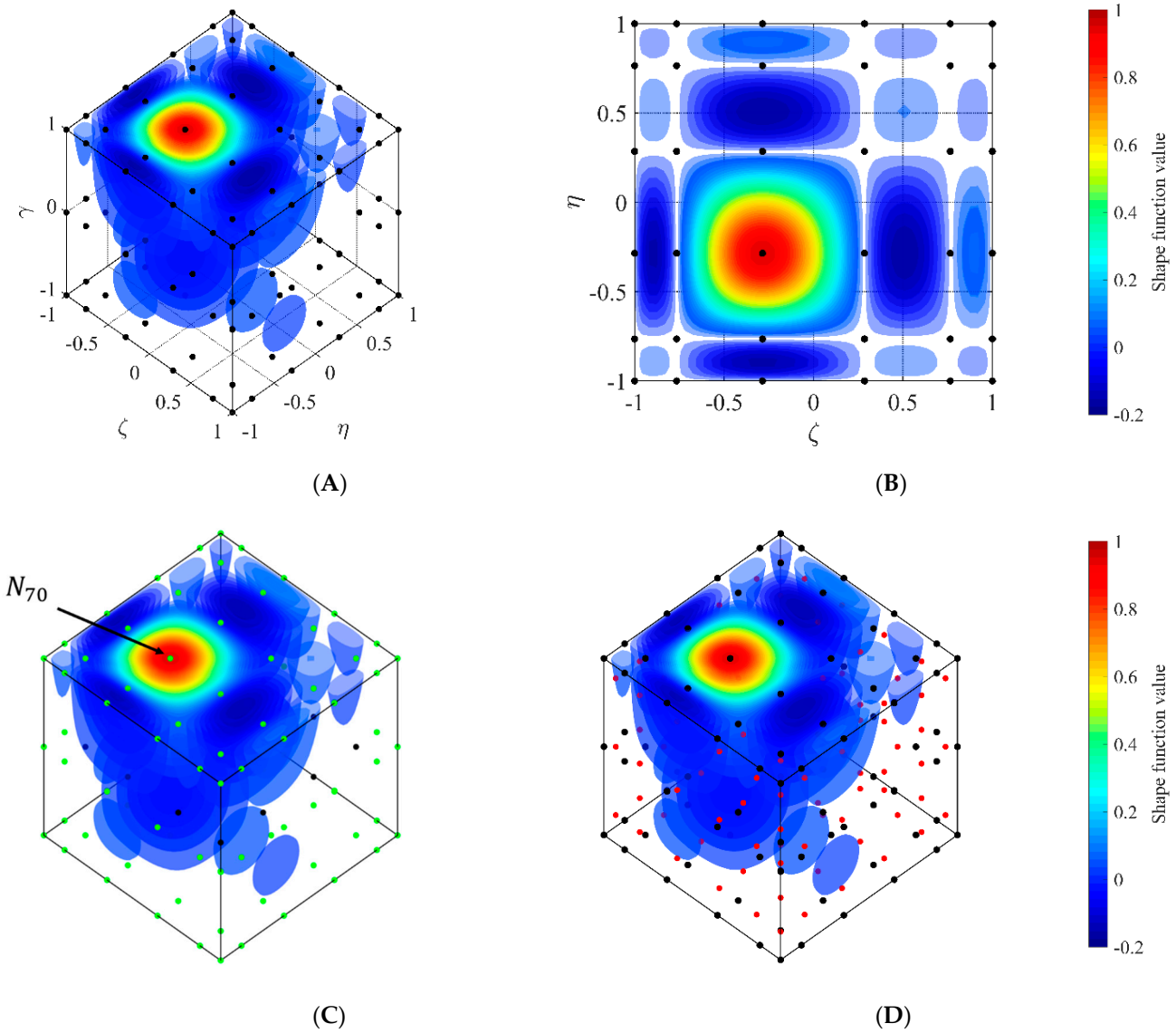


Figure 1. Schematic of a Lagrangian element with Legendre–Gauss–Lobatto nodes and integration quadrature. $n_\xi = 6$, $n_\eta = 6$, and $n_\zeta = 3$, (**A**) isometric, (**B**) top view, (**C**) Legendre–Gauss–Lobatto integration quadrature on the surface (green), (**D**) Gauss–Legendre integration quadrature for the region (red).

2.3. Development of a Computationally Time-Efficient Algorithm to Compute Equivalent Nodal Force from Pressure Load

The procedure to implement Hadamard multiplication to compute equivalent nodal force is presented in this section. First, the implementation of the Lagrangian element is discussed. In this paper, the Legendre–Gauss–Lobatto integration quadrature for a Lagrangian element is computed such that the sequence of node numbers is the same as the sequence of the integration points. A schematic of such an element is presented in Figure 1C, with all integration points on the surface (green) shown to coincide with the nodes on the surface. Also, for a shape function N_i ($i = 70$ in the figure), the i indicates both the node number and the integration point number of the element. Note: the Gauss–Legendre integration quadrature can be used for region integration, as shown in Figure 1D. As the node numbers and integration points follow the same coordinate and same sequence, for an integration point number i, Equations (2) and (3) are as follows:

$$\{\chi\}_i = \{0 \quad \dots \quad N_i = 1 \quad \dots \quad 0\} \tag{15}$$

$$[N]_i = \begin{bmatrix} 0 & 0 & 0 & \dots & N_i = 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & N_i = 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & N_i = 1 & \dots & 0 & 0 & 0 \end{bmatrix} \tag{16}$$

Next, the algorithm that reduces the equivalent nodal force calculation into a single Hadamard multiplication utilizing Equations (15) and (16) is developed. Following Equation (15), Equation (6) is reduced into

$$P_i = p_i \tag{17}$$

Similarly, following Equation (16) and replacing P_i with p_i, the components of $\{f_{trac}\}$ are as follows:

$$\begin{Bmatrix} f_{trac_{3i-2}} \\ f_{trac_{3i-1}} \\ f_{trac_{3i}} \end{Bmatrix} = [N]_i^T w_i \{d\Gamma\}_i^T p_i \tag{18}$$

Equation (18) can be written in terms of Hadamard multiplication as follows:

$$\begin{Bmatrix} f_{trac_{3i-2}} \\ f_{trac_{3i-1}} \\ f_{trac_{3i}} \end{Bmatrix} = \left\{ [N]_i^T w_i \{d\Gamma\}_i^T \right\} \circ \begin{Bmatrix} p_i \\ p_i \\ p_i \end{Bmatrix}^T \tag{19}$$

Due to the linear independence of the components of $\{f_{trac}\}$ from $3i-2$ to $3i$, the force vector for an element can be written as follows:

$$\{f_{trac}\}^T = \left(\sum_{i=1}^{i=n_g} [N]_i^T w_i \{d\Gamma\}_i^T \right) \circ \{f_p\}^T \tag{20}$$

During assembly, if node i of element e_1 and node j of the element e_2 have the same coordinate and are denoted as node k in the global coordinate, the k components of the force vector of the global coordinate are as follows:

$$\begin{Bmatrix} F_{trac_{3k-2}} \\ F_{trac_{3k-1}} \\ F_{trac_{3k}} \end{Bmatrix} = \begin{Bmatrix} f_{trac_{3i-2}} \\ f_{trac_{3i-1}} \\ f_{trac_{3i}} \end{Bmatrix}_{e_1} + \begin{Bmatrix} f_{trac_{3j-2}} \\ f_{trac_{3j-1}} \\ f_{trac_{3j}} \end{Bmatrix}_{e_2} \tag{21}$$

Following Equation (19), Equation (21) can be written as

$$\begin{Bmatrix} F_{trac_{3k-2}} \\ F_{trac_{3k-1}} \\ F_{trac_{3k}} \end{Bmatrix} = \left\{ \left\{ [N]_i^T w_i \{d\Gamma\}_i^T \right\}_{e_1} + \left\{ [N]_j^T w_j \{d\Gamma\}_j^T \right\}_{e_2} \right\} \circ \begin{Bmatrix} p_k \\ p_k \\ p_k \end{Bmatrix}^T \tag{22}$$

Hence, the total force vector is as follows:

$$\{F_{trac}\}^T = \left(\sum_{j=1}^{j=e} \left(\sum_{i=1}^{i=n_g} [N]_i^T w_i \{d\Gamma\}_i^T \right) \right) \circ \{F_p\}^T \tag{23}$$

where

$$\{F_p\}^T = \{ \{p_1 \quad p_1 \quad p_1\} \quad \{p_2 \quad p_2 \quad p_2\} \quad \dots \quad \{p_{n_n} \quad p_{n_n} \quad p_{n_n}\} \} \tag{24}$$

where n_n is the total number of nodes of the assembled model. The summation term of Equation (24) is defined as $\{F_{P\_unity}\}$, which follows

$$\{F_{P\_unity}\}^T = \left(\sum_{j=1}^{j=e} \left(\sum_{i=1}^{i=n_g} [N]_i^T w_i \{d\Gamma\}_i^T \right) \right) \tag{25}$$

Although the Gauss–Legendre quadrature is widely used in finite element applications and requires two fewer integration points than Gauss–Lobatto to integrate a polynomial exactly, Equation (17) cannot be implemented as the nodal coordinates do not coincide with the integration points. Hence, Equation (6) needs to be used, restricting the development towards Equation (23).

If large deformation is not involved, $\{F_{P\_unity}\}$ is constant for a model; hence, it can be precomputed before the simulation, and only the Hadamard multiplication needs to be carried out between 2 vectors according to Equation (25). The procedure and the number of arithmetic operations are determined for the conventional and current methods in Tables 1 and 2, respectively. In Table 2, the operations are computed for the precomputed and runtime phases.

Table 1. Arithmetic operations of the conventional method.

Operation	Operation Description	No of Arithmetic Operations
Extract $\{p\}_e^{\cdot}$ from $\{p\}$ of Equation (6)	1: Elementwise addition $\underset{n_e \times 1}{\{p\}_e^{\cdot}} = \underset{n_e \times 1}{\{p\}_e^{\cdot}} + \underset{n_n \times 1}{\{p\}}$	n_e
Compute P_i, Equation (6)	2: Matrix vector multiplication $\underset{1 \times 1}{P_i} = \underset{1 \times n_e}{\{\chi\}_i} \times \underset{n_e \times 1}{\{p\}_e^T}$	$2n_e$
Compute $\{f_{trac}\}$, Equation (4)	3.1: Matrix–matrix multiplication $\underset{3n_e \times 1}{\{a\}} = \underset{3n_e \times 3}{[N]_i^T} \times \underset{3 \times 1}{\{d\Gamma\}_i^T}$	$12n_e$
	3.2: Scalar–scalar multiplication $\underset{1 \times 1}{b} = \underset{1 \times 1}{P_i} \times \underset{1 \times 1}{w_i}$	1
	3.3: Scalar–vector multiplication $\underset{3n_e \times 1}{\{c\}} = \underset{3n_e \times 1}{\{a\}} \times \underset{1 \times 1}{b}$	$3n_e$
	3.4: Sum to $\{f_{trac}\}$ $\underset{3n_e \times 1}{\{f_{trac}\}} = \underset{3n_e \times 1}{\{f_{trac}\}} + \underset{3n_e \times 1}{\{c\}}$	$3n_e$
Assembly of the force vector for one surface	4.1: Operation 1 once and 2 to 3, n_g times	$(20n_e + 1)n_g + n_e$
	4.2: Elementwise addition $\underset{3n_n \times 1}{\{F_{trac}\}} = \underset{3n_n \times 1}{\{F_{trac}\}} + \underset{3n_e \times 1}{\{f_{trac}\}}$	$3n_n$
Total for the whole model	5: Operation 4, n_E times	$\big((20n_e + 1)n_g + n_e + 3n_n\big)n_E$

The algorithm flowchart for both methods is presented in Figure 2, with the precomputation and runtime phases marked as blue and yellow, respectively. The pressure load in an input to the FE models is used for the equivalent nodal force computation. It is evident from Figure 2B that the precomputation phase handles most of the operations, leaving only $6nn$ operations during the runtime phase for such computation. Note that the conventional method involves $\big((20n_e + 1)n_g + n_g + 3n_n\big)n_E \gg 6n_n$ operations that have to be carried out during runtime (Figure 2A).

Table 2. Arithmetic operations of the current method.

Operation	Operation Description	No of Arithmetic Operations for Precomputation	No of Arithmetic Operation Runtimes
Compute $\left\{F_{P\_unity}\right\}$, Equation (25)	1.1: Matrix–matrix multiplication $\underset{3n_e \times 1}{\{a\}} = \underset{3n_e \times 3}{[N]_i^T} \times \underset{3 \times 1}{\{d\Gamma\}_i^T}$	$12n_e$	
	1.2: Scalar–vector multiplication $\underset{3n_e \times 1}{\{c\}} = \underset{3n_e \times 1}{\{a\}} \times \underset{1 \times 1}{w_i}$	$3n_e$	
	1.3: Sum to $\{d\}$ $\underset{3n_e \times 1}{\{d\}} = \underset{3n_e \times 1}{\{d\}} + \underset{3n_e \times 1}{\{c\}}$	$3n_e$	
	1.4 : Operations 1.1 to 1.3, n_g times	$18n_e \times n_g$	
	1.5: Elementwise addition $\underset{3n_n \times 1}{\left\{F_{P\_unity}\right\}} = \underset{3n_n \times 1}{\left\{F_{P\_unity}\right\}} + \underset{3n_e \times 1}{\{d\}}$	$3n_n$	
	1.6 : Operations 1.4 and 1.5, n_E times	$(18n_e \times n_g + 3n_n)n_E$	
Compute $\{F_p\}$, Equation (24)	2: Elementwise addition $\underset{3n_n \times 1}{\{F_p\}} = \underset{3n_n \times 1}{\{F_P\}} + \underset{n_n \times 1}{\{p\}}$		$3n_n$
Compute $\{F_{trac}\}$, Equation (23)	3: Hammard multiplication $\underset{3n_n \times 1}{\{F_{trac}\}} = \underset{3n_n \times 1}{\left\{F_{P\_unity}\right\}} \circ \underset{3n_n \times 1}{\{F_p\}}$		$3n_n$
Total for the whole model		$(18n_e \times n_g + 3n_n)n_E$	$6n_n$

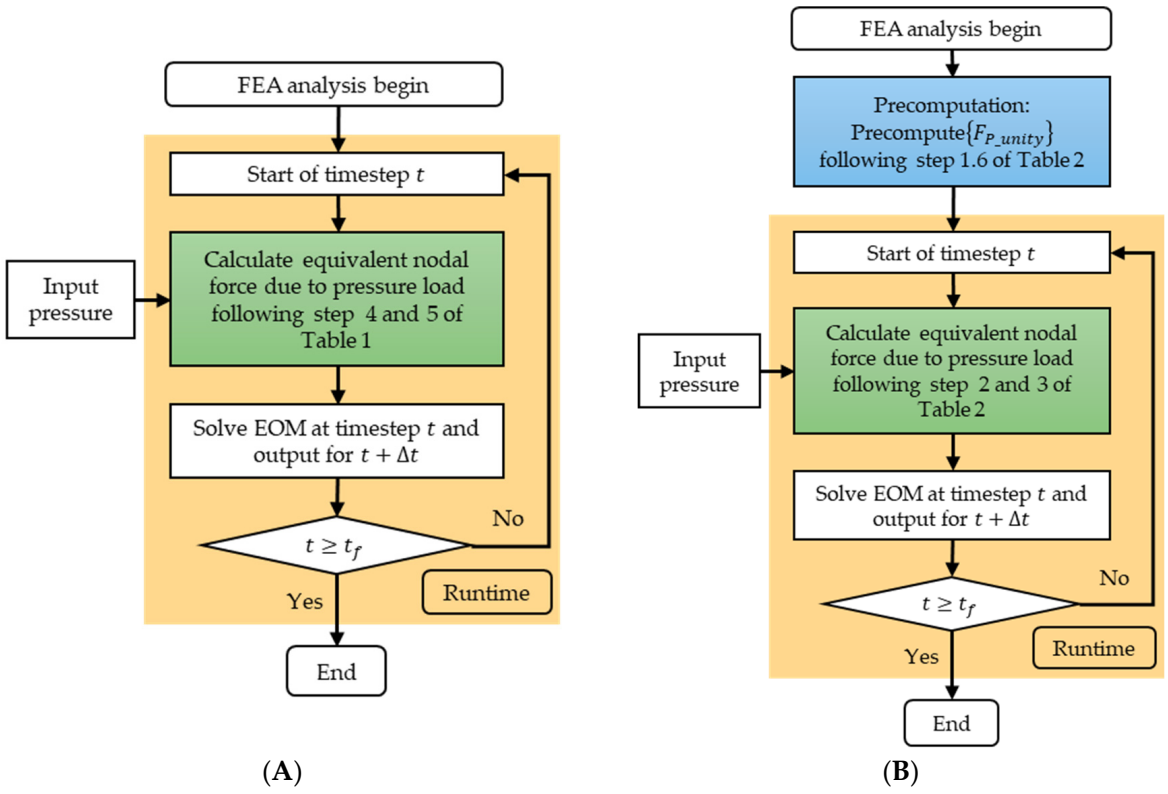
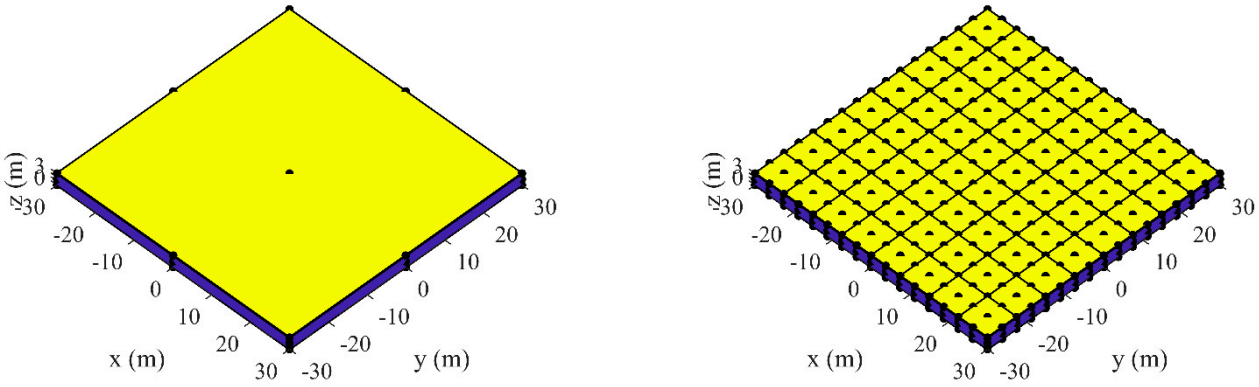


Figure 2. Flowchart for (**A**) conventional method using Table 1 and (**B**) current method using Table 2.

Additionally, as the $\{F_{P\_unity}\}$ is a one-dimensional vector of size $3n \times 1$, the memory requirement for storing the vector is the same as the force vector for the whole model computed during the assembly operation. Hence, this current method does not increase the memory overhead with respect to the conventional method.

2.4. Application: Model Details

First, the efficiency in terms of computational time is tested with a plate model. Although, for 3D implementation, a single layer of elements is not recommended, the objective is to prove computational efficiency. Hence, the algorithm is tested to a simple model: a one-element layer thick plate of dimension 30 m × 30 m × 3 m, where pressure load is applied on the top surface (yellow surface of Figure 3). The Gauss–Legendre quadrature is used for region integration, whereas the Gauss–Lobatto is used for equivalent nodal force calculation from the pressure load.



(**A**) (**B**)

Figure 3. Plate model with (**A**) 1 element and (**B**) 64 elements with 27-node elements.

As only one of the surfaces for each element is involved, the total number of surfaces is equal to the number of elements. The computational time with respect to the conventional method is measured as the runtime ratio, which is the ratio of CPU time following the current and conventional method during runtime.

Two cases are carried out. First, the number of elements increases from 1 to 121 as 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, and 121, by changing the number of elements along x and y and keeping just 1 element along z. The equivalent nodal forces are calculated for a pressure load of 1 *Pa*.

Next, for the case of 36 elements, the nodes of a single element are increased from 2 to 9 as n_η and n_ξ and n_ζ is kept as 3. A schematic of one of the cases, $n_\xi = n_\eta = 7$, is presented in Figure 4.

Next, a convergence study is carried out using the plate model. The static displacement with the material property described in Table 3 was utilized for convergence analysis. The edge of the plate on the $z = 0$ plane was considered fixed for the boundary condition, and a pressure load of 0.1 MPa was implemented. The finest mesh of all of these test cases is with 121 elements. Although increasing the number of elements is expected to reduce error, the convergence behavior can be shown with the 121-element mesh. Hence, the displacement of the fine mesh with 121 elements was computed and set as the reference.

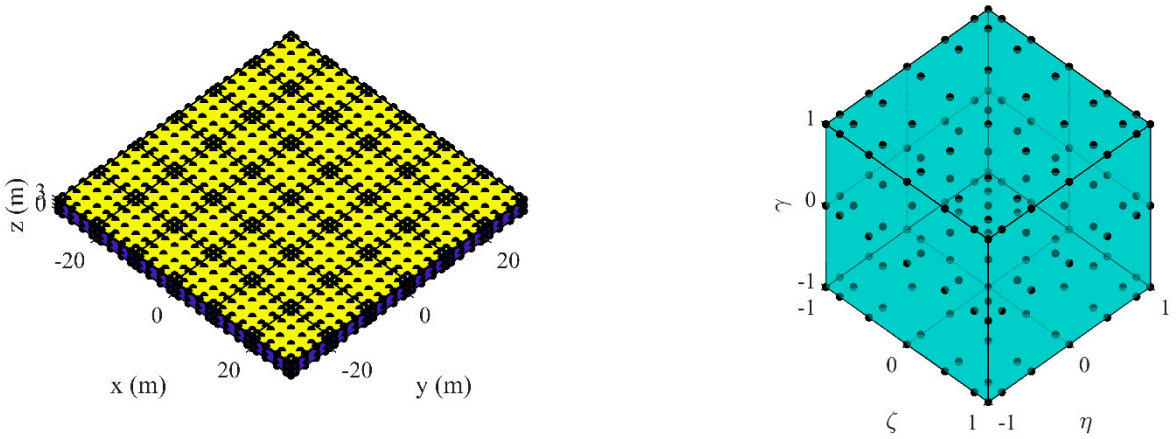


Figure 4. Plate model (**Left**) of 36 elements with the 147-node element (**Right**) $n_\xi = n_\eta = 7$ and $n_\zeta = 3$.

Table 3. Material properties.

Material Properties	Value
Modulus of elasticity	68 GPa
Poission's ratio	0.3
Density	2703 kg/m^3
Damping [C]	0.00001[K]

With u_{121} determined, the difference in displacement or error for other models Δu is computed as follows:

$$\Delta u = \frac{(u - u_{121})}{u_{121}} \times 100\% \tag{26}$$

Finally, this method is implemented on a space habitat model. A space habitat dome model with inner and outer radii of 2.5 and 2.9 m is modeled using 64 27-node quadratic elements, 819 nodes, and a 2457 DOF, as shown in Figure 5. The habitat has two surfaces: red indicates the inner, and blue indicates the outer surface, subjected to pressure load P_{inner} and P_{outer}, respectively. The yellow region from $z = 0$ to -1 is developed for ground consideration, and the habitat will be connected to the ground. For the analysis, the nodes at the yellow region with $z < 0$ are considered fixed-end boundary conditions.

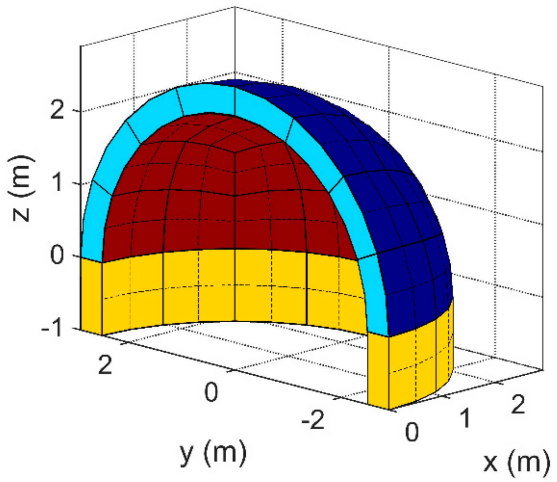


Figure 5. The space habitat FE model.

Dynamic analyses are carried out using the traction load following both current and conventional methods. The time-varying pressure load is set as follows:

$$P_{outer} = 10^4 y \times \sin(2\pi t) \tag{27}$$

$$P_{inner} = 101325 + 10^2 \sin(20\pi t) \tag{28}$$

For analysis, first, during the precomputation stage, a static analysis with $P_{inner} = 101,325$ and $P_{outer} = 0$ is performed. Next, the inverse of $[\overline{K}]$ is precomputed following Equation (9). Finally, the dynamic simulation is performed as the runtime stage, following the NB integration scheme (Equations (8)–(12)) with timestep $\Delta t = 0.0001$ s. Besides displacement, velocity, and acceleration, the stress is calculated for the integration points following the procedure described in [6]. Finally, the stress is extrapolated to the nodal points.

3. Results and Discussion

3.1. Computational Efficiency and Convergence Test

The runtime ratio is obtained following both conventional and current methods for the plate subjected to pressure load. The obtained equivalent nodal force for both procedures matches and verifies the procedure. For the study with the increase in element number, the results are presented in Figure 6A, where the current method is 5.3 times faster for one element. As the number of elements increases, the ratio reaches 27.5 asymptotically. The runtime ratio for the increase in element order is illustrated in Figure 6B, where it is evident that the current method is at least > 20 times faster than the conventional method, but the change in element order does not affect the computational performance predictably.

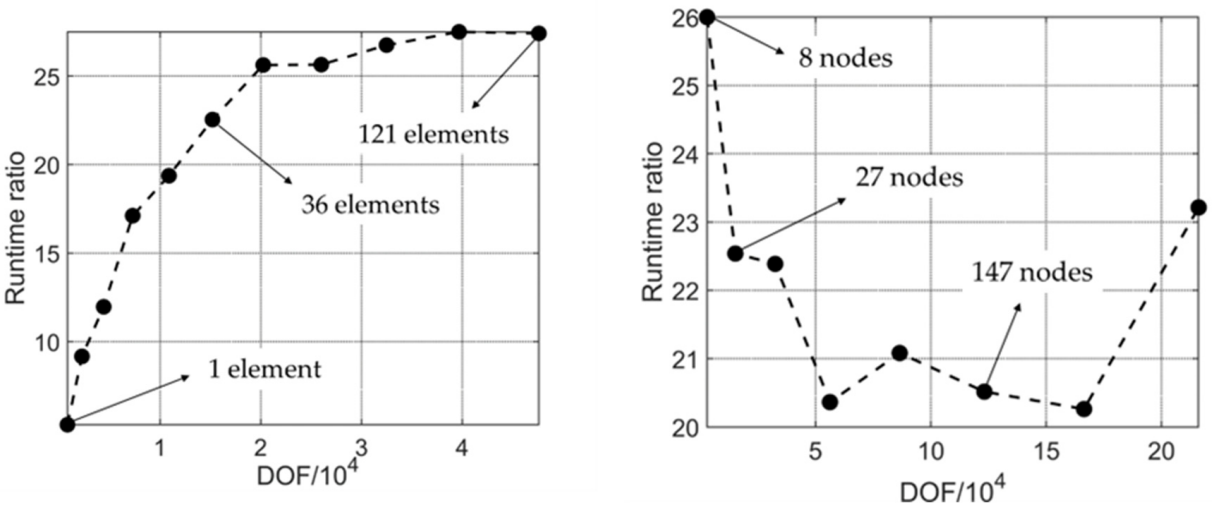


Figure 6. The runtime ratio for (**A**) an increase in element number with 27-node elements and (**B**) an increase in element order with 36 elements.

The displacement for the 121-element model is computed as $u_{121} = 16.12$ mm. A 3D overview of the displacement profile is shown in Figure 7. Equation (26) is used to obtain the error for the increase in the number of elements and the increase in the element order, presented in Figures 8A and 8B, respectively. It is obvious from the plot that the error decreases with an increase in both the number of elements and element order.

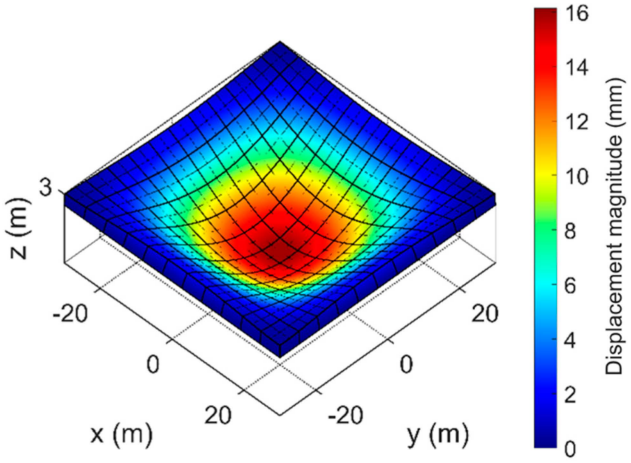


Figure 7. Plate model with 121 27-node brick elements.

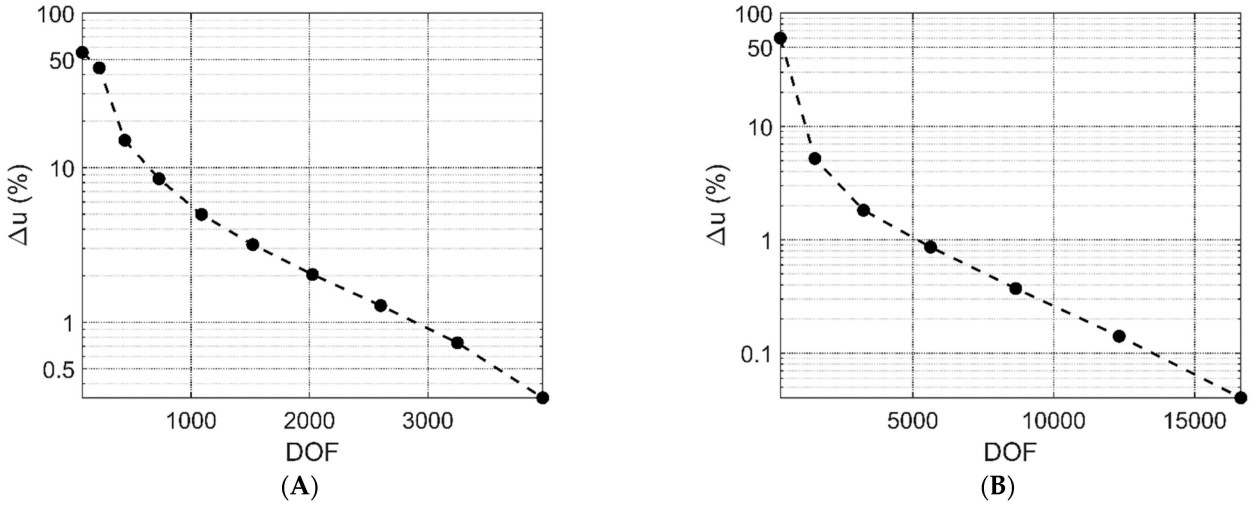
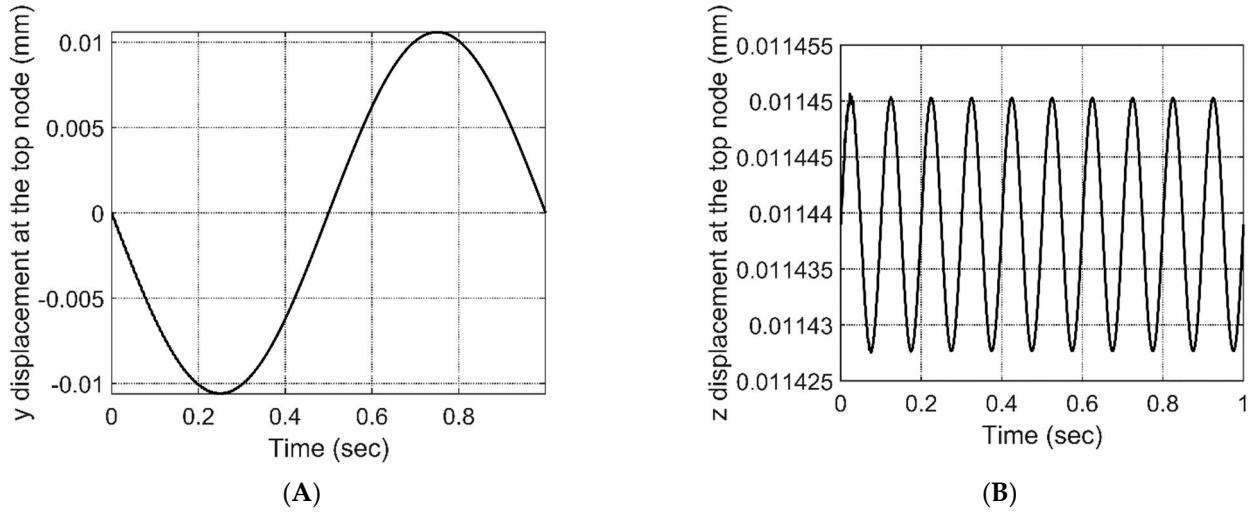


Figure 8. Convergence for (**A**) an increase in element number with 27-node elements and (**B**) an increase in element order with 36 elements.

3.2. Application to Space Habitat Models

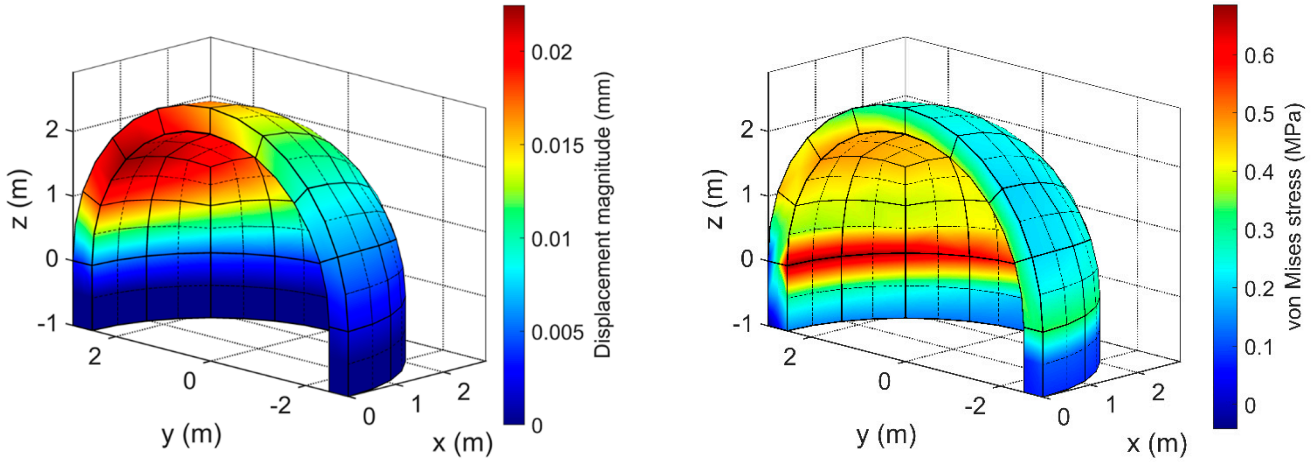
The displacement profile for the dynamic scheme for the topmost node at z = 2.9 is shown in Figure 9A along y and Figure 9B along the *z*-axis, which shows that the frequency of the sinusoidal behavior of the displacement matches the loading frequency. A 3D profile of the displacement and von Mises stress are presented in Figures 10A and 10B at 0.75 s, respectively.

The time required for the pressure calculation following the current and conventional method, as well as the NB integration with precomputed $[\overline{K}]$, is presented in Figure 11. It is obvious from the figure that the current method (around 6 s) takes significantly less time compared to the conventional method (around 175 s), which is higher than the NB integration scheme (around 55 s). The calculated runtime ratio is 26.25.



(**A**) (**B**)

Figure 9. Displacement at the top node at z = 2.9 m along (**A**) y- and (**B**) *z*-axis.



(**A**) (**B**)

Figure 10. (**A**) Displacement and (**B**) von Mises stress profile at 0.75 s.

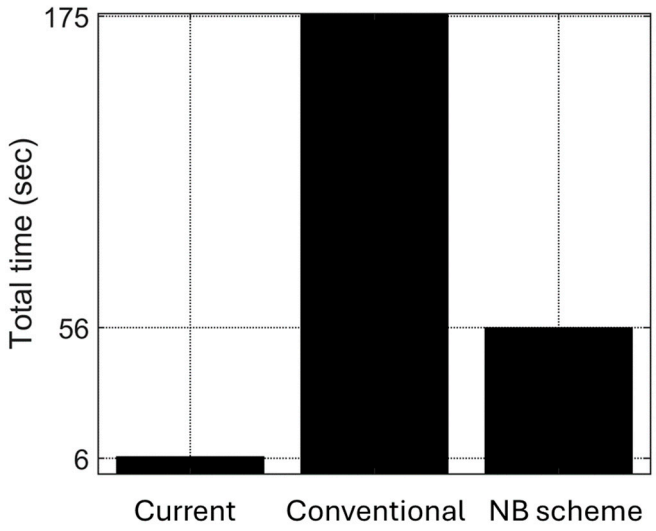


Figure 11. Total time taken by the current and conventional methods for equivalent nodal force calculation from pressure load and NB integration during the dynamic simulation.

3.3. Limitations

This work has three limitations: First, only the Lagrangian element can benefit from this method, as nodes of the serendipity family elements do not coincide with the LGL integration quadrature. Second, time-invariant $\{d\Gamma\}$ is required to precompute Equation (25). Hence, this method may not be applicable to large deformations [24]. Finally, it primarily considers elements with a single surface subjected to traction loading. If multiple surfaces of an element experience unequal pressure loads, the method presented cannot be directly applied. While the detailed implementation of such cases is beyond the scope of this paper, a procedure to address this scenario is outlined here. Figure 12 illustrates an example where elements e_1 and e_2 each have two faces subjected to traction loads. For element e_1, node i experiences equal pressure loading on both surfaces, denoted as a. Conversely, for the element e_2, node j is subjected to differing traction loads from two distinct surfaces. To handle this, the method involves partitioning all surfaces into two groups, Γ_1 and Γ_2, represented by red and blue, respectively. Subsequently, Equation (23) is applied independently for each group and calculates the $\{F_p\}$ vector. Finally, a Hadamard product is employed to compute the global force vector, as shown in Equation (29).

$$\{F_{trac}\}^T = \left[\left\{F_{P_{unity}}\right\}\Big|_{\Gamma_1} \quad \left\{F_{P_{unity}}\right\}\Big|_{\Gamma_2} \quad \dots\right]^T \circ \left[\{F_p\}|_{\Gamma_1} \quad \{F_p\}|_{\Gamma_2} \quad \dots\right]^T \tag{29}$$

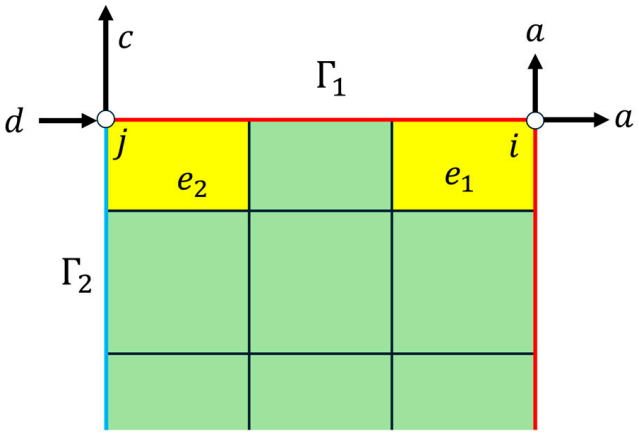


Figure 12. Surface partitioning to account for corner nodes subjected to two different traction loads from two different surfaces.

4. Conclusions

A computationally time-efficient method was developed to compute equivalent nodal force due to pressure load. Elements of the Lagrangian family with Gauss–Lobatto nodes and integration quadrature were implemented in such a way that the integration points follow the same sequence as the nodes. Through the implementation of such an element, the computation of the equivalent nodal force is reduced into a single Hadamard multiplication. Computational efficiency was established by computing the arithmetic complexity of this method. This method is implemented in a single-element thick plate model with different element densities. As the number of elements increases, the runtime ratio increases, surpassing 20 for 36 elements for a one-element thick plate model. It is also observed that the increase in element order decreases the runtime ratio by around 22% for the 36-element density but still outperforms the conventional method. Finally, a habitat model is developed, and dynamic analysis is carried out with a time-varying pressure load on two different surfaces that showed a runtime ratio over 20 by the current method to compute the equivalent nodal force from the pressure load.

Author Contributions: Conceptualization, A.S.; methodology, A.S.; software, A.S.; validation, A.S., and A.M. (Arsalan Majlesi); formal analysis, A.S.; investigation, A.S. and A.M. (Arsalan Majlesi); resources, A.S. and A.M. (Arturo Montoya); data curation, A.S. and A.M. (Arsalan Majlesi); writing—original draft preparation, A.S. and A.M. (Arsalan Majlesi); writing—review and editing, A.M. (Arturo Montoya); visualization, A.S.; supervision, A.M. (Arturo Montoya); project administration, A.M. (Arturo Montoya); funding acquisition, A.M. (Arturo Montoya). All authors have read and agreed to the published version of the manuscript.

Funding: This material was based on work carried out under the Resilient Extra-Terrestrial Habitat Institute (RETHi) supported by a Space Technology Research Institute grant (No. 80NSSC19K1076) from NASA's Space Technology Research Grants Program. This paper does not have any conflicts of interest.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

$[m]$	Mass matrix for an element
$[c]$	Damping matrix for an element
$[k]$	Stiffness matrix for an element
$\{\ddot{u}\}$	Acceleration vector for an element
$\{\dot{u}\}$	Velocity vector for an element
$\{u\}$	Displacement vector for an element
$\{r_e\}$	Equivalent nodal force due to external forces on element e
$\{f_{body}\}$	Body force vector of an element
$\{f_{trac}\}$	Tractor force vector of an element
$\{F_{trac}\}$	Traction vector of the assembled model
$\{F_{body}\}$	Body force vector of the assembled model
$[M]$	Mass matrix for the assembled model
$[K]$	Stiffness matrix for the assembled model
$[C]$	Damping matrix for the assembled model
$\{\ddot{U}\}$	Acceleration vector for the assembled model
$\{\dot{U}\}$	Velocity vector for the assembled model
$[\overline{K}]$	Equivalent stiffness matrix for the assembled model
$\{U\}$	Displacement vector for the assembled model
$\{\varphi\}$	Normal surface traction vector
N_i	Shape function corresponds to node i of an element
$[J]$	Jacobian matrix
P_i	Traction magnitude on an integration point i
w_i	Weight of an integration point i
$\{\chi\}$	vector of shape functions
n	Number of nodes
n_ξ	Number of nodes along ξ axis
n_η	Number of nodes along η axis
n_ζ	Number of nodes along ζ axis
p_i	Pressure at node i
n_e	Number of nodes of an element e
n_g	Number of integration points of an element e
n_E	Number of elements of the assembled model
n_n	Number of nodes of the assembled model

Abbreviations

FE	Finite element
FEM	Finite element method
LGL	Legendre–Gauss–Lobatto
NB	Newmark-Beta
EOM	Equation of motion
SoS	System of systems

References

1. Sadd, M.H. Elasticity: Theory, Applications, and Numerics. Academic Press: Cambridge, MA, USA, 2009.
2. Reismann, H. On the forced motion of elastic solids. *Appl. Sci. Res.* **1968**, *18*, 156–165. [CrossRef]
3. Zienkiewicz, O.; Taylor, R.; Zhu, J.Z. *The Finite Element Method: Its Basis and Fundamentals: Seventh Edition*; Butterworth-Heinemann: Oxford, UK, 2005.
4. Bathe, K.-J. *Finite Element Procedures*; Prentice Hall Education: Upper Saddle River, NJ, USA, 1996.
5. Papadopoulos, P.; Solberg, J.M. A Lagrange multiplier method for the finite element solution of frictionless contact problems. *Math. Comput. Model.* **1998**, *28*, 373–384. [CrossRef]
6. Perić, D.; Owen, D.R.J. Computational model for 3-D contact problems with friction based on the penalty method. *Int. J. Numer. Methods Eng.* **1992**, *35*, 1289–1309. [CrossRef]
7. Fagcang, H.; Stobart, R.; Steffen, T. A review of component-in-the-loop: Cyber-physical experiments for rapid system development and integration. *Adv. Mech. Eng.* **2022**, *14*, 16878132221109969. [CrossRef]
8. Delp, C.; Cooney, L.; Dutenhoffer, C.; Gostelow, R.; Jackson, M.; Wilkerson, M.; Kahn, T.; Piggott, S. The Challenge of Model-based Systems Engineering for Space Systems, Year 2. *Insight* **2009**, *12*, 36–39. [CrossRef]
9. Szarazi, J.; Reichwein, A.; Bock, C. Integrating Finite Element Analysis with Systems Engineering Models. In Proceedings of the NAFEMS World Congress, Stockholm, Sweden, 11 June 2017.
10. Dyke, S.J.; Marais, K.; Bilionis, I.; Werfel, J.; Malla, R. Strategies for the design and operation of resilient extraterrestrial habitats. In Proceedings of the Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems, Online, 22–26 March 2021; p. 1159105. [CrossRef]
11. Shahriar, A.; Montoya, H.; Majlesi, A.; Avila, D.; Montoya, A. Coupling Independent Solid Mechanics-Based Systems in a System-of-Systems Modeling Framework. *AIAA J.* **2024**, *62*, 1–16. [CrossRef]
12. Maghareh, A.; Lenjani, A.; Krishnan, M.; Dyke, S.; Bilionis, I. Role of cyber-physical testing in developing resilient extraterrestrial habitats. *Earth Space* **2021**, 1059–1068.
13. Teukolsky, S.A. Short note on the mass matrix for Gauss–Lobatto grid points. *J. Comput. Phys.* **2015**, *283*, 408–413. [CrossRef]
14. Palacz, M.; Krawczuk, M.; Żak, A. Spectral Element Methods for Damage Detection and Condition Monitoring. In *Smart Innovation, Systems and Technologies*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 549–558. [CrossRef]
15. Ostachowicz, W.; Kudela, P.; Krawczuk, M.; Zak, A. *Guided Waves in Structures for SHM: The Time-Domain Spectral Element Method*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2012.
16. Patera, A.T. A spectral element method for fluid dynamics: Laminar flow in a channel expansion. *J. Comput. Phys.* **1984**, *54*, 468–488. [CrossRef]
17. Komatitsch, D.; Tromp, J. Spectral-element simulations of global seismic wave propagation—I. Validation. *Geophys. J. Int.* **2002**, *149*, 390–412. [CrossRef]
18. Gautschi, W. High-order Gauss–Lobatto formulae. *Numer. Algorithms* **2000**, *25*, 213–222. [CrossRef]
19. Swarztrauber, P.N. On computing the points and weights for Gauss--Legendre quadrature. *SIAM J. Sci. Comput.* **2003**, *24*, 945–954. [CrossRef]
20. Million, E. The hadamard product. *Course Notes* **2007**, *3*, 1–7.
21. Smith, M. *ABAQUS/Standard User's Manual, Version 6.9*; Dassault Systèmes Simulia Corp.: Johnston, RI, USA, 2009.
22. Maździarz, M. Unified isoparametric 3D lagrangeFinite elements. *CMES Comput. Model. Eng. Sci.* **2010**, *66*, 1–24.
23. Soman, R.; Kudela, P.; Balasubramaniam, K.; Singh, S.K.; Malinowski, P. A study of sensor placement optimization problem for guided wave-based damage detection. *Sensors* **2019**, *19*, 1856. [CrossRef] [PubMed]
24. Ouyang, P.-F.; Li, D.M.; Xie, J.-X. Modeling nonlinear deformation of slender auxetic structures under follower loads with complex variable meshfree methods. *Mech. Adv. Mater. Struct.* **2024**, *31*, 4969–4983. [CrossRef]