Finite-element Time Discretizations for the Unsteady Euler Equations

Nathan L. Mundis * Dimitri J. Mavriplis †

Department of Mechanical Engineering, University of Wyoming, Laramie, Wyoming 82071-3295

This work examines finite-element time discretizations for the Euler equations and methods for the robust and efficient solution of these discretizations. Specifically, the spectral-element (SEMT) and discontinuous-Galerkin (DGMT) methods in time are derived and examined in detail. To solve the SEMT and DGMT Euler equations, the flexible variant of the Generalized Minimal Residual method (FGMRES), utilizing the full second-order accurate spatial Jacobian and complete temporal coupling of the chosen time discretization, is implemented. The FGMRES solver developed utilizes a block-colored Gauss-Seidel (BCGS) preconditioner augmented by a defect-correction process to increase its effectiveness. This preliminary examination of the SEMT and DGMT shows that these time discretization have promise for the solution of the unsteady Euler equations, but also that the behavior of the methods around discontinuities must be improved to make them broadly applicable to unsteady flow problems. The investigation conducted in this work indicates that SEMT, rather than DGMT, is likely the better time discretization for use in solution of the Euler equations.

I. Introduction

The spectral-element and discontinuous-Galerkin methods in time discretize the time dimension into a number of time elements. Within each of these elements, the variation in time is described with polynomial functions. Although the SEMT method is referred to as a spectral-element method within the literature,[1,2] it can also be viewed as a continuous-Galerkin finite-element method; in other words, this paper compares continuous and discontinuous finite-element time discretizations. The spectral-element terminology arises because so-called spectral-element methods exhibit spectral-like error convergence, i.e. as the polynomial degree increases within the elements, the magnitude of the error decreases at an increasing rate. This exponential error convergence is a hallmark of spectral methods, such as the time-spectral method.[3] Since continuous-Galerkin methods exhibit similar error convergence, they are referred to as spectral-element methods. It should be noted, however, that discontinuous-Galerkin methods also exhibit exponential error convergence with increasing polynomial degree.

The SEMT and DGMT can be used to describe both hyperbolic and periodic problems in time, so they are applicable to all of the same problems as standard time-implicit methods. Unsteady aerodynamic problems of particular interest include analysis of flutter and self-excited oscillation (limit-cycles).[4,5]

The idea of using space-time finite elements to discretize the time domain was first proposed by Oden,[6] Desai et al.,[7] Fried,[8] and Argyris and Scharpf.[9] These finite elements can be coupled in one of three ways depending on the type of problem to be solved: time-marching backward-only coupling, monolithic-time coupling, and periodic coupling. The first two of these can be applied to all types of problems, while the last is only appropriate for purely periodic problems, just like the time-spectral method.[3]

In backward-only coupling, each element is coupled only to its immediately previous element with subsequent elements being solved sequentially. In the monolithic-time approach, the entire time domain for which a solution needs to be found is solved at once. Elements are coupled both to the immediately previous element and to the next element, meaning that all elements must be solved simultaneously. Kurdi and Beran[1] believe the monolithic time approach is better, for SEMT, because "the global-time projection of

*Post-doctoral Research Associate, AIAA Member; email: nmundis@uwyo.edu.

†Professor, AIAA Associate Fellow; email: mavripl@uwyo.edu.

the response, transforms the time-dependent differential equation into an algebraic form, which in contrast to time-marching, presents a strong connection between the unsteady (transient) solution and the system parameters." Apparently, these properties are particularly beneficial in the optimal control of limit-cycle oscillations (LCOs) in various aircraft[10,11] because global-time allows for the sensitivity of the LCO amplitude to be computed.[1]

Time-marching finite elements can employ Hamilton's law of varying action,[12] weighted residual methods,[13] discontinuous-Galerkin methods,[14] and spectral-element methods.[1] Space-time time-marching finite-elements include formulations based on least squares[15,16,17,18] and discontinuous-Galerkin methods.[19,20,21,22,23,24] These time-marching schemes use the same backward-only coupling as the results presented herein. It should also be noted that for linear problems, backward-only coupled SEMT and DGMT are identical to some fully-implicit Runge-Kutta (FIRK) methods, specifically SEMT is equivalent to the Lobatto IIIA FIRK method while DGMT on Gauss-Lobatto points is equivalent to Lobatto IIIC. Thus, the results and implementation of finite-element methods in time has implications for the analogous FIRK methods, even when applied to non-linear problems.

Conversely, finite-difference time-marching schemes are much more commonplace and include linear multi-step methods (BDF1 and BDF2), multi-stage Runge-Kutta methods,[25] and the implicit Newmark scheme.[26,27] Much attention is given to these schemes for the unsteady solution of the Euler and Navier-Stokes equations.[28,29,30,31]

Like time-spectral methods,[3] SEMT and DGMT can take advantage of additional parallelism by solving all points in time within a given element simultaneously. This parallelism is extended even further when the monolithic-time or periodic couplings are used, as more points in time can be solved at once. The periodic SEMT or DGMT might have an advantage over TS for problems with a broad range of harmonic content. In the TS discretization, higher harmonics must be captured by adding time-instances, i.e. through p-refinement alone. In contrast, SEMT might be able to capture these areas of large gradients by clustering finite elements in time around them, i.e. through h-refinement. This potential advantage of periodic SEMT and DGMT over TS is one main reason for this exploration of finite-element methods in time.

The following sections describe the SEMT and DGMT discretization of the Euler equations and how they are solved using the preconditioned flexible variant of the Generalized Minimal Residual method (FGMRES). Then, several solutions using the SEMT are explored, for both the Euler equations and several ordinary differential equations (ODEs). These solutions are discussed to ascertain when the SEMT works well and what its shortcomings are. Next, the DGMT is explored and is compared to the SEMT to ascertain the strengths and weaknesses of the two time-discretization schemes. Finally, conclusions about the SEMT and DGMT are discussed and possible future improvements to address their shortcomings are proposed.

II. Governing Equations and Solution Method

A. The Euler Equations

The Euler equations in conservative form can be written as follows:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0 \tag{1}$$

where $\mathbf{U}$ represents the vector of conserved quantities (mass, momentum, and energy) which in two dimensions is given as follows:

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}$$

and $\mathbf{F}(\mathbf{U})$ represents the convective fluxes, which have x and y components as follows:

$$\mathbf{F_x}(\mathbf{U}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho E + p)u \end{bmatrix}, \qquad \mathbf{F_y}(\mathbf{U}) = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\rho E + p)v \end{bmatrix}.$$

Integrating over a (moving) control volume $\Omega(t)$, the following can be obtained:

$$\int_{\Omega(t)} \frac{\partial \mathbf{U}}{\partial t} dV + \int_{\Omega(t)} (\nabla \cdot \mathbf{F}(\mathbf{U})) dV = 0. \tag{2}$$

Applying the divergence theorem to the second term in the equation above produces the following form for the Euler equations:

$$\int_{\Omega(t)} \frac{\partial \mathbf{U}}{\partial t} dV + \int_{\partial\Omega(t)} (\mathbf{F}(\mathbf{U}) \cdot \tilde{\mathbf{n}}) dS = 0. \tag{3}$$

Using the differential identity

$$\frac{\partial}{\partial t} \int_{\Omega(t)} \mathbf{U} dV = \int_{\Omega(t)} \frac{\partial \mathbf{U}}{\partial t} dV + \int_{\partial\Omega(t)} \mathbf{U}(\dot{\mathbf{x}} \cdot \tilde{\mathbf{n}}) dS \tag{4}$$

where $\dot{\mathbf{x}}$ and $\tilde{\mathbf{n}}$ are the velocity and normal of the interface $\partial\Omega(t)$, respectively, equation (3) becomes:

$$\frac{\partial}{\partial t} \int_{\Omega(t)} \mathbf{U} dV + \int_{\partial\Omega(t)} (\mathbf{F}(\mathbf{U}) - \mathbf{U}\dot{\mathbf{x}}) \cdot \tilde{\mathbf{n}} dS = 0. \tag{5}$$

Considering $\mathbf{U}$ as cell averaged quantities, these equations are discretized in space as follows:

$$\frac{\partial}{\partial t}(V\mathbf{U}) + \mathbf{R}(\mathbf{U}, \dot{\mathbf{x}}(t), \tilde{\mathbf{n}}(t)) = 0 \tag{6}$$

where $\mathbf{R}(\mathbf{U}, \dot{\mathbf{x}}, \tilde{\mathbf{n}}) = \int_{\partial\Omega(t)} (\mathbf{F}(\mathbf{U}) - \dot{\mathbf{x}}\mathbf{U}) \cdot \tilde{\mathbf{n}} dS$ is the spatial residual, representing the discrete convective fluxes in arbitrary-Lagrangian-Eulerian (ALE) form and V denotes the control volume. In the discrete form, $\dot{\mathbf{x}}(t)$ and $\tilde{\mathbf{n}}(t)$ now represent the time varying velocities and surface normals of the control-volume boundary faces.

The Euler equations are discretized by a central-difference finite-volume scheme with additional matrix-based artificial dissipation on hybrid meshes which may include triangles and quadrilaterals in two dimensions. Second-order accuracy (on smoothly varying meshes) is achieved using a two-pass construction of the artificial dissipation operator, which corresponds to an undivided biharmonic operator. A single unifying face-based data-structure is used in the flow solver for all types of elements. For a given face, the residual contribution of that face can be written as:

$$\mathbf{R}_{1stO,ik}(\mathbf{U}, \dot{\mathbf{x}}(t), \tilde{\mathbf{n}}(t)) = (\mathbf{F}_i(\mathbf{U}_i) + \mathbf{F}_k(\mathbf{U}_k) - \mathbf{U}_{ik}\dot{\mathbf{x}}_{ik}) \cdot \tilde{\mathbf{n}} \Delta S + \kappa \underline{\mathbf{T}} |\underline{\Lambda}| \underline{\mathbf{T}}^{-1} (\mathbf{U}_i - \mathbf{U}_k) \tag{7}$$

for first-order matrix dissipation, where $\underline{\mathbf{T}}$ is the left-eigenvector matrix, $\underline{\Lambda}$ is the eigenvalue matrix, and $\underline{\mathbf{T}}^{-1}$ is the right-eigenvector matrix of the convective fluxes. For second-order matrix dissipation, the residual on a face can be written as follows:

$$\mathbf{R}_{2ndO,ik}(\mathbf{U}, \dot{\mathbf{x}}(t), \tilde{\mathbf{n}}(t)) = (\mathbf{F}_i(\mathbf{U}_i) + \mathbf{F}_k(\mathbf{U}_k) - \mathbf{U}_{ik}\dot{\mathbf{x}}_{ik}) \cdot \tilde{\mathbf{n}} \Delta S + \kappa \underline{\mathbf{T}} |\underline{\Lambda}| \underline{\mathbf{T}}^{-1} (\mathbf{L}_i(\mathbf{U}) - \mathbf{L}_k(\mathbf{U})) \tag{8}$$

where $\mathbf{L}(\mathbf{U})$ is the undivided Laplacian operator, taken as:

$$\mathbf{L}_p(\mathbf{U}) = \sum_{q=1}^{neighbors} (\mathbf{U}_q - \mathbf{U}_p) \tag{9}$$

In both cases, κ is an empirical constant with a typical value of $1/2$ for first-order matrix dissipation and $1/8$ for second-order matrix dissipation.

B. Finite-element Methods in Time

Finite-element methods are applied in the time dimension to find a periodic, quasi-periodic, or transient solution to the Euler Equations. It should be noted that, although SEMT is referred to as a "spectral method" in the literature, because if offers spectral-like p-convergence, it is in fact a finite-element method using Gauss-Lobatto quadrature points and a Lagrange polynomial basis.[1] First, the continuous time,

trial solution for a single time element m is discretized using Nth degree Lagrange polynomials on $N+1$ quadrature points, as follows:

$$(V\mathbf{U})^{(m)}(\zeta) = \sum_{k=0}^{N} (V\mathbf{U})^{(m)}(\zeta_k)\psi_k^{(m)}(\zeta) \tag{10}$$

where $(V\mathbf{U})^{(m)}$ is the continuous time, trial function for element m, $\psi_k^{(m)}$ is the kth Lagrange polynomial of degree N in time element m given by the following:

$$\psi_k(\zeta) = \prod_{\eta=0,\ldots,N,\eta\neq k} \left(\frac{\zeta-\zeta_\eta}{\zeta_k-\zeta_\eta}\right). \tag{11}$$

ζ_k and ζ_η are the ζ locations of quadrature points k and η where $\zeta \in [-1,1]$. The transformation from t to ζ and the inverse transform are $t = t_{m-1} + \frac{1}{2}(\zeta+1)T^{(m)}$ and $\zeta = \frac{2}{T^{(m)}}(t-t_{m-1}) - 1$ where, in this case, t_{m-1} is the time at which element m begins and $T^{(m)} = t_m - t_{m-1}$ is the length of element m with t_m being its rightward endpoint.

The Lobatto polynomials are the derivatives of the Legendre polynomials: $L_{o_i}(\zeta) = L'_{i+1}(\zeta)$ where the Legendre polynomials are defined as follows:

$$L_i(\zeta) = \frac{1}{2^i i!}\frac{d^i(\zeta^2-1)^i}{d\zeta^i}.$$

The Gauss-Lobatto quadrature points occur at the zeros of the completed Lobatto polynomials: $L^c_{o_{N+1}}(\zeta) = (1-\zeta^2)L_{o_{N-1}}$

To discretize the Euler equations using SEMT or DGMT, one begins with the general form of the Euler equations as give by equation (6). Substituting the trial solution given by equation (10) into (6) and minimizing the residuals using the Bubnov-Galerkin method,[32,1] the following is obtained for time element m:

$$\int_{-1}^{1} \nu(\zeta)\left[\frac{d(V\mathbf{U})^{(m)}}{d\zeta} + \frac{T^{(m)}}{2}\mathbf{R}(\mathbf{U}^{(m)},\dot{\mathbf{x}}^{(m)},\tilde{\mathbf{n}}^{(m)})\right] d\zeta = 0 \tag{12}$$

where $\nu(\zeta)$ is a weighting function taken as the pth Lagrange polynomial of degree N, $\psi_p^{(m)}$. Integrating by parts for $p = 0,\ldots,N$, the following is obtained for each element m:

$$(V\mathbf{U})^{(m)}\psi_p^{(m)}|_{-1}^{1} - \int_{-1}^{1}\left((V\mathbf{U})^{(m)}\frac{d\psi_p^{(m)}}{d\zeta} - \frac{T^{(m)}}{2}\mathbf{R}(\mathbf{U}^{(m)},\dot{\mathbf{x}}^{(m)},\tilde{\mathbf{n}}^{(m)})\psi_p^{(m)}\right) d\zeta = 0 \tag{13}$$

Exploiting the properties of the Lagrange polynomials, equation (13) is efficiently integrated over the Gauss-Lobatto quadrature points, including the end points, as follows:

$$\int_{-1}^{1} Q\,d\zeta = \sum_{p=0}^{N} Q(\zeta_p)\omega_p \tag{14}$$

where Q is any generic function of ζ and ω_p is the Gauss-Lobatto quadrature weight at node p. In matrix form for each element m, this becomes as follows:

$$\underline{\mathbf{\Psi}}^{(m)}\begin{bmatrix} V\mathbf{U}(\zeta_0) \\ \vdots \\ V\mathbf{U}(\zeta_N)\end{bmatrix}^{(m)} = \underline{\mathbf{I}}_{\omega}^{(m)}\begin{bmatrix} \mathbf{R}(\mathbf{U}(\zeta_0),\dot{\mathbf{x}}(\zeta_0),\tilde{\mathbf{n}}(\zeta_0)) \\ \vdots \\ \mathbf{R}(\mathbf{U}(\zeta_N),\dot{\mathbf{x}}(\zeta_N),\tilde{\mathbf{n}}(\zeta_N))\end{bmatrix}^{(m)} \tag{15}$$

where

$$\underline{\mathbf{\Psi}}^{(m)} = \begin{bmatrix} \left(\frac{d\psi_0}{d\zeta}|_{\zeta_0}\omega_0+1\right) & \frac{d\psi_0}{d\zeta}|_{\zeta_1}\omega_1 & \ldots & \frac{d\psi_0}{d\zeta}|_{\zeta_{N-1}}\omega_{N-1} & \frac{d\psi_0}{d\zeta}|_{\zeta_N}\omega_N \\ \frac{d\psi_1}{d\zeta}|_{\zeta_0}\omega_0 & \frac{d\psi_1}{d\zeta}|_{\zeta_1}\omega_1 & \ldots & \frac{d\psi_1}{d\zeta}|_{\zeta_{N-1}}\omega_{N-1} & \frac{d\psi_1}{d\zeta}|_{\zeta_N}\omega_N \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{d\psi_{N-1}}{d\zeta}|_{\zeta_0}\omega_0 & \frac{d\psi_{N-1}}{d\zeta}|_{\zeta_1}\omega_1 & \ldots & \frac{d\psi_{N-1}}{d\zeta}|_{\zeta_{N-1}}\omega_{N-1} & \frac{d\psi_{N-1}}{d\zeta}|_{\zeta_N}\omega_N \\ \frac{d\psi_N}{d\zeta}|_{\zeta_0}\omega_0 & \frac{d\psi_N}{d\zeta}|_{\zeta_1}\omega_1 & \ldots & \frac{d\psi_N}{d\zeta}|_{\zeta_{N-1}}\omega_{N-1} & \left(\frac{d\psi_N}{d\zeta}|_{\zeta_N}\omega_N-1\right) \end{bmatrix}, \tag{16}$$

and

$$\underline{\mathbf{I}}_{\omega}^{(m)} = \frac{T^{(m)}}{2}\begin{bmatrix} \omega_0 & 0 & \dots & 0 & 0 \\ 0 & \omega_1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \omega_{N-1} & 0 \\ 0 & 0 & \dots & 0 & \omega_N \end{bmatrix}. \tag{17}$$

In equation (16), the first and last elements on the diagonal of $\underline{\mathbf{\Psi}}$ have additional ± 1 terms, corresponding to the temporal flux contributions from the element to itself, which manifest themselves as the first term in equation (13).

As was noted previously, the focus thus far has been on a single time element. These finite elements in time can now be coupled in one of three different ways depending on the computational resources available and the type of problem being solved. For the purposes of the present work, backward-only coupling has been used exclusively and so only backward-only coupling will be presented.

1. *Backward-only Coupling*

The simplest way in which the finite elements in time can be coupled is to couple each element only to its immediately previous element. For SEMT this amounts to treating each element as an initial value problem where the initial value for that element is the ending value of the previous element. For DGMT, backward-only coupling means that each element inherits a flux that is a function of the values in the immediately previous element only. For SEMT and DGMT on Gauss-Lobatto points, these formulations resemble first-order backward-difference formulae, where the values in each time element are only dependent on the final value from the previous element in time. Past elements have no dependence on future elements.

For SEMT and backward-only coupling, the first row of the $\underline{\mathbf{\Psi}}^{(m)}$ matrix given in equation (16) is replaced with zeroes, except the first matrix element of the row, which is replaced with one as follows:

$$\underline{\mathbf{\Psi}}^{(m)} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ \frac{d\psi_1}{d\zeta}|_{\zeta_0\omega_0} & \frac{d\psi_1}{d\zeta}|_{\zeta_1\omega_1} & \dots & \frac{d\psi_1}{d\zeta}|_{\zeta_{N-1}\omega_{N-1}} & \frac{d\psi_1}{d\zeta}|_{\zeta_N\omega_N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{d\psi_{N-1}}{d\zeta}|_{\zeta_0\omega_0} & \frac{d\psi_{N-1}}{d\zeta}|_{\zeta_1\omega_1} & \dots & \frac{d\psi_{N-1}}{d\zeta}|_{\zeta_{N-1}\omega_{N-1}} & \frac{d\psi_{N-1}}{d\zeta}|_{\zeta_N\omega_N} \\ \frac{d\psi_N}{d\zeta}|_{\zeta_0\omega_0} & \frac{d\psi_N}{d\zeta}|_{\zeta_1\omega_1} & \dots & \frac{d\psi_N}{d\zeta}|_{\zeta_{N-1}\omega_{N-1}} & \left(\frac{d\psi_N}{d\zeta}|_{\zeta_N\omega_N} - 1\right) \end{bmatrix}, \tag{18}$$

while the $\underline{\mathbf{I}}_{\omega}^{(m)}$ matrix remains unchanged. For each element, the value corresponding to $\zeta = -1$ is not computed, rather this value is supplied by the initial condition for the element as specified above. Thus, although the two matrices are of dimension $(N+1)\times(N+1)$, since the first row and column represent the coupling to the initial value, only N values, corresponding to all other values of ζ remain to be solved in each time element. This corresponds to one less than the number of quadrature points in the element and to the degree of the Lagrange polynomials used. Subsequent elements are then solved sequentially.

On the other hand, for DGMT utilizing backward-only coupling, the flux contribution to the first matrix element comes from the previous time element with no contribution to this flux coming from the the present element itself. As such, the $+1$ contribution to the first element is removed from the matrix given by equation (16) as follows:

$$\underline{\mathbf{\Psi}}^{(m)} = \begin{bmatrix} \frac{d\psi_0}{d\zeta}|_{\zeta_0\omega_0} & \frac{d\psi_0}{d\zeta}|_{\zeta_1\omega_1} & \dots & \frac{d\psi_0}{d\zeta}|_{\zeta_{N-1}\omega_{N-1}} & \frac{d\psi_0}{d\zeta}|_{\zeta_N\omega_N} \\ \frac{d\psi_1}{d\zeta}|_{\zeta_0\omega_0} & \frac{d\psi_1}{d\zeta}|_{\zeta_1\omega_1} & \dots & \frac{d\psi_1}{d\zeta}|_{\zeta_{N-1}\omega_{N-1}} & \frac{d\psi_1}{d\zeta}|_{\zeta_N\omega_N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{d\psi_{N-1}}{d\zeta}|_{\zeta_0\omega_0} & \frac{d\psi_{N-1}}{d\zeta}|_{\zeta_1\omega_1} & \dots & \frac{d\psi_{N-1}}{d\zeta}|_{\zeta_{N-1}\omega_{N-1}} & \frac{d\psi_{N-1}}{d\zeta}|_{\zeta_N\omega_N} \\ \frac{d\psi_N}{d\zeta}|_{\zeta_0\omega_0} & \frac{d\psi_N}{d\zeta}|_{\zeta_1\omega_1} & \dots & \frac{d\psi_N}{d\zeta}|_{\zeta_{N-1}\omega_{N-1}} & \left(\frac{d\psi_N}{d\zeta}|_{\zeta_N\omega_N} - 1\right) \end{bmatrix}. \tag{19}$$

Instead of having a flux contribution from itself, the corresponding flux contribution from the previous element is added. Since Gauss-Lobatto points are being used, this previous element flux contribution is only

dependent on the last point from the previous element and has the form given as follows:

$$F^{(m)}_{(m-1)} = -\left(V\mathbf{U}\right)^{N}_{(m-1)} \tag{20}$$

where $F^{(m)}_{(m-1)}$ is the elemental coupling flux from element $(m-1)$ to element (m), which is equal the negative of the volume-weighted solution values at end of the previous element. Ultimately, this coupling for DGMT means that DGMT retains all $N+1$ degrees of freedom per time element.

2. *Final Form of the SEMT and DGMT Euler Equations*

After the global time-coupling and quadrature-weight matrices are assembled, both terms in equation (15) are multiplied by the inverse of the global quadrature-weight matrix so that equation (15) has all non-identity matrices multiply the conserved variables state vector and the spatial residual vector remains unchanged. After this multiplication, the finite-element Euler equations have the following form:

$$\underline{\mathbf{E}}\overrightarrow{VU} + \overrightarrow{R} = 0 \tag{21}$$

where $\underline{\mathbf{E}} = -\underline{\mathbf{\Omega}}_G^{-1}\underline{\mathbf{\Psi}}_G$, $\underline{\mathbf{\Omega}}_G$ is the global weight matrix, and $\underline{\mathbf{\Psi}}_G$ is the global time-instance coupling matrix. When backward-only coupling is used, as herein, these matrices are equivalent to the element weight and coupling matrices, respectively, i.e. $\underline{\mathbf{\Omega}}_G = \underline{\mathbf{I}}^{(m)}_{\omega}$ and $\underline{\mathbf{\Psi}}_G = \underline{\mathbf{\Psi}}^{(m)}$, since each element is only coupled to the immediately previous element.

When backward-only coupling is used, this same procedure is used for each time element individually, since each time element is solved separately and sequentially coupled only to the immediately previous time element.

C. Fully Implicit Method

A common approach for solving the system of Euler equations consists of adding a pseudo-time term as:

$$\frac{\partial}{\partial\tau}(V\mathbf{U}) + \frac{\partial}{\partial t}(V\mathbf{U}) + \mathbf{R}\left(\mathbf{U}(t), \dot{\mathbf{x}}(t), \tilde{\mathbf{n}}(t)\right) = 0 \tag{22}$$

where $\frac{\partial}{\partial\tau}(V\mathbf{U})$ is the pseudo-time derivative term which is discretized using a first-order backward-difference formula (BDF1), and $\frac{\partial}{\partial t}(V\mathbf{U})$ is the physical-time derivative term which can be discretized using any time approach, for instance SEMT. The system of equations (22) is pseudo-time stepped until a pseudo-time steady state is achieved for the given physical-time discretization. For explicit pseudo-time stepping approaches, it has been shown that the pseudo-time step is limited by stability considerations. In the case of the second-order accurate time-implicit (BDF2) method, the following stability criterion applies:[33,34]

$$\Delta\tau_n = CFL\frac{V^n}{\parallel\lambda\parallel + \frac{V^n}{\Delta t}} \tag{23}$$

where λ is the spectral radius of the spatial discretization operator $\mathbf{R}(\mathbf{U}^n, \dot{\mathbf{x}}^n, \tilde{\mathbf{n}}^n)$ and Δt is the physical-time step size. The second term in the denominator limits the pseudo-time step size to being smaller than the physical-time step size, i.e. $\Delta\tau_n < \Delta t$. Generally, $\parallel\lambda\parallel \gg \frac{V^n}{\Delta t}$ so the physical-time step consideration can be neglected. In the case of finite-element time discretizations, the pseudo-time step size is also limited, similarly to the above limit, as follows:

$$\Delta\tau_{(m)} = CFL\frac{V^{(m)}}{\parallel\lambda\parallel + \frac{V^{(m)}N^2}{T^{(m)}}} \tag{24}$$

where N is the degree of the Lagrange polynomial basis used and $T^{(m)}$ is the physical-time length of the spectral element m. This restriction has not been formally derived, rather it is arrived upon by analogy to the idea that $\Delta\tau_{(m)} < \Delta t$, where Δt is taken as the shortest physical-time distance between two quadrature points within the element (m). This restriction is summarized for $N \in \{1,...,12\}$ in Table (1), below. As can be seen, for the values of N given in the table $\frac{1}{N^{3/2}}$ best approximates the smallest physical-time step; however, at $N = 12$, this approximation is already slightly less restrictive than what it should be. For this

reason, and to make the code extensible to even higher-orders in time, the $\frac{1}{N^2}$ restriction has been used. The use of this restriction accords with the authors' experience with the SEMT and DGMT: originally, a $\frac{1}{N}$ restriction was placed on the pseudo-time step as it was believed that the pseudo-time step should scale with the degree of the polynomial. At higher orders and when the physical-time element size was small, however, the $\frac{1}{N}$ restriction was found to be unstable, and the $\frac{1}{N^2}$ pseudo-time step restriction was introduced to restore stability.

Table 1. Finite-element pseudo-time step restriction with physical-time step

$\mathbf{N}$	$\mathbf{\Delta t_{min}}$	$\frac{\mathbf{1}}{\mathbf{N}}$	$\frac{\mathbf{1}}{\mathbf{N^{3/2}}}$	$\frac{\mathbf{1}}{\mathbf{N^2}}$
1	1.0000	1.0000	1.0000	1.0000
2	0.5000	0.5000	0.3536	0.0250
3	0.2764	0.3333	0.1925	0.1111
4	0.1727	0.2500	0.1250	0.0625
5	0.1175	0.2000	0.0894	0.0400
6	0.0849	0.1667	0.0680	0.0278
7	0.0641	0.1429	0.0540	0.0204
8	0.0501	0.1250	0.0442	0.0156
9	0.0402	0.1111	0.0370	0.0123
10	0.0330	0.1000	0.0316	0.0100
11	0.0276	0.0909	0.0274	0.0083
12	0.0233	0.0833	0.0241	0.0069

These restrictions on $\Delta\tau$ result in convergence degradation as the length of the time element decreases or as N increases. The impact of these pseudo-time step size restrictions can be greatly lessened by resorting to an implicit approach in pseudo-time. Such an approach has been derived in reference[35] using a first-order backwards-difference scheme in pseudo-time.

A more general strategy consists of devising a Newton approach for solving the fully-coupled non-linear equations at all points within a time element given by equation (22) with the chosen temporal discretization. The Newton scheme takes the form:

$$\underline{\mathbf{A}}\Delta\mathbf{U} = -\mathbf{R}_T\left(V\mathbf{U}^n, \dot{\mathbf{x}}^n, \tilde{\mathbf{n}}^n, t\right) \tag{25}$$

where $\mathbf{R}_T$ is the total residual including the contribution of the selected time discretization and the resulting Jacobian matrix is given as $\underline{\mathbf{A}} = \left[\frac{\partial R_T}{\partial \mathbf{U}}\right]$. For example, the Jacobian is given by the following for BDF2:

$$\underline{\mathbf{A}} = \frac{V^n}{\Delta\tau}\underline{\mathbf{I}} + \frac{3V^n}{2\Delta t}\underline{\mathbf{I}} + \underline{\mathbf{J}} \tag{26}$$

where the first term is a BDF1 pseudo-time term included to enhance diagonal dominance, the second term is the BDF2 physical-time term, $\underline{\mathbf{I}}$ is the identity matrix of the appropriate size given the spatial discretization, and $\underline{\mathbf{J}}$ is the Jacobian of the spatial discretization alone: $\underline{\mathbf{J}} = \left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right]$.

The Jacobian for SEMT and DGMT with either backward-only coupling or monolithic-time coupling is written as follows in terms of the $E_{j,k}$ which are the values in the jth row and kth column of the $\underline{\mathbf{E}}$ matrix that multiplies the conserved variable vectors in equation (21). As noted previously, the initial value or temporal-flux term can be incorporated as a source term, which is added as a constant to the right hand side (residual) of the equation and thus not included in the Jacobian matrix. For simplicity, the rows and columns of this matrix number $j,k \in [1,M]$ where $M = N$ for SEMT backward-only coupling and $M = N+1$

for DGMT backward-only coupling:

$$\underline{\mathbf{A}} = \begin{bmatrix} D_1 + \underline{\mathbf{J}}_1 & V^2 E_{1,2}\underline{\mathbf{I}} & \dots & V^{M-1}E_{1,M-1}\underline{\mathbf{I}} & V^M E_{1,M}\underline{\mathbf{I}} \\ V^1 E_{2,1}\underline{\mathbf{I}} & D_2 + \underline{\mathbf{J}}_2 & \dots & V^{M-1}E_{2,M-1}\underline{\mathbf{I}} & V^M E_{2,M}\underline{\mathbf{I}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ V^1 E_{M-1,1}\underline{\mathbf{I}} & V^2 E_{M-1,2}\underline{\mathbf{I}} & \dots & D_{M-1} + \underline{\mathbf{J}}_{M-1} & V^M E_{M-1,M}\underline{\mathbf{I}} \\ V^1 E_{M,1}\underline{\mathbf{I}} & V^2 E_{M,2}\underline{\mathbf{I}} & \dots & V^{M-1}E_{M,M-1}\underline{\mathbf{I}} & D_M + \underline{\mathbf{J}}_M \end{bmatrix} \tag{27}$$

where it was necessary to make the following substitution, $D_j = \left(\frac{1}{\Delta\tau_j} + E_{j,j}\right) V^j \underline{\mathbf{I}}$, to allow the Jacobian to fit in the required paper space, and where a diagonal pseudo-time term can be included as shown for enhanced diagonal dominance of the Jacobian matrix. In the above matrix, $\underline{\mathbf{J}}_j$ corresponds to the Jacobian of the spatial discretization operator evaluated at time instance j.

Returning to equation (25), each non-linear Newton iteration requires solving the linear system given by the following:

$$\underline{\mathbf{A}}\Delta\mathbf{U} = -\mathbf{R}_T\left(V\mathbf{U}^n, \dot{\mathbf{x}}^n, \tilde{\mathbf{n}}^n, t\right) \tag{28}$$

where $\mathbf{R}_T$ represents the residual of the complete system, including the time-discretization terms, (i.e. right hand side of equation (25)), and $\underline{\mathbf{A}}$ is a large matrix spanning all spatial and temporal degrees of freedom. Since direct inversion of $\underline{\mathbf{A}}$ is generally intractable, an inexact Newton scheme can be formulated using an approximate representation of $\underline{\mathbf{A}}$ which is simpler to invert. One possible simplification is to replace the exact spatial Jacobian in each diagonal block $\underline{\mathbf{J}}_i$ by the corresponding first-order Jacobian $\underline{\mathbf{J}}_{i,1stO}$ as is often done for steady-state solvers. Using the full $\underline{\mathbf{A}}$ matrix with $\underline{\mathbf{J}}_{i,1stO}$ the system may be solved in a block Jacobi fashion following:

$$\left[\frac{V^i}{\Delta\tau_i}\underline{\mathbf{I}} + \underline{\mathbf{J}}_{i,1stO}\right]\Delta\mathbf{U}_i^{l+1} = -\mathbf{R}_\mathbf{T}(\mathbf{U}) - \sum_{j\neq i}\left[V^j d_i^j \underline{\mathbf{I}}\right]\Delta\mathbf{U}_j^l \tag{29}$$

where the block size corresponds to the entire spatial domain for each time instance and where l denotes the block Jacobi iteration index.

D. Block-colored Gauss-Seidel Linear Solver

As can be seen, this approach requires the inversion the first-order spatial Jacobian (augmented with a pseudo-time term) at each iteration. This may be accomplished using a suitable iterative solver such as block-colored Gauss-Seidel (BCGS). In this case, the block now corresponds to the 4×4 block diagonal matrix at each cell of the spatial discretization, and the iterative scheme can be written as:

$$\left[\frac{V^i}{\Delta\tau_i}\underline{\mathbf{I}} + \left[\underline{\mathbf{D}}_{i,1stO}\right]\right]\Delta\mathbf{U}_i^{l+1} = -\mathbf{R}_\mathbf{T}(\mathbf{U}) - \sum_{j\neq i}\left[V^j d_i^j \underline{\mathbf{I}}\right]\Delta\mathbf{U}_j^l - \left[\underline{\mathbf{O}}_{i,1stO}\right]\Delta\mathbf{U}_i^l \tag{30}$$

where $\underline{\mathbf{D}}_{i,1stO}$ denotes the 4×4 block matrix for the current cell, $\underline{\mathbf{O}}_{i,1stO}$ refers to the off-diagonal blocks for neighboring mesh cells, and $\Delta\mathbf{U}_i^l$ are the spatial off-diagonal elements of the ith time instance. Although this equation describes a block Jacobi iteration (at the mesh-cell level), a block-colored Gauss-Seidel scheme can be recovered with a few simple modifications. First, the computational elements are divided into computational "colors" such that no two adjacent elements share the same color. This coloring allows the Gauss-Seidel method to be run in parallel. The BCGS method then updates (in parallel) all spatial elements of each individual color (with the different colors updated in sequence), such that each update uses the newest information available for all other colors. The BCGS algorithm, written in residual form for a generic linear system is given in Algorithm (1).

In Algorithm (1), ζ is the maximum number of BCGS iterations allowed, n_{colors} is the number of colors into which the elements have been divided, and n_{e_j} is the number of elements having the jth color. Additionally, $\underline{\mathbf{D}}_k$ is the diagonal block of the Jacobian for element k, and is inverted directly using LU-decomposition. The specification $\mathbf{x}_{current}$ is used to indicate that, some colors will have information from the previous iteration $\mathbf{x}_i$ while other colors might have already been updated during the current iteration $\mathbf{x}_{i+1}$; in other words, whether from the previous or current iteration, the most up-to-date information at the time of the evaluation of line 5 is used.

Algorithm 1 : Block-colored Gauss-Seidel

1: Given $\underline{\mathbf{A}}\mathbf{x} = \mathbf{b}$
2: **for** i=1,...,ζ **do**
3: **for** j=1,...,n_{colors} **do**
4: **for** k=1,...,n_{e_j} **do**
5: Compute $\mathbf{r}_{i,k} = \mathbf{b} - \underline{\mathbf{A}}\mathbf{x}_{current}$
6: Compute $\Delta\mathbf{x}_{i,k} = \underline{\mathbf{D}}_k^{-1}\mathbf{r}_{i,k}$
7: Update $\mathbf{x}_{i+1,k} = \mathbf{x}_{i,k} + \Delta\mathbf{x}_{i,k}$
8: **end for**
9: **end for**
10: Compute $R_{L,i} = \|\mathbf{r}_i\|_2$
11: If satisfied Stop.
12: **end for**

E. Generalized Minimal Residual Method

Despite the additional stability the above, implicit method affords, the SEMT and DGMT Euler equations still become difficult to solve as the number of points within a time element increases. With an increasing number of points within a time element, the Jacobian given in equation (27) proceeds farther and farther from diagonal dominance. To restore diagonal dominance, a decreasingly small pseudo-time step must be used. Thus, the potential efficiency gains of the finite-element time discretizations over time-implicit methods begin to evaporate.

To regain the efficiency improvements afforded by SEMT and DGMT, a linear solver that does not require diagonal dominance of the Jacobian should be used. The Generalized Minimum Residual method is such a solver. A welcome byproduct of this solver choice is that GMRES also more fully couples the various points in time within an element. This additional coupling arises because the Hessenberg matrix that is directly inverted as part of GMRES is constructed using the full Jacobian of the linear system coupled over all time instances; whereas, the matrix that is directly inverted during each iteration of a stationary iterative method is some easily invertible part of the full Jacobian (usually the diagonal blocks). In other words, stationary iterative methods ignore some information (and thereby some coupling), but GMRES uses this information and preserves the full coupling across all time instances.

A flexible variant of the GMRES algorithm as described by Saad[36] is used. This flexible variant allows the use of an iterative method as preconditioner. The flexible GMRES (FGMRES) algorithm proceeds as given in Algorithm (2).

Algorithm 2 : Flexible GMRES

1: Given $\underline{\mathbf{A}}\mathbf{x} = \mathbf{b}$
2: Compute $\mathbf{r}_0 = \mathbf{b} - \underline{\mathbf{A}}\mathbf{x}_0$, $\beta = \|\mathbf{r}_0\|_2$, and $\mathbf{v}_1 = \mathbf{r}_0/\beta$
3: **for** j=1,...,n **do**
4: Compute $\mathbf{z}_j := \underline{\mathbf{P}}^{-1}\mathbf{v}_j$
5: Compute $\mathbf{w} := \underline{\mathbf{A}}\mathbf{z}_j$
6: **for** i=1,...,j **do**
7: $h_{i,j} := (\mathbf{w}, \mathbf{v}_i)$
8: $\mathbf{w} := \mathbf{w} - h_{i,j}\mathbf{v}_j$
9: **end for**
10: Compute $h_{j+1,j} = \|\mathbf{w}\|_2$ and $\mathbf{v}_{j+1} = \mathbf{w}/h_{j+1,j}$
11: Define $\underline{\mathbf{Z}}_m := [\mathbf{z}_1, \ldots, \mathbf{z}_m]$, $\underline{\bar{\mathbf{H}}}_m = \{h_{i,j}\}_{1\le i\le j+1; 1\le j\le m}$
12: **end for**
13: Compute $\mathbf{y}_m = argmin_y \|\beta\mathbf{e}_1 - \underline{\bar{\mathbf{H}}}_m\mathbf{y}\|_2$ and $\mathbf{x}_m = \mathbf{x}_0 + \underline{\mathbf{Z}}_m\mathbf{y}_m$
14: If satisfied Stop, else set $\mathbf{x}_0 \leftarrow \mathbf{x}_m$ and GoTo 1.

In this description, $\underline{\mathbf{A}}$ corresponds to the full time-discretized Jacobian matrix $[\mathbf{A}]$ defined in equation (27), which may or may not be augmented with a pseudo time step, $\mathbf{b}$ corresponds to the negative of the non-linear total residual $-\mathbf{R_T}(\mathbf{U})$, and $\mathbf{x}$ is the non-linear update $\Delta\mathbf{U}$ to be computed.

Preconditioning is applied in line 4 of the algorithm. The BCGS linear solver is used for preconditioning.

A pseudo-time step must be applied to the BCGS system (i.e. added to $\underline{\mathbf{P}}$) to ensure diagonal dominance and convergence. To solve the minimization problem in line 13 of the algorithm, QR-factorization by means of Givens rotations is utilized.[36] Finally, line 14 of the algorithm indicates that this algorithm is, in fact, truncated, restarting GMRES using n Krylov vectors per restart.

Although Algorithm (2) shows the minimization problem outside the loop over Krylov vectors, in fact, this minimization is updated as each additional Krylov vector is added. This is done so that the current value of the linear residual is known for each iteration j. The FGMRES algorithm exits whenever this residual has either converged a specified amount or has converged to machine zero.

A pseudo-time step is used in the matrix $\underline{\mathbf{A}}$ in the FGMRES algorithm as well. By using this pseudo-time step within FGMRES itself the routine is modified positively as follows: an update that will avoid over-correction issues is used; the pseudo-time step is allowed to grow as the residual decreases so quadratic convergence can be retained; and the convergence rate of the FGMRES linear solver itself is increased (i.e. fewer Krylov vectors and/or restarts are required to converge the linear system) because the pseudo-time step term makes the Jacobian better conditioned for Krylov subspace methods. Thus, two different pseudo-time steps are used: one in the second-order Jacobian $\underline{\mathbf{A}}$ used by the FGMRES algorithm itself and another in the first-order Jacobian $\underline{\mathbf{P}}$ used in the BCGS portion of the preconditioner. Because the Jacobian used in the BCGS preconditioner must be diagonally dominant, the preconditioner pseudo-time step is almost always smaller and grows more slowly than the FGMRES pseudo-time step. Conversely, the pseudo-time step used in the FGMRES algorithm grows rapidly such that the diagonal term becomes vanishingly small and an exact Newton method is recovered after several orders of magnitude decrease in the non-linear residual.

1. *Preconditioning for FGMRES*

Because the pseudo-time step size needed to preserve diagonal-dominance of the Jacobian in the BCGS algorithm remains relatively small, in order to bypass this limitation, a preconditioner for FGMRES is formed as a defect-correction method applied to the residual of equation (29) as:

$$\left[\frac{V^i}{\Delta\tau_{BCGS}}\underline{\mathbf{I}}+\underline{\mathbf{J}}_{i,1stO}\right](\Delta\mathbf{U}_i^{l+1}-\Delta\mathbf{U}_i^l)= \tag{31}$$
$$-\mathbf{R_T}(\mathbf{U})-\sum_{j\neq i}\left[V^j d_i^j\underline{\mathbf{I}}\right]\Delta\mathbf{U}_j^l-\left[\frac{V^i}{\Delta\tau_{FGMRES}}\underline{\mathbf{I}}+\underline{\mathbf{J}}_i\right]\Delta\mathbf{U}_i^l$$

where the right-hand-side corresponds to the residual of equation (29). A dual-iteration strategy is required for this preconditioner (thus the term defect-correction). Inner BCGS iterations are used to invert the left-hand side matrix providing an updated value for $\Delta\mathbf{U}_{i+1}^l$, which is then substituted into the right-hand side terms, and the process is repeated, effectively driving the right-hand-side residual to zero after a number of outer iterations. The Jacobian used on the RHS is the second-order Jacobian given above. The advantage of this approach comes from the fact that the pseudo-time step on the right-hand side residual is generally much larger than that required for stability of the BCGS iterative scheme. This solver/preconditioner combination is abbreviated "GMRES-DC" for ease in this work.

F. Implementation

The various points within a time element in the SEMT and DGMT approaches are coupled and must be solved simultaneously. This coupling can be implemented serially, whereby a single point in time is solved at any given moment, and then transmits its update to the next point in time, which is then solved and the process repeated sequentially until all points within a time element have been updated. However, since the coupling only comes through a source term, each individual point within a time element may be solved in parallel with the other points within that element. This introduces an additional dimension for achieving parallelism compared to time-implicit computations, where progress in the time dimension is necessarily sequential. In this implementation, two levels of parallelism can be introduced, the first in the spatial dimension, and the second in the time dimension where the various points within a time element are solved by spawning multiple instances of the spatial solver on a parallel computing cluster. The implementation uses MPI for parallelism in the time dimension; that is, each point within a time element can utilize its own separate computational processor or, in fact, more than one. This arrangement allows computational resources to scale with the complexity of the problem.

One of the drawbacks of the SEMT and DGMT is that each point within a time element must broadcast its entire solution field to all other points within a time element, which can result in a significant amount of communication. Various strategies for communicating the different points within a time element to all processors have been investigated. Currently, a round-robin approach is implemented, where each processor sends its solution vector to a single neighboring processor. The received time-instance solution vector is added to the time derivative source term on the local processor, and then passed on to the next processor. By repeating this procedure $\Upsilon - 1$ times, where Υ is the number or points within a time element, the complete time derivative involving summations from all points within a time element is accumulated without the requirement of creating a local temporary copy of all the additional time-instance solution vectors or performing any communication intensive broadcast operations.

III. SEMT Results

The recently implemented SEMT time-discretization using backward-only coupling is validated first by solving an ordinary differential equation and then by examining a full solution of the SEMT Euler equations. In some cases, the expected theoretical error convergence behavior is observed; however, in other cases it is not. Attempts are made to explain the causes of less than theoretically possible error convergence. After the SEMT is examined, the same test cases are rerun using DGMT and their results are compared to the SEMT results.

A. Validation of the SEMT Discretization

To validate the correct implementation of the backward-only SEMT, the following ordinary differential equation is solved using the SEMT:

$$\frac{dU}{dt} = U\big(\cos(t) - b\big) \tag{32}$$

where U is the scalar solution variable and b is a constant to be specified. The exact solution of this ODE has the following form:

$$U(t) = e^{\left(\sin(t) - bt\right)} + c_0 \tag{33}$$

where c_0 is an integration constant whose value depends on the initial condition. In the case presented herein, $b = 1/100$ and the initial condition is chosen such that $U(0) = 1$ which corresponds to $c_0 = 0$. This solution contains strong periodic content with a period of 2π and a decaying transient that asymptotically approaches zero as time approaches infinity, i.e. $\lim_{t\to\infty} U(t) = 0$, as shown in Figure (1) for one period. Equation (32) is solved using the exact same solver as will be used subsequently for the flow equations with the following modifications to allow an ODE to be solved using the already established framework of the code:

1. A 4×4 grid of squares with edge length 1 is used.
2. The residual at each grid cell center is redefined according to the ODE used.
3. Preconditioning in FGMRES is turned off so that the Jacobian need not have invertible diagonal blocks.
4. The CFL number is set to a large number (larger than the inverse of machine zero) such that the diagonal term, which contains the inverse of the pseudo-time step, will be less than machine zero.

Figure (2) shows the $L_2 - norm$ of the error when the exact solution is compared to the analytical solution within the period $[0 : 2\pi]$. Each curve corresponds to a different polynomial order within each time element as given in the legend, while the x-axis gives the number of elements into which the specified time range is discretized. If the solution is sufficiently smooth, the slope of these lines should be the same as, if not greater than, the degree of the polynomial used within the elements.[37] Indeed, for $N = 4$ the slope is 4.10 between $N_{ele} = 8$ and $N_{ele} = 512$. For $N = 5$ the slope is 5.52 between $N_{ele} = 4$ and $N_{ele} = 128$. For $N = 6$ the slope is 6.56 between $N_{ele} = 4$ and $N_{ele} = 64$. For $N = 8$ the slope is 7.89 between $N_{ele} = 1$ and $N_{ele} = 16$. For $N = 10$ the slope is 9.88 between $N_{ele} = 1$ and $N_{ele} = 8$. The slopes are calculated over what appears to be the longest linear region of the error convergence curve and neglect regions that appear to either have too few elements to show the proper slope, such as between $N_{ele} = 1$ and $N_{ele} = 8$ for $N = 4$, and regions where the lower bound of machine zero seems to have affected the slope. The good agreement of these slopes with the

theoretically possible values gives confidence that the SEMT discretization is implemented correctly. Figure (3) plots error versus polynomial degree for $N_{ele} = 4$ and $N_{ele} = 8$. As can be seen, except for between $N = 8$ and $N = 10$ for $N_{ele} = 4$, the expected exponential convergence of a spectral method is exhibited.

B. Full SEMT Solution of the Euler Equations

With the implementation of the SEMT having been verified in the last section, it now follows that a full solution of the time-accurate Euler equations will be found. The flow problem chosen uses a NACA-0012 mesh with 8,747 triangular elements given in Figure (4). A test case with a free stream Mach number $M_\infty = 0.25$ is chosen to ensure no shock waves, which could be possible sources of error, are present. The airfoil motion itself is also rather simple and is given by the following equation:

$$\alpha(t) = \begin{cases} 0 & t \leq 0.05T \\ \frac{\alpha_A}{2}\left[1 - \cos\left(\frac{\omega t - 0.05T}{0.9}\right)\right] & 0.05T < t < 0.95T \\ 0 & t \geq 0.95T \end{cases} \tag{34}$$

where T is the total time frame (or period) of the solution, ω is the angular frequency that corresponds to that period, and $\alpha_A = 1.0^o$ is the maximum pitch angle attained. The reduced frequency of the motion $k_c = 0.001$ is very low so that the solution will be "quasi-steady," i.e. lift will vary linearly with angle of attack since unsteady effects are negligible. This reduced frequency corresponds to a period length of approximately $T = 10620.5$ non-dimensional seconds.

Figure (5) plots the $L_2 - norm$ of the C_L error for SEMT solutions, when compared to a reference BDF2 solution using 322,560 time steps per period, with the number of spectral elements used N_{ele} on the x-axis. It should be noted that when this RMS error is calculated, the SEMT solutions are reconstructed such that the C_L can be calculated at exactly the same 322,560 points in time as are used in the reference BDF2 solution. SEMT solutions using 1st, 2nd, 3rd, and 4th degree polynomials within the elements are shown as given in the legend. When the error convergence slopes are calculated between $N_{ele} = 4$ and $N_{ele} = 128$ for these four curves, they are found to be 1.85 for $N = 1$, 2.42 for $N = 2$, 2.60 for $N = 3$, and 2.51 for $N = 4$. Although the first-order and second-order accurate solutions, exhibit the expected slope, the higher-order solutions do not. The saw-tooth pattern of the 2nd and higher-order accurate solutions is also noteworthy. All of the relative valleys in the error occur, not coincidentally, when N_{ele} is a multiple of 20. Taking into consideration the motion of the airfoil given by equation (34), it is apparent that when N_{ele} is a multiple of 20, the endpoint of an element lies exactly on the point at which the airfoil begins its motion and also exactly on the point at which the airfoil returns to zero angle of attack.

Figure (6) plots the reference solution in solid black on the right axis and the instantaneous C_L error for the $N_{ele} = 20$ and $N_{ele} = 64$ fourth-order solutions, which have about the same C_L error $L_2 - norm$ over the time frame. As can be seen, for both solutions, the C_L error is concentrated around the points at which the airfoil begins and ends its motion. In addition, the magnitude of these oscillations is larger when an element spans this transition ($N_{ele} = 64$), than when the endpoint of an element and this transition coincide ($N_{ele} = 20$). Figure (7) plots exactly the same quantities as the previous plot, except on a scale one-hundred times finer so that the details of the error between the transition regions can be seen. This figure shows that if the transition regions were neglected, the $L_2 - norm$ of the C_L error for $N_{ele} = 64$ would indeed be considerably lower than it would be for the $N_{ele} = 20$ solution. In other words, the non-ideal error convergence observed for this case seems to result almost entirely from the transition regions.

In hindsight, this result should have been expected: the motion as defined by equation (34) is continuous in pitch angle and its first derivative; however, discontinuities exist in the second and higher derivatives. On the other hand, the polynomial basis is continuous in all derivatives (although all derivatives higher than the degree of the polynomial are identically zero). Because of these facts, it seems intuitive that having such a polynomial span a second-derivative discontinuity would causes oscillations in that polynomial. When the SEMT discretization is applied to first-order partial differential equations, like the Euler equations, derivative discontinuities are allowed at the end points of the elements; hence, when the end point of an element coincides with the point at which the discontinuity occurs, there are fewer oscillations, but, as observed, oscillations still do occur, contrary to what would be expected. It is believed that, although the discontinuity in pressure (which is integrated to find lift) occurs at exactly the end point of a spectral-element in time for the cells that have a boundary face on the airfoil surface, it would occur at a later time, i.e. not

at the element end point in the next layer of cells because of the additional time needed for the fluid to convect to those cells after the airfoil begins its motion. In essence, the first-order effect of the transition of the airfoil into motion can be mitigated by having a spectral element in time end at the point at which the airfoil motion begins; however, the second-order effect, in the off-body cells, still affects the C_L solution by causing oscillations in the subsequent spectral element in time.

C. An ODE with Second-derivative Discontinuity

To confirm the accuracy of the previous analysis regarding derivative discontinuities, an ordinary differential equation whose solution is similar to the motion as described by equation (34) is solved numerically. This ODE and its solution are given as follows, respectively:

$$\frac{dU}{dt} = \begin{cases} 0 & t \leq \frac{\pi}{10} \\ \frac{U_A}{0.9}\sin\left(\frac{t-\frac{\pi}{10}}{0.9}\right) & \frac{\pi}{10} < t < \frac{19\pi}{10} \\ 0 & t \geq \frac{19\pi}{10} \end{cases} \tag{35}$$

$$U(t) = \begin{cases} 0 & t \leq \frac{\pi}{10} \\ \frac{U_A}{2}\left[1-\cos\left(\frac{t-\frac{\pi}{10}}{0.9}\right)\right] & \frac{\pi}{10} < t < \frac{19\pi}{10} \\ 0 & t \geq \frac{19\pi}{10} \end{cases} \tag{36}$$

where for the chosen problem $U_A = 6$ is the height of the impulse and the period as shown is necessarily 2π. Figure (8) plots the $L_2 - norm$ of the solution error over one period versus N_{ele} for elements with the specified polynomial degree. As can be seen, the same sawtooth pattern as was exhibited by the error in the higher-order SEMT solutions of the Euler equations is also exhibited by the error in the SEMT solutions of this ODE. The first-order and second-order SEMT error curves have average slopes of 0.99 and 2.02, respectively, between $N_{ele} = 4$ and $N_{ele} = 1024$. If only the N_{ele} numbers that are not multiples of 20 are considered, the higher-order SEMT error curves track the $N = 2$ curve pretty closely, i.e. their accuracy is reduced to second order. However, if only those N_{ele} numbers that are multiples of 20 are considered, the slope of the $N = 3$ curve is 3.01 between $N_{ele} = 20$ and 960, the slope of the $N = 4$ curve is 4.01 between $N_{ele} = 20$ and 480, and the slope of the $N = 5$ curve is 5.31 between $N_{ele} = 20$ and 120. For this ODE, as long as the second-order discontinuity occurs exactly at the endpoint of an element, the expected order of error convergence is observed. If, however, this discontinuity occurs within an element, the error converges at close to second-order accuracy regardless of the degree of polynomials used within the spectral elements in time. Figure (9) plots the exact solution on the right axis and the instantaneous error on the left axis for the two solutions noted in the legend. The two solutions plotted have similar $L_2 - norm$s of error over the period considered. As can be seen, for the $N_{ele} = 20$ solution, the instantaneous error varies sinusoidally over the period and somewhat within each element. On the other hand, the error in the $N_{ele} = 1024$ solution oscillates at the second-derivative discontinuity, and then tracks the exact solution closely, except with a constant 3×10^{-7} offset, before it oscillates again at the second discontinuity, after which is has a small offset from zero. This solution also highlights that the oscillations that occur in the error of the SEMT Euler equations when N_{ele} is a multiple of 20 are unique to those solutions, and may very well be due to the second-order flow effects discussed above.

D. A Continuously Differentiable Airfoil Motion

The goal of the exploration of the SEMT solution of the Euler equations herein is to demonstrate the expected error convergence for higher-order elements. Thus far, this goal has not been achieved; however, the analysis up to this point has indicated that the solution of the NACA-0012 airfoil undergoing a continuously differentiable motion might produce the desired results.

For this second test case, a similar motion as was used for the first test case is prescribed. An airfoil starts at zero angle of attack and then pitches up to a prescribed angle and next pitches back down to zero, but without the beginning and ending intervals where it remains at zero angle of attack. This motion is given by the following equation:

$$\alpha(t) = \alpha_A \frac{1-\cos\left(\omega t\right)}{2}. \tag{37}$$

Before this case is considered for the Euler equations, a similar ODE is solved to confirm the expected behavior. This ODE and its solution are given as follows:

$$\frac{dU}{dt} = U_A \frac{\sin t}{2} \tag{38}$$

$$U(t) = U_A \frac{1-\cos t}{2}. \tag{39}$$

Here, $U_A = 6.0$ has the same value as in the previous ODE given by equation (35). Figure (10) plots the error curves, as has been done previously, for $N = 4, 5, 6, 8$, and 10. All curves exhibit the expected slope in at least some limited region (the very limited region between 2 and 4 elements in the case of 10th order elements). Figures (11) and (12), which plot the instantaneous error over the period, show that the error both among the elements and within the elements themselves, is well behaved and conforms to expectations.

With the expected results of the SEMT numerical solution of the ODE analogous to the prescribed airfoil motion having been confirmed, the focus now transitions to the SEMT solution of the Euler equations for the motion described by equation (37) with $\alpha_A = 8.0^o$, for this case, $M_\infty = 0.25$, and $k_c = 0.001$, as previously. The value of α_A was increased to increase the maximum magnitude of C_L and hopefully with it the range of error produced in the solutions. It should be noted that $\alpha_A = 8.0^o$ is still well within the linear region of the lift curve for the NACA-0012 airfoil. Figure (13) plots the C_L error convergence curves for spectral elements of various polynomial degree. The slopes for 1st and 2nd order accurate schemes appear to have the appropriate values. For $N = 3$ the slope between $N_{ele} = 3$ and 64 is 3.80; for $N = 4$ the slope between $N_{ele} = 2$ and 12 is 5.07. Finally, for $N = 5$ the slope between $N_{ele} = 3$ and 8 is 5.44. As can be seen, the C_L error in these higher-order elements converges as expected over a limited range. However, as is also apparent from this figure, after $N_{ele} = 64$ for the $N = 3$ curve, after $N_{ele} = 12$ for the $N = 4$ curve, and after $N_{ele} = 8$ for the $N = 5$ curve, the slope of the C_L error convergence changes abruptly to around first-order accuracy. To ascertain why this change might occur, instantaneous C_L error plots are examined in Figures (14-16). The first of these figures shows two solutions with about the same C_L error $L_2 - norm$ over the period. The second figure is identical to the first except the range of the y-axis is changed so that the smaller variations in C_L error can be examined more closely. The last of these figures examines the error for a much more highly resolved solution than the fist two. Perhaps the first thing that is noticed in these figures is that two periods of the solution are plotted. As can be seen, there is a very large error in the first spectral element. This large error, on the order of 10^{-4} is caused because of a mis-specification of the initial condition. The steady state solution of the airfoil at zero angle of attack was used for the initial condition; however, even though the motion of the airfoil is slow enough to be considered "quasi-steady," it is not equivalent to an actual steady flow. As such, it is nearly impossible to specify the correct initial condition for such a motion.

Because of this initial error, Figure (13) uses the second period of motion for its results. Three possible explanations as to why the slope of the error convergence decreases after a certain point as the number of spectral elements in time increases are as follows:

1. Since the problem is solved hyperbolically, but is actually periodic in nature, transients exist that have not yet been dissipated by the second period

2. The initial oscillations introduce an offset to the solution, as is seen in the solution of the ODE given in Figure (9)

3. The initial oscillations cause periodically repeating, but smaller, oscillations in the entire future time history of the solution

These possibilities will now be examined in the order presented. The first seems unlikely. When the instantaneous error is examined in Figure (16), it very clearly repeats in almost exactly the same pattern for the first and second periods (after the initial oscillations) and especially in the second half of the period. If transients were still dissipating, it would be expected that the pattern of the error in the second period would be dissimilar to that of the first.

The second possibility also seems unlikely. From all three of the instantaneous error plots, and especially those two with a narrower range, no clear offset is visible. The error appears to oscillate randomly around a mean of zero. It should be noted, however, that the C_L error that is being examined is error in an integrated quantity. Such offsets might instead manifest themselves as offsets in the pressure in individual

cells surrounding the airfoil. Still, it seems intuitive that such pressure offsets would necessarily lead to a constant lift offset as well. For this reason, the second explanation is also discounted.

Finally, the third possibility seems most likely. As can be seen from all three instantaneous error plots, there are large oscillations in the C_L error, even after the initial interval. These oscillations have no corollary in the ODE error plots produced in this work. Additionally, the error oscillations repeat in the second period almost identically to the first. Certainly, it seems completely believable that an initial oscillation on the order of 10^{-4} could cause residual oscillations on the order of 10^{-8} even a long time later. For these reasons, the idea that the initial oscillations corrupt the rest of the solution seems most plausible.

It should be noted that all of the SEMT results have been converged to a residual tolerance of 1×10^{-14} using the GMRES-DC solver, as described above, applied to the backward-only coupling of the SEMT Euler equations. It is possible that the discontinuous-Galerkin methods in time (DGMT) might offer better error convergence than the SEMT.[14] The additional degree of freedom allowed by the discontinuous nature of the end points of each time element in the DGMT might cause oscillations in elements that contain discontinuities to be less severe. The breadth of literature on DGMT[38,13,14,39,40,41,42] is impressive, and, DGMT is compared to SEMT in the next section.

IV. DGMT Results

The DGMT as outlined above will now be applied to the same test problems to which the SEMT was applied in the last section. It is expected that the error of the DGMT will converge at least as order $N+1$ where, as before, N is the degree of the polynomial basis used within the elements. In other words, it is expected that the h-convergence curves will have slopes equivalent to the number of degrees of freedom in time for each time element.

Figure (17) compares the convergence of SEMT and DGMT for the ODE given in equation (32) and for expected fourth, fifth, sixth, eighth, and tenth orders of accuracy. As can be seen, if N_{SE} is the degree of the polynomial basis used in the SEMT solutions and N_{DG} is the degree of the polynomial basis used in the DGMT solutions, then $N_{DG} = N_{ST} - 1$ solutions converge at about the same order-of-accuracy, as expected. In other words, the two lines in each subfigure of Figure (17) have about the same slope, although there can be an offset.

Figure (18) examines the behavior of the first Euler equations test case, whose motion is described by equation (34). As in the last figure, DGMT and SEMT are compared for different orders-of-accuracy in each subplot. The first thing to note is that for comparison of C_L-error using the Euler equations, $N_{DG} = N_{ST}$ solution errors converge at about the same order; in other words, the expected one-order-higher error convergence of DGMT as compared to SEMT does not hold for the more complicated Euler equations. Indeed, for $N_{DG} = N_{ST} = 1$, the SEMT curve has slope 1.85 and the DGMT curve has slope 1.99; for $N_{DG} = N_{ST} = 2$, the SEMT curve has slope 2.42 and the DGMT curve has slope 2.46; for $N_{DG} = N_{ST} = 3$, the SEMT curve has slope 2.60 and the DGMT curve has slope 2.62. All of these slopes are calculated between $N_{ele} = 4$ and $N_{ele} = 128$ for both SEMT and DGMT. It should be noted that no DGMT comparison data for $N_{ST} = 4$ was produced. The second trend that should be noted is that the sawtooth pattern present in the SEMT data is repeated in the DGMT data. This disproves the hypothesis that perhaps the added flexibility in allowing the value-discontinuity at the endpoints might help to lessen the impact of a second-derivative discontinuity within an element on the error convergence.

Next, the convergence of the ODE given by equation (35) is compare in Figure (19). As can be seen, the $N_{DG} = N_{ST} - 1$ error convergence equality has been restored for this ODE. Secondly, the sawtooth pattern is again present, as expected. For $N_{DG} = N_{ST} - 1 = 2$, the slope of the SEMT curve is 2.02, as above, while the slope of the DGMT curve is 1.99 between $N_{ele} = 4$ and $N_{ele} = 1024$. Again, if only the $N_{ele} \neq \times 20$ points are considered DGMT error convergence reduces to around second-order accuracy. On the other hand, if only the points that are multiples of twenty are considered, the orders of error convergence are as expected. Specifically, the slope of the $N_{DG} = N_{ST} - 1 = 2$ curve is 3.01 for SEMT and 3.00 for DGMT between $N_{ele} = 20$ and 960, the slope of the $N_{DG} = N_{ST} - 1 = 3$ curve is 4.01 for SEMT and 4.00 for DGMT between $N_{ele} = 20$ and 480, and the slope of the $N_{DG} = N_{ST} - 1 = 4$ curve is 5.31 for SEMT and 4.99 for DGMT between $N_{ele} = 20$ and 120. Thus, a second-derivative discontinuity affects SEMT and DGMT in exactly the same way.

When SEMT and DGMT are compared for the ODE given by equation (38) in Figure (20), the two methods can be seen to proceed as expected for all orders of accuracy tested.

Finally, Figure (21) compares DGMT to SEMT for the Euler equations and airfoil motion give by equation (37). As can be seen, as with the previous comparison involving the Euler equations, C_L-error convergence curves have about the same slopes when $N_{DG} = N_{ST}$. Otherwise, the corresponding error convergence curves look very similar. For $N_{DG} = N_{ST} = 1$, the slope between $N_{ele} = 4$ and $N_{ele} = 128$ is 1.84 for SEMT and 1.98 for DGMT. For $N_{DG} = N_{ST} = 2$, the slope between $N_{ele} = 4$ and $N_{ele} = 128$ is 2.93 for SEMT and 3.03 for DGMT. For $N_{DG} = N_{ST} = 3$, the slope between $N_{ele} = 4$ and $N_{ele} = 32$ is 3.98 for SEMT and 4.03 for DGMT. For $N_{DG} = N_{ST} = 4$, the slope between $N_{ele} = 2$ and $N_{ele} = 8$ is 5.00 for SEMT and 5.01 for DGMT.

Thus, for sufficiently smooth solutions of the Euler equations, it can be seen that the DGMT with $N_{DG} = N_{ST}$ has error convergence that is slightly better than SEMT. However, it was expected that DGMT error would converge like $N_{DG} = N_{ST} - 1$, as occurs for the ODEs tested in this section. Given that DGMT experiences the same order-reduction as SEMT when discontinuities exist in the solution time history, SEMT seems like the better choice for solution of the Euler equations since SEMT has one fewer degree of freedom per time element than DGMT when $N_{DG} = N_{ST}$.

V. Conclusions and Future Work

The spectral-element method in time shows promise for the time-accurate solution of the Euler equations. In this preliminary investigation, slow moving, low Mach number problems were considered for validation purposes. Even with these "easy" problems, some obvious areas for concern and improvement have been identified. Primarily, higher-order SEMT does not handle discontinuities well. This is not limited to discontinuities in the solution variables only, but in their time derivatives as well. This issue must be addressed for higher-order SEMT to become competitive. Fortunately, experience with higher-order spatial methods already provides two paths for addressing this issue. First, h-refinement around discontinuities and order-reduction of elements containing those discontinuities provides the first path forward. An alternative path exists whereby using artificial viscosity in time to damp oscillations within elements containing discontinuities could also achieve the desired results. In the present work, the spectral-element method described falls into the family of continuous-Galerkin methods. It was thought that by allowing discontinuity in the solution variables themselves at element end points, the oscillations produced by derivative discontinuities within an element might decrease in amplitude. As such, discontinuous-Galerkin methods in time (DGMT) were also attempted, but found to have exactly the same issues with derivative discontinuities as SEMT.

As mentioned previously, only the backward-only coupling of the SEMT and DGMT discretizations have been implemented and tested. Not only was this coupling method easiest to implement, it also could use the already developed first-order accurate quasi-periodic time-spectral solver, presented in previous work,[3] efficiently without modification because, within each element, the SEMT and DGMT matrices are dense. Although, this same solver could be used for the monolithic-time approach (with the SEMT or DGMT matrix), and the periodic time-spectral solver[3] could be used for periodic SEMT and DGMT, such use would be inefficient. The full SEMT and DGMT matrices for either of these two latter couplings is not dense, but block-diagonal. As such, these matrices contain many zeroes, which would be included if the existing solvers were used, since they assume dense matrices. In future work, these two additional coupling methods for SEMT and DGMT will be implemented with their own specific solvers to retain the efficiency these couplings afford.

Although, more work is needed to make either the SEMT or DGMT time-marching schemes generally applicable to a broad range of solutions of the Euler and Navier-Stokes equations, initial indications point toward the SEMT being the better choice as it offers nearly the same level of error as DGMT for a given number of temporal elements, but with one less degree of freedom per element.

References

[1]Kurdi, M. H. and Beran, P. S., "Spectral element method in time for rapidly actuated systems," *Journal of Computational Physics*, Vol. 227, No. 3, 2008, pp. 1809–1835.

[2]Ben-Tal, A., Bar-Yoseph, P., and Flashner, H., "Space-time spectral element method for optimal slewing of a flexible beam," *International Journal of Numerical Methods in Engineering*, Vol. 39, No. 18, 1996, pp. 3101–3121.

[3]Mundis, N. L., *The development of a robust, efficient solver for spectral and spectral-element time discretizations*, Ph.D. thesis, University of Wyoming, 2014.

[4]Beran, P., Petit, C., Millman, D., and Halley, J., "Uncertainty quantification of limit-cycle oscillations," *Journal of*

Computational Physics, Vol. 217, No. 1, 2006, pp. 217–247.

[5]Borri, M., "Helicopter rotor dynamics by finite element time approximation," *Computers Mathematics with Applications*, Vol. 12A, No. 1, 1986, pp. 149–160.

[6]Oden, J., "A general theory of finite elements II. Applications," *International Journal of Numerical Methods in Engineering*, Vol. 1, No. 3, 1969, pp. 247–259.

[7]Desai, C., Oden, J., and Johnson, L., "Evaluation and analyses of some finite element and finite difference procedures for time dependent problems," Tech. Rep. Accession No.: ADA009739, AEWES-Misc-Paper-S-75-7, 1975.

[8]Fried, I., "Finite element analysis of time-dependent phenomena," *AIAA Journal*, Vol. 7, No. 6, 1969, pp. 1170–1173.

[9]Argyris, J. and Scharpf, D., "Finite elements in time and space," *Nuclear Engineering and Design*, Vol. 10, No. 4, 1969, pp. 456–464.

[10]Palaniappan, K., Beran, P., and Jameson, A., "Optimal control of LCOs in aerostructural systems," AIAA Paper 2006-1621, May 2006.

[11]Nadarajah, S. and Jameson, A., "Optimum shape design for unsteady flows with time-accurate continuous and discrete adjoint methods," *AIAA Journal*, Vol. 45, No. 7, 2007, pp. 1478–1491.

[12]Riff, R. and Baruch, M., "Time finite element discretization of Hamilton's law of varying action," *AIAA Journal*, Vol. 22, No. 9, 1984, pp. 1310–1318.

[13]Aharoni, D. and Bar-Yoseph, P., "Mixed finite element formulations in the time domain for solution of dynamics problems," *computational Mechanics*, Vol. 9, No. 5, 1992, pp. 359–374.

[14]Bar-Yoseph, P., Fisher, D., and Gottlieb, O., "Spectral element methods for nonlinear temporal dynamical systems," *Computational Mechanics*, Vol. 18, No. 4, 1996, pp. 302–313.

[15]Tezduyar, T., Behr, M., and Liou, J., "New strategy for finite element computations involving moving boundaries and interfaces. The deforming-spatial domain space-time procedure. I. The concept and the preliminary numerical tests," *Computer Methods in Applied Mechanics and Engineering*, Vol. 94, No. 3, 1992, pp. 339–351.

[16]Tezduyar, T., Sathe, S., Keedy, R., and Stein, K., "Space-time finite element techniques for computation of fluid-structure interactions," *Computer Methods in Applied Mechanics and Engineering*, Vol. 195, No. 17-18, 2006, pp. 2002–2027.

[17]Pontaza, J. and Reddy, J., "Space-time coupled spectral/hp least-squares finite element formulation for the incompressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 197, No. 2, 2004, pp. 418–459.

[18]Bell, B. and Surana, K., "A space-time coupled p-version least squares finite element formulation for unsteady two-dimensional Navier-Stokes equations," *International Journal for Numerical Methods in Engineering*, Vol. 39, No. 15, 1996, pp. 2593–2618.

[19]Klaij, C., van der Vegt, J., and van der Ven, H., "Space-time discontinuous Galerkin method for the compressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 217, No. 2, 2006, pp. 589–611.

[20]Klaij, C., van der Vegt, J., and van der Ven, H., "Pseudo time-stepping methods for space-time discontinuous Galerkin discretizations of the compressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 219, No. 2, 2006, pp. 622–643.

[21]van der Vegt, J. and van der Ven, H., "Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressibel flows. I. General formulation," *Journal of Computational Physics*, Vol. 182, No. 2, 2002, pp. 546–585.

[22]Bar-Yoseph, P. and Moses, E., "Space-time spectral element methods for unsteady convection diffusion problems," *International Journal of Numerical Methods for Heat and Fuild Flow*, Vol. 7, No. 2-3, 1997, pp. 215–235.

[23]Bar-Yoseph, P., Moses, E., Zrahia, U., and Yarin, A., "Space-time spectral element methods for one-dimensional nonlinear advection-diffusion problems," *Journal of Computational Physics*, Vol. 119, No. 1, 1995, pp. 62–74.

[24]Hulbert, G. and Hughes, T., "Space-time finite element methods for second-order hyperbolic equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 84, No. 3, 1990, pp. 327–348.

[25]Fehlberg, E., "Classical fifth-, sixth-, seventh-, and eighth-order Runge-Kutta formulas with step size control," Tech. rep., NASA Technical Report, 1968.

[26]Newmark, N., "A method of computation for structural dynamics," *ASCE Journal of the Engineering Mechanics Division*, Vol. 5, No. EM3, 2002, pp. 67–94.

[27]Hilber, H., Hughes, T., and Taylor, R., "Improved numerical dissipation for time integration algorithms in structural dynamics," *Earthquake Engineering and Structural Dynamics*, Vol. 5, 1977, pp. 283–292.

[28]Yao, J., Jameson, A., Alonso, J., and Liu, F., "Development and validation of a massively parallel flow solver for turbomachinery flows," *Journal of Propulsion and Power*, Vol. 17, No. 3, 2001, pp. 659–668.

[29]Cockburn, B. and Shu, C., "Runge-Kutta discontinuous Galerkin methods for convection-dominated problems," *Journal of Scientific Computing*, Vol. 16, No. 3, 2001, pp. 173–261.

[30]Giles, M., "Stability analysis of a Galerkin/Runge-Kutta Navier-Stokes discretisation on unstructured tetrahedral grids," *Journal of Computational Physics*, Vol. 132, No. 2, 1997, pp. 201–214.

[31]Nikitin, N., "Third-order-accurate semi-implicit Runge-Kutta scheme for incompressible Navier-Stokes equations," *International Journal for Numerical Methods in Fluids*, Vol. 51, No. 2, 2006, pp. 221–233.

[32]Burnett, D., *Finite Element Analysis: from Concepts to Applications*, Addison-Wesley, 1988.

[33]Helenbrook, B. T., "Artificial Compressibility Preconditioning for Incompressible Flows Under All Conditions," AIAA Paper 2006-0689, 2006.

[34]Helenbrook, B. T. and Cowles, G. W., "Preconditioning for dual-time-stepping simulations of the shallow water equations including Coriolis and bed friction effects," *Journal of Computational Physics*, Vol. 227, No. 9, 2008, pp. 4425–4440.

[35]Sicot, F., Puigt, G., and Montagnac, M., "Block-Jacobi Implicit Algorithm for the Time Spectral Method," *AIAA Journal*, Vol. 46, No. 12, 2008, pp. 3080–3089.

[36]Saad, Y., *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, 1996.

[37]Szabó, B. and Babuška, I., *Finite Element Analysis*, John Wiley & Sons, 1991.

[38]Bar-Yoseph, P., "Time finite element methods for initial value problems," *Applied Numerical Mathematics*, Vol. 33, No. 1-4, 2000, pp. 435–445.

[39]Bar-Yoseph, P. and Elata, D., "An efficient L_2 Galerkin finite element method for multi-dimensional nonlinear hyperbolic systems," *International Journal of Numerical Methods in Engineering*, Vol. 29, No. 6, 1990, pp. 1229–1245.

[40]Bar-Yoseph, P., Elata, D., and Israeli, M., "On the generalized L_2 Galerkin finite element method for linear hyperbolic equations," *International Journal of Numerical Methods in Engineering*, Vol. 36, No. 4, 1993, pp. 679–694.

[41]Hulbert, G. M., "Time finite element methods for structural dynamics," *International Journal of Numerical Methods in Engineering*, Vol. 33, No. 2, 1992, pp. 307–331.

[42]Li, X. D. and Wiberg, N.-E., "Structural dynamic analysis by a time-discontinuous Galerkin finite element method," *International Journal of Numerical Methods in Engineering*, Vol. 39, No. 12, 1996, pp. 2131–2152.

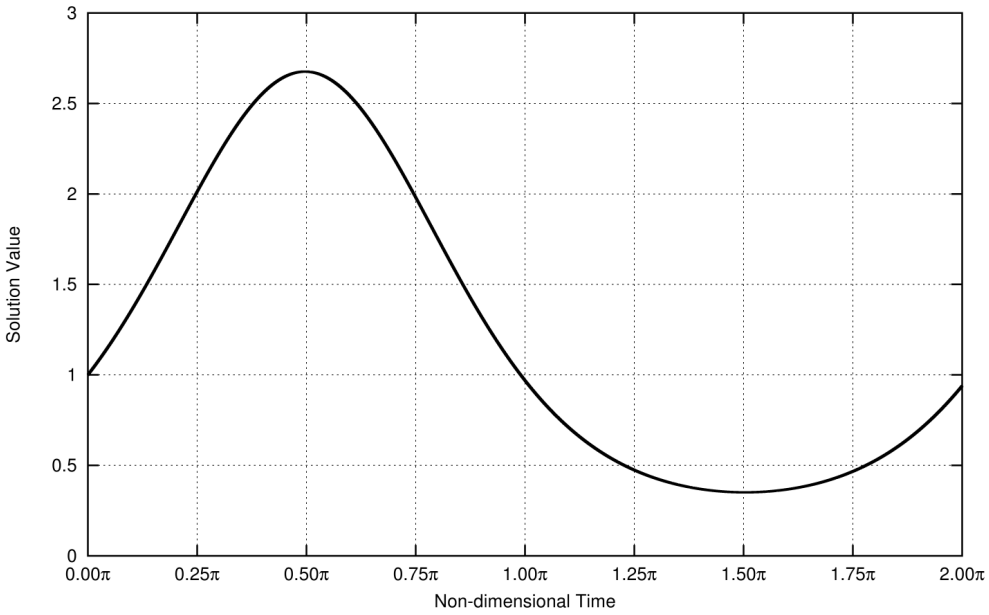


Figure 1. Exact solution of the ODE in equation (32) with $b = 1/100$ and $U(0) = 1$

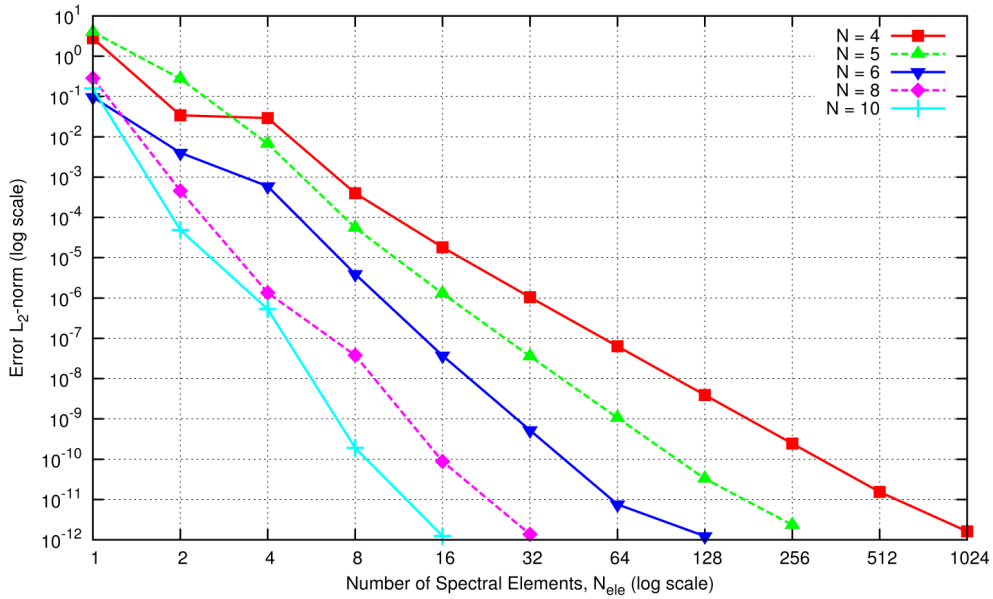


Figure 2. $L_2 - norm$ **of the SEMT solution error for the ODE in equation (32) with the given number of spectral elements in time (x-axis) and the specified polynomial degree**

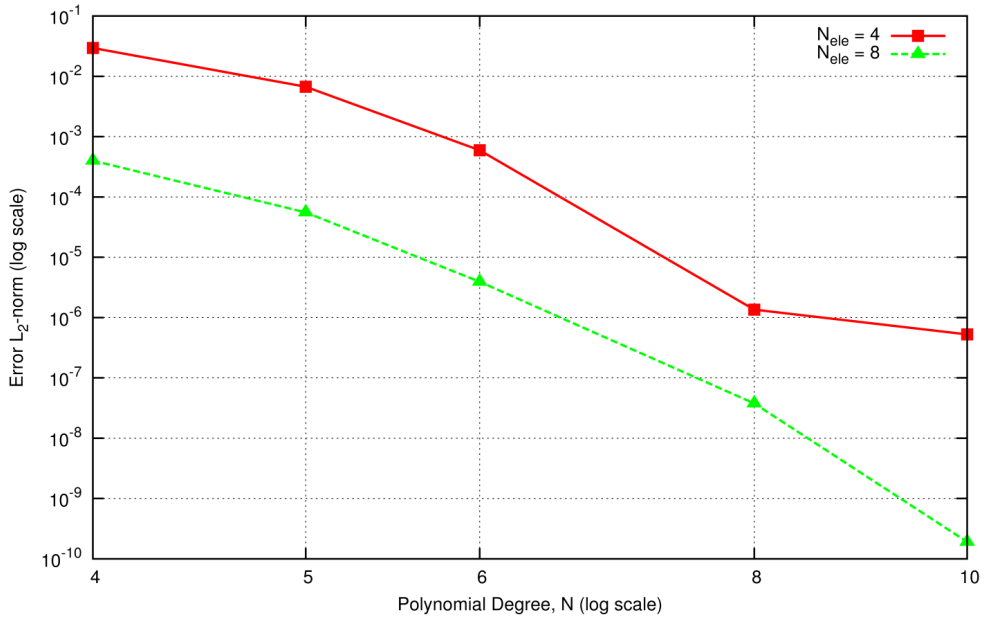


Figure 3. $L_2 - norm$ **of the SEMT solution error for the ODE in equation (32) with the given polynomial degree (x-axis) and the specified number of spectral elements in time**

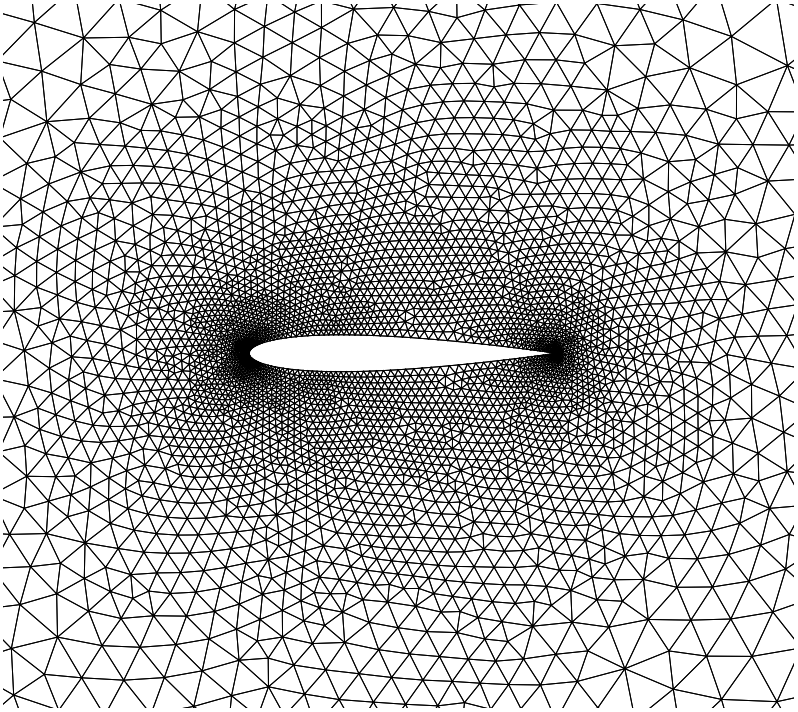


Figure 4. Near field mesh for the NACA-0012 airfoil

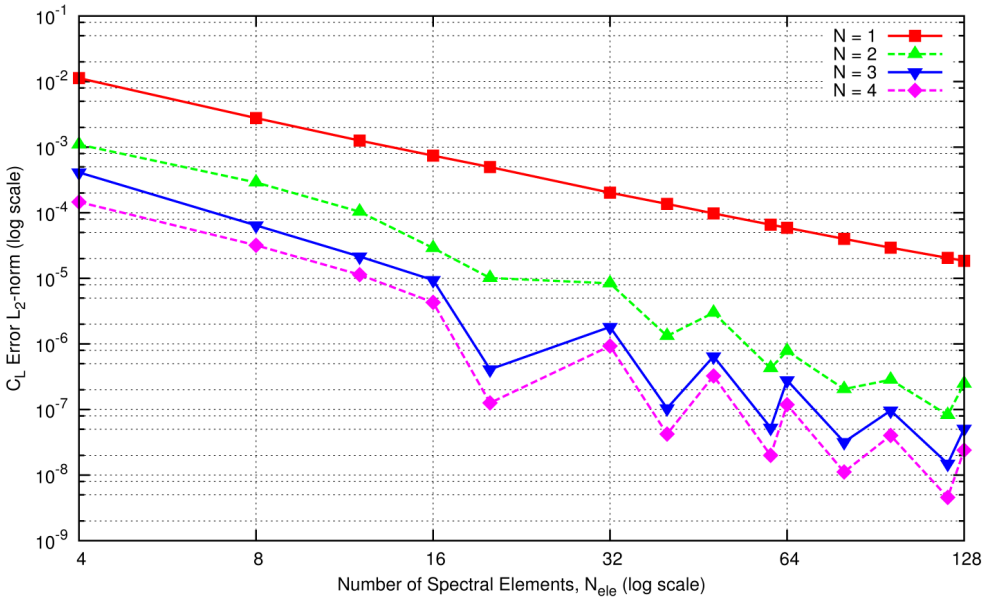


Figure 5. $L_2 - norm$ **of the SEMT solution** C_L **error for the Euler equations with the airfoil motion prescribed in equation (34) with the given number of spectral elements in time (x-axis) and the specified polynomial degree**

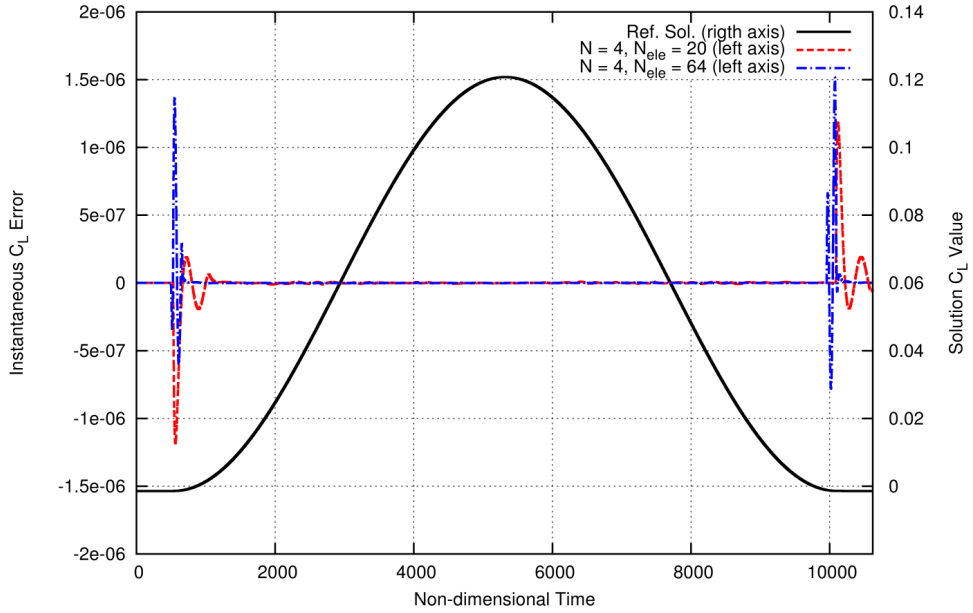


Figure 6. Instantaneous C_L error of the SEMT solutions versus non-dimensional time for the Euler equations with the airfoil motion prescribed in equation (34) and specified polynomial degree and number of spectral elements in time

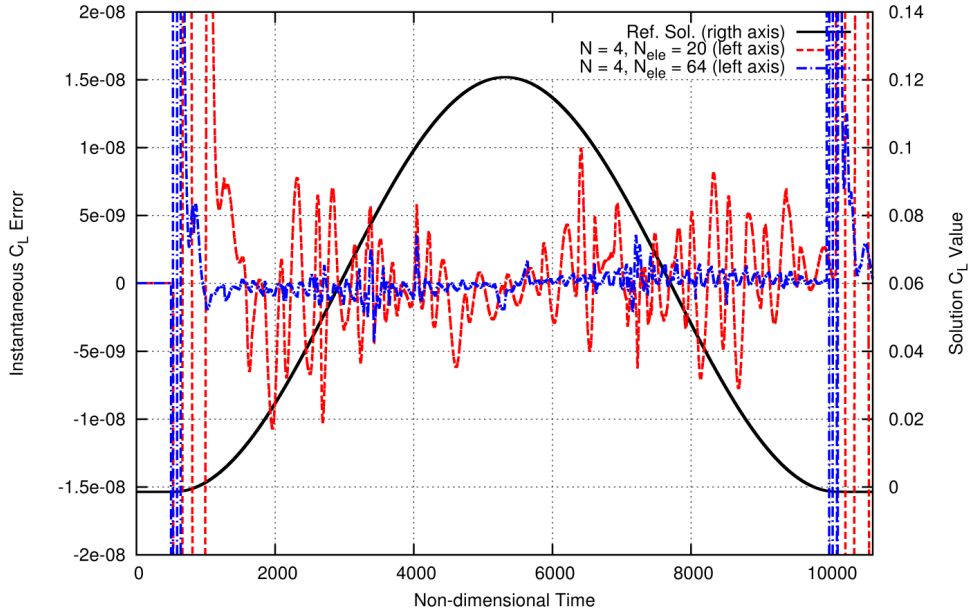


Figure 7. Instantaneous C_L error of the SEMT solutions versus non-dimensional time for the Euler equations with the airfoil motion prescribed in equation (34) and specified polynomial degree and number of spectral elements in time

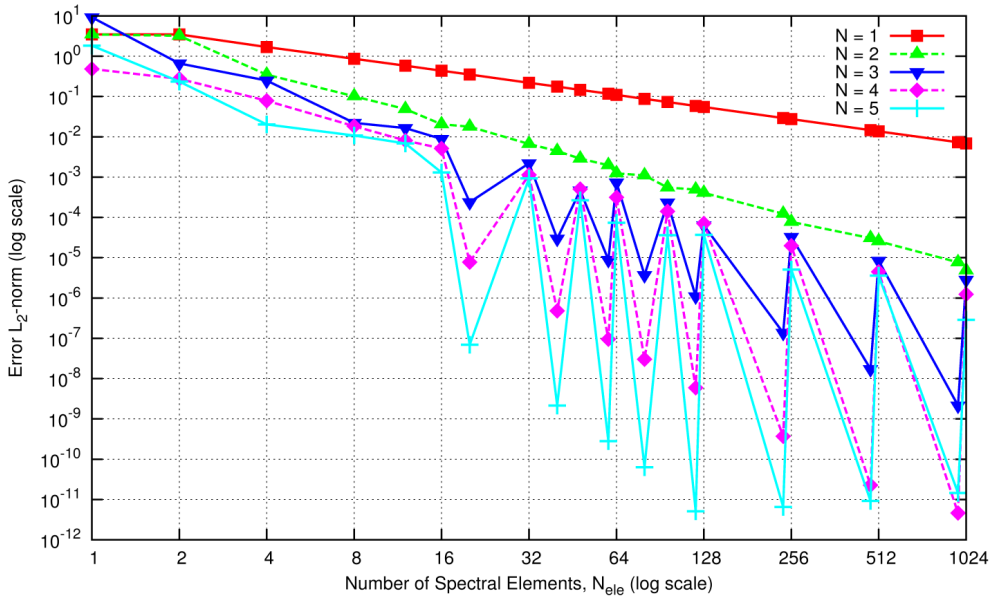


Figure 8. $L_2 - norm$ **of the SEMT solution error for the ODE in equation (35) with the given number of spectral elements in time (x-axis) and the specified polynomial degree**

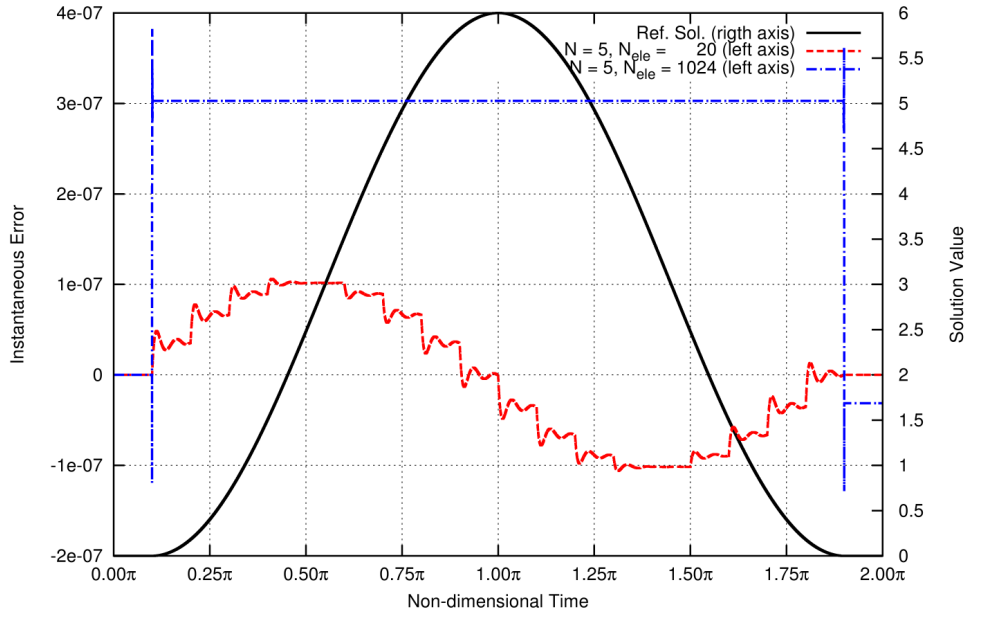


Figure 9. Instantaneous error of the SEMT solutions versus non-dimensional time for the ODE in equation (35) and specified polynomial degree and number of spectral elements in time

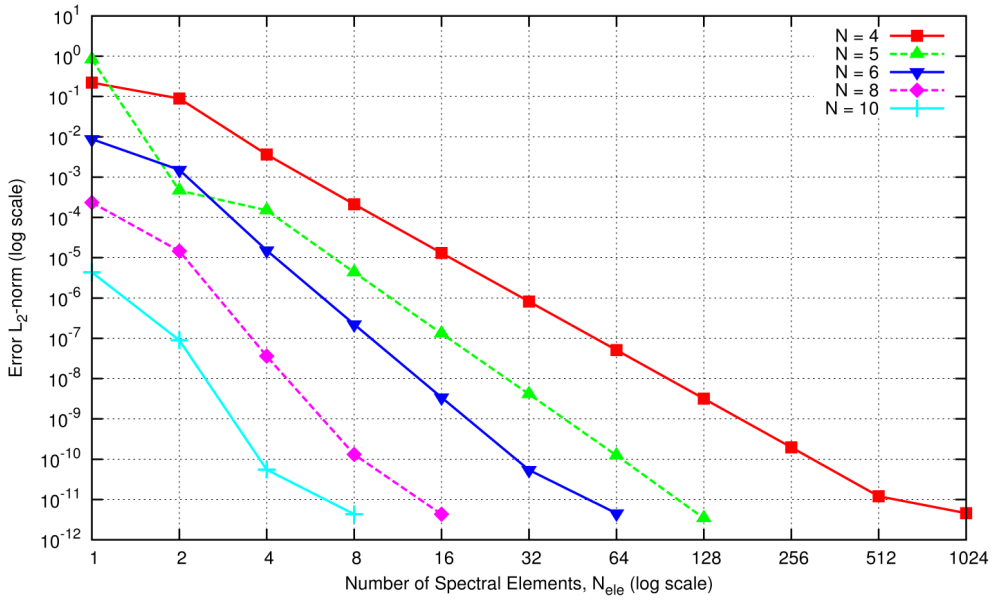


Figure 10. $L_2 - norm$ **of the SEMT solution error for the ODE in equation (38) with the given number of spectral elements in time (x-axis) and the specified polynomial degree**

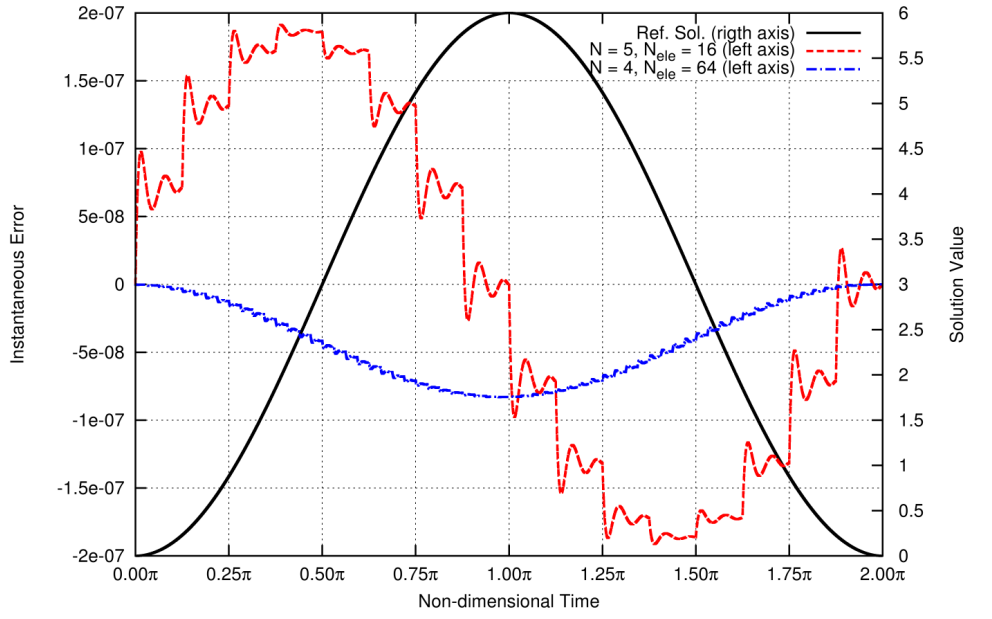


Figure 11. Instantaneous error of the SEMT solutions versus non-dimensional time for the ODE in equation (38) and specified polynomial degree and number of spectral elements in time

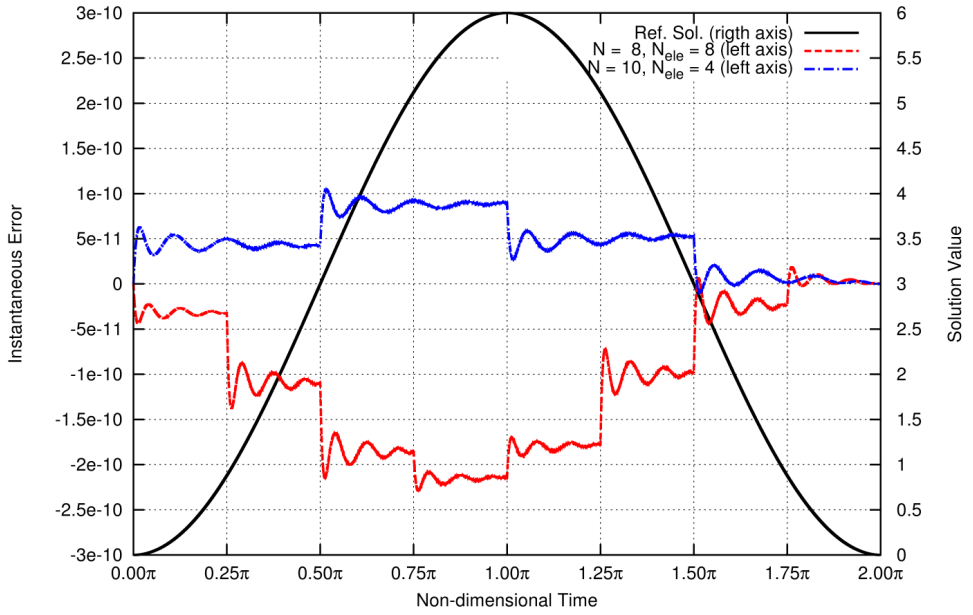


Figure 12. Instantaneous error of the SEMT solutions versus non-dimensional time for the ODE in equation (38) and specified polynomial degree and number of spectral elements in time

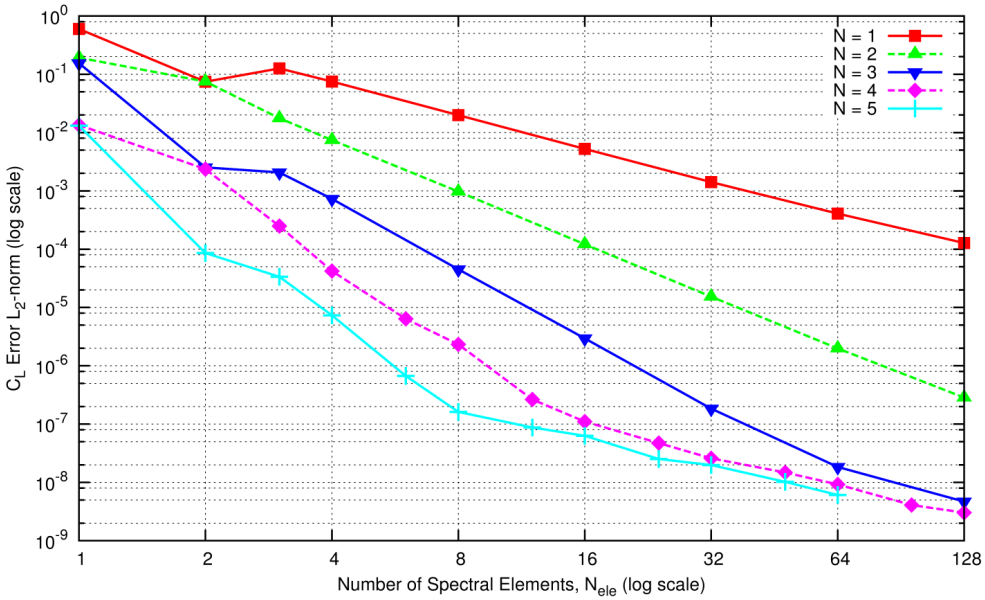


Figure 13. $L_2 - norm$ of the SEMT solution C_L error for the Euler equations with the airfoil motion prescribed in equation (37) with the given number of spectral elements in time (x-axis) and the specified polynomial degree

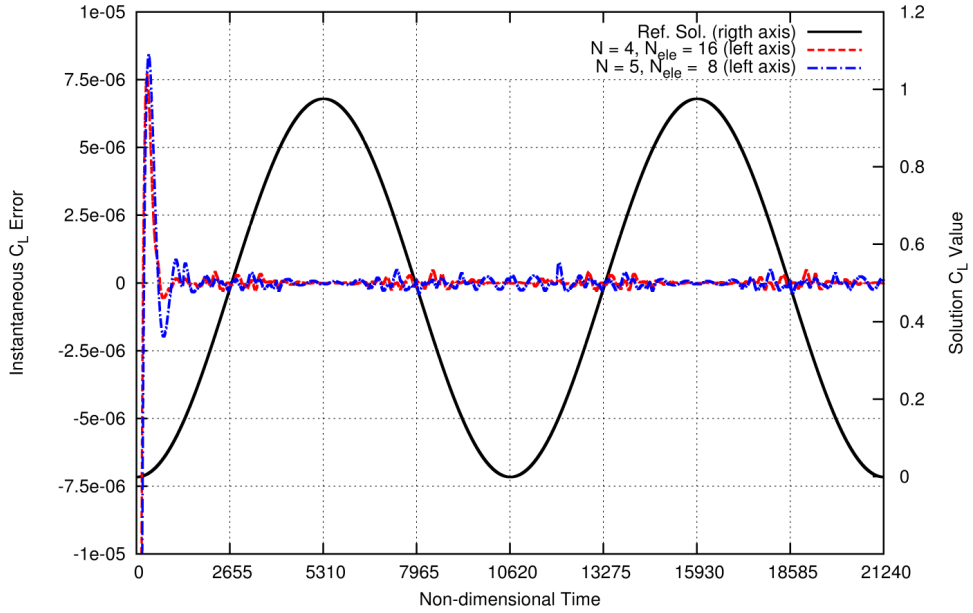


Figure 14. Instantaneous C_L error of the SEMT solutions versus non-dimensional time for the Euler equations with the airfoil motion prescribed in equation (37) and specified polynomial degree and number of spectral elements in time

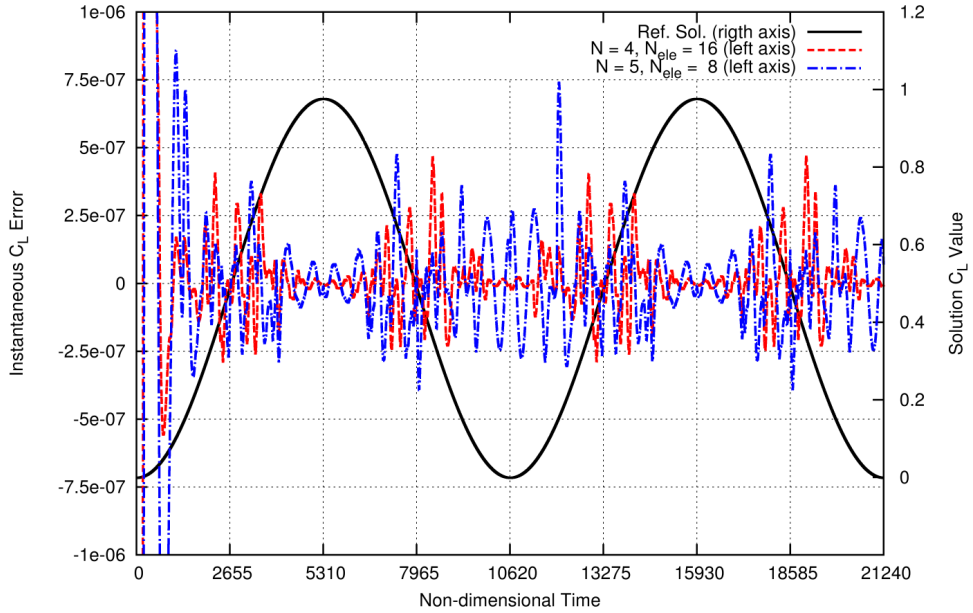


Figure 15. Instantaneous C_L error of the SEMT solutions versus non-dimensional time for the Euler equations with the airfoil motion prescribed in equation (37) and specified polynomial degree and number of spectral elements in time

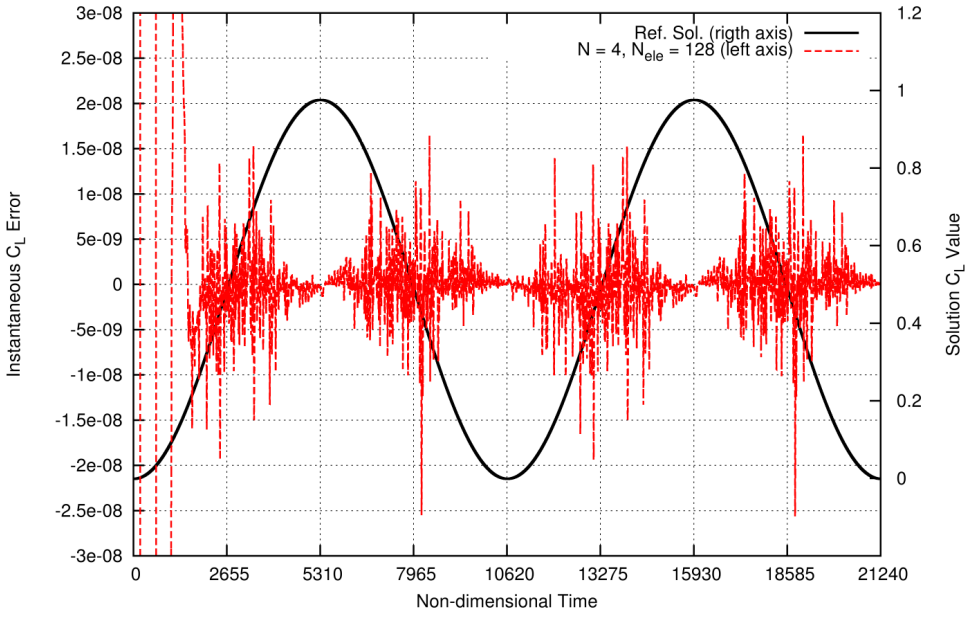
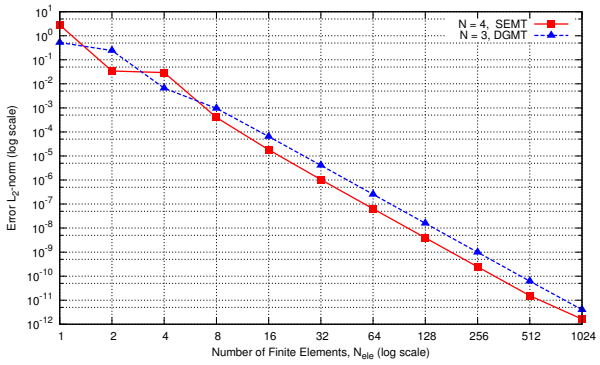
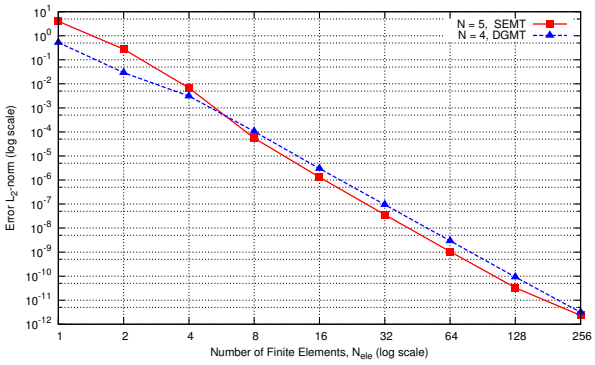


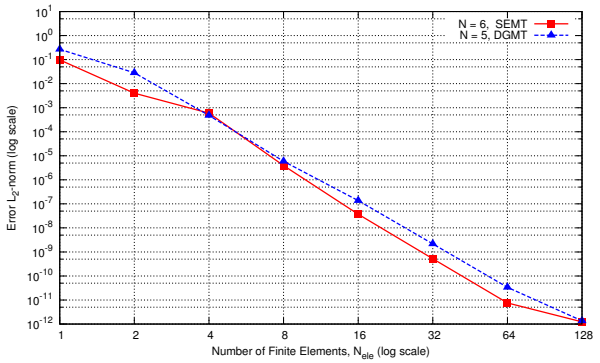
Figure 16. Instantaneous C_L error of the SEMT solutions versus non-dimensional time for the Euler equations with the airfoil motion prescribed in equation (37) and specified polynomial degree and number of spectral elements in time



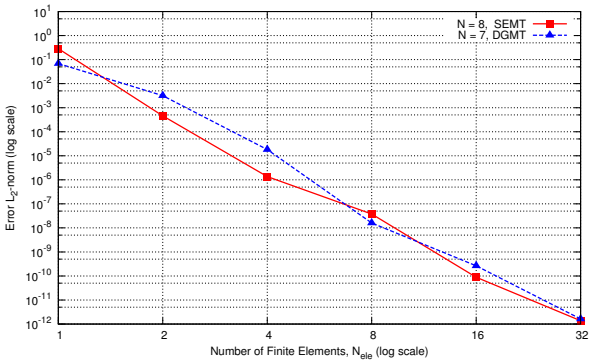
(a)



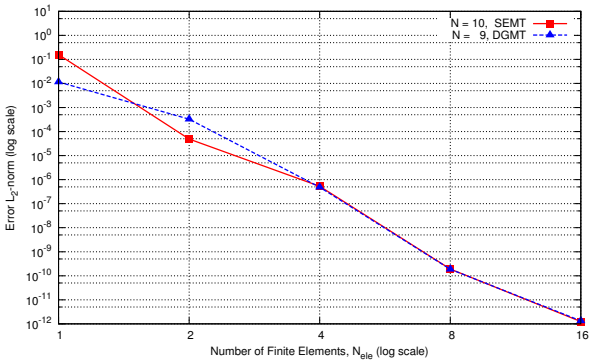
(b)



(c)

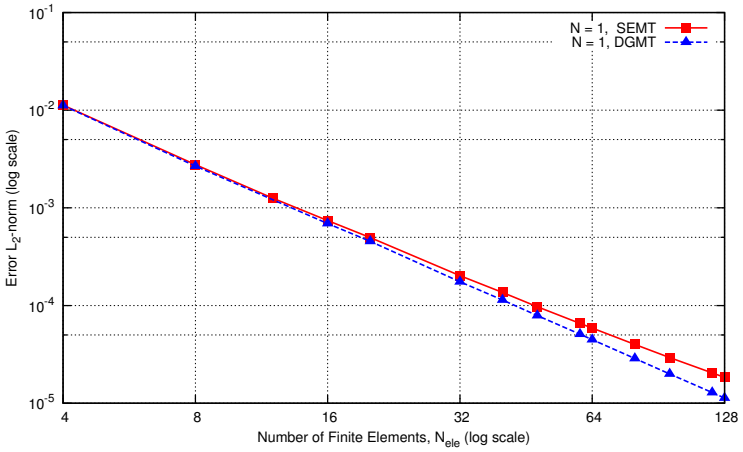


(d)

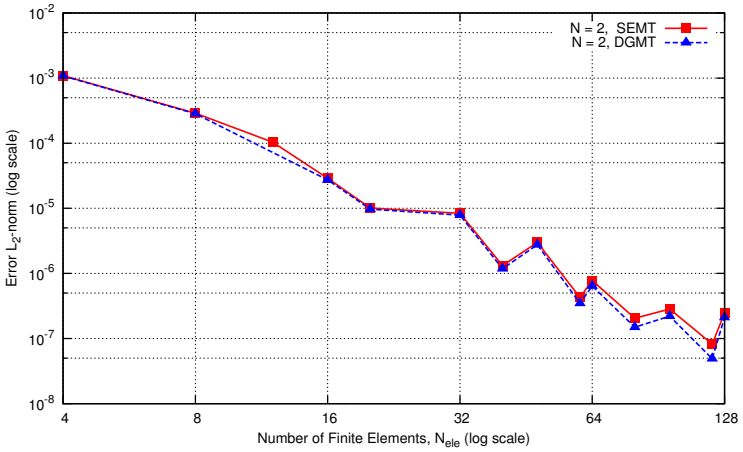


(e)

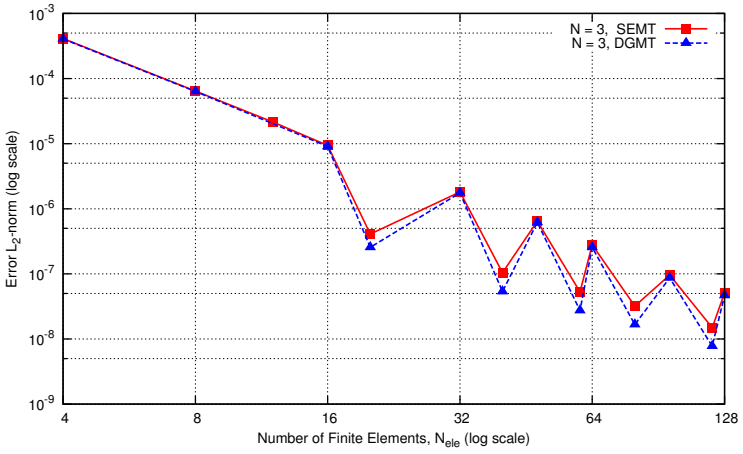
Figure 17. Comparison of the $L_2 - norm$ of the SEMT and DGMT solution error for the ODE in equation (32) with the given number of finite elements in time (x-axis) and the polynomial degrees specified in the legends



(a)

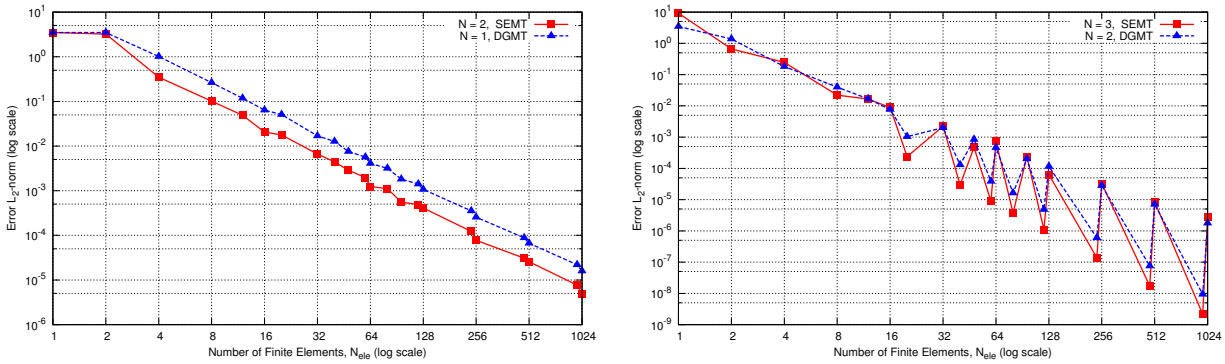


(b)



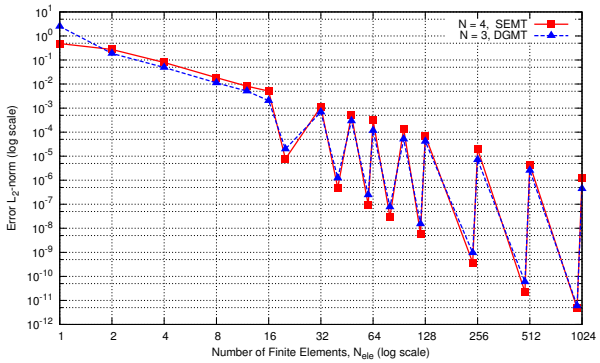
(c)

Figure 18. Comparison of the $L_2 - norm$ of the SEMT and DGMT C_L error for the airfoil motion given by equation (34) with the given number of finite elements in time (x-axis) and the polynomial degrees specified in the legends

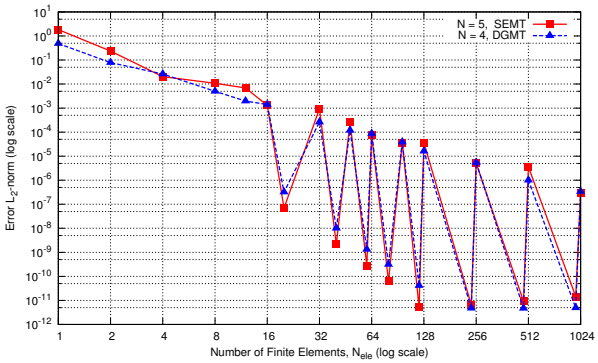


(a)

(b)



(c)



(d)

Figure 19. Comparison of the $L_2 - norm$ of the SEMT and DGMT solution error for the ODE in equation (35) with the given number of finite elements in time (x-axis) and the polynomial degrees specified in the legends

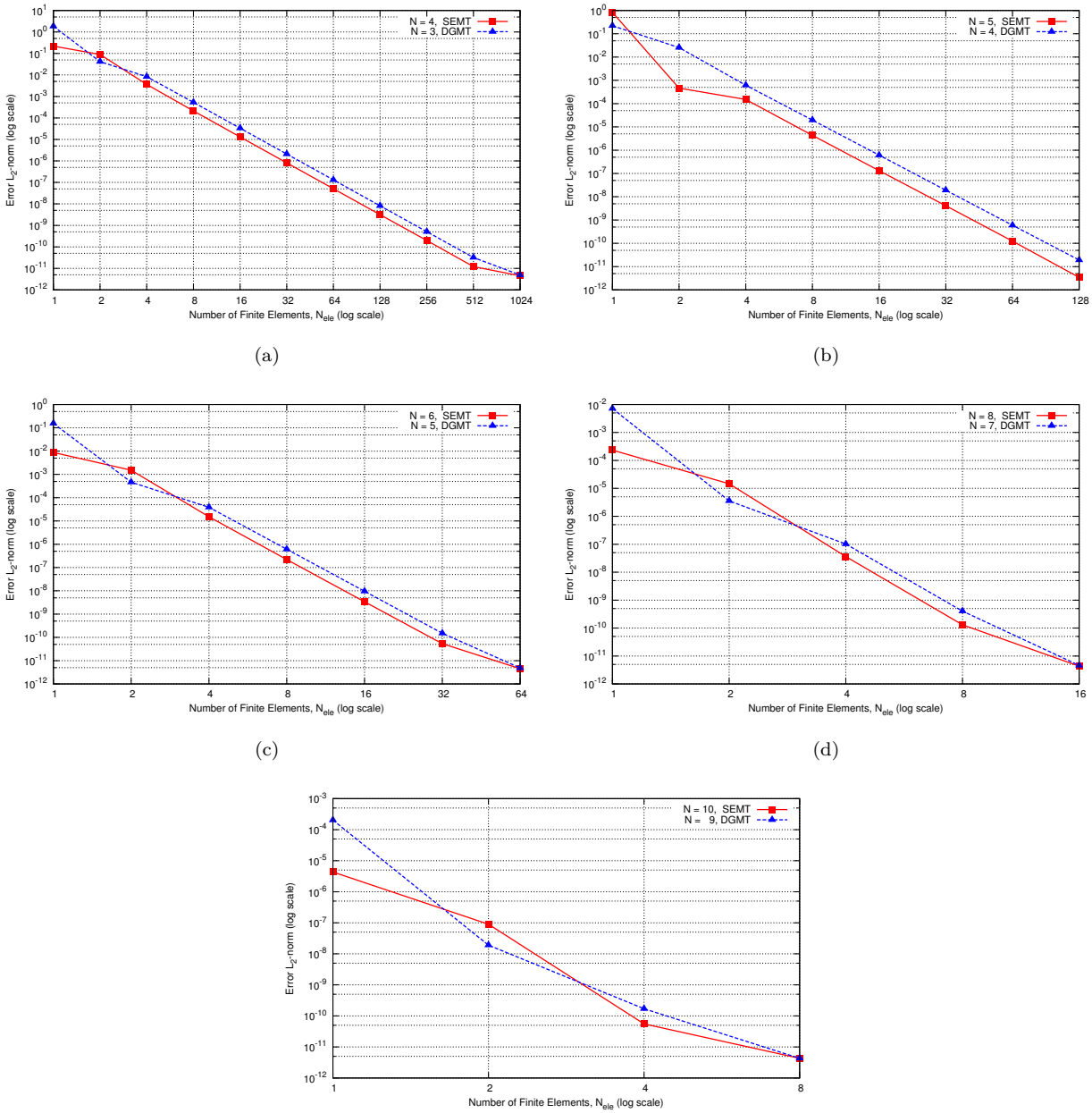


Figure 20. Comparison of the $L_2 - norm$ of the SEMT and DGMT solution error for the ODE in equation (38) with the given number of finite elements in time (x-axis) and the polynomial degrees specified in the legends

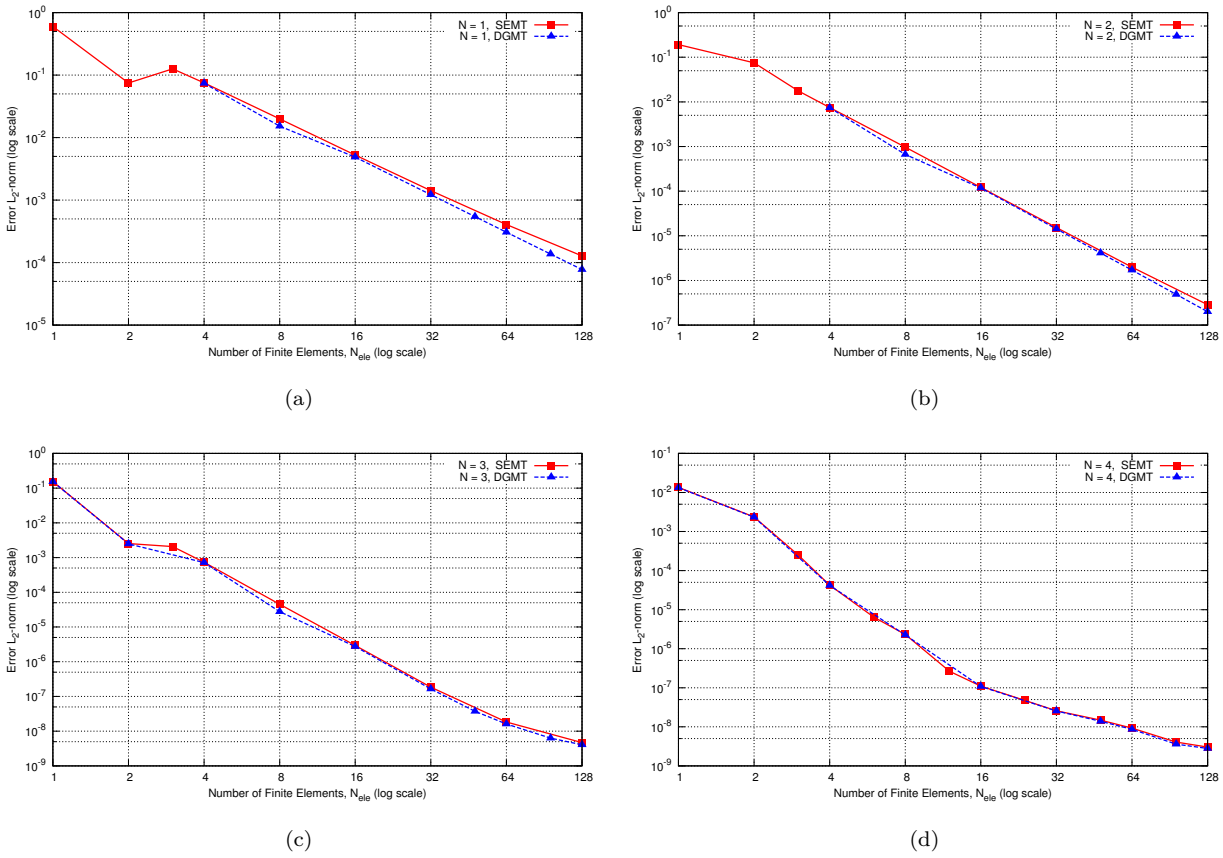


Figure 21. Comparison of the $L_2 - norm$ of the SEMT and DGMT C_L error for the airfoil motion given by equation (37) with the given number of finite elements in time (x-axis) and the polynomial degrees specified in the legends