Comparison of Finite Element Methods for Linearized Elastodynamics

Reza Abedi, Scott T. Miller

June 12, 2014

Abstract

We are providing the first quantitative comparison of numerical methods for elastodynamics, also referred to as *structural dynamics*. We shall only compare finite element methods.

1 Introduction

What are the popular/industry standard methods for solid mechanics and structural dynamics applications? How can we provide guidance to the novice user or entry-level engineer on which methods to choose?

TODO: Reza and Philip

2 Finite element formulations

1. Spacetime methods
   (a) SDG with causal meshing – Reza and Philip
   (b) SDG with timeslabs (later)
   (c) Time-discontinuous (later)
2. Method of lines: discretize space first
   (a) Continuous Galerkin (1-field & 2-field) – Scott

(b) Discontinuous Galerkin (multifield formulations) – Scott

(c) Lumped versus consistent mass matrix – Scott

(d) Temporal discretization

i. Explicit Runge-Kutta methods, traditional and SSP – Scott
ii. Implicit RK methods
iii. Implicit-explicit RK (IMEX) methods
iv. Newmark family of integrators (implicit)
v. Bathe integrator (implicit)
vi. Backward difference formulae (implicit)
vii. Others?

To be specific, we need to give the full finite element formulations, as well as quadrature schemes/rules used. Polynomial orders, etc. Optimal convergence rates.

3 One-field continuous Galerkin formulation

Spatial discretization via continuous finite elements yield the semi-discrete equation

$$M\ddot{U} + KU = F, \tag{1}$$

with the matrices given by

$$M_{ij} = \int_V \hat{u}_i \rho \hat{u}_j \, dV, \quad K_{ij} = \int_V \nabla \hat{u}_i \mathbb{C}[\mathrm{Sym}(\nabla \hat{u}_j)] \, dV, \quad F_i = \int_V \hat{u}_i \rho b_i \, dV \tag{2}$$

In all of the following time integration schemes, I shall attempt to write the equations so that we are solving for the displacements U, even though it may be more convenient to solve for the velocities or accelerations in some cases. The main differences present are the scalar coefficients and the presence of the mass matrix rather than the stiffness on the right-hand side of the equation.

3.1 Backward Euler

Backward Euler timestepping yields

$$\dot{U}^{n+1} = \frac{U^{n+1} - U^n}{\Delta t}, \quad \ddot{U} = \frac{U^{n+1}}{(\Delta t)^2} - \frac{U^n}{(\Delta t)^2} - \frac{\dot{U}^n}{\Delta t} \tag{3}$$

Substituting into (1), we have

$$\left(\frac{1}{(\Delta t)^2}M + K\right)U^{n+1} = F + M\left(\frac{U^n}{(\Delta t)^2} + \frac{\dot{U}^n}{\Delta t}\right). \tag{4}$$

3.2 Bathe's method

Bathe's method [1] is a two-stage, second order accurate implicit time integration method. The first step is of size $\Delta t/2$ using the trapezoidal rule. The second step is a backward difference formula (BDF2) scheme using the half-step and previous values. BDF schemes of order 2 or greater are linear multistep schemes. Note that this method is essentially leap-frogging applications of the trapezoidal and BDF2 schemes, each with half of the chosen time step size.

Applying the trapezoidal rule with step size $(\Delta t/2)$, the time derivatives are

$$\dot{U}^{n+1/2} = -\dot{U}^n + \frac{4}{\Delta t}\left(U^{n+1/2} - U^n\right), \quad \ddot{U}^{n+1/2} = -\ddot{U}^n + \frac{4}{\Delta t}\left(\dot{U}^{n+1/2} - \dot{U}^n\right). \tag{5}$$

The linear algebraic system generated from (1) and (5) is

$$\left(\frac{16}{(\Delta t)^2}M + K\right)U^{n+1/2} = F^{n+1/2} + M\left[\frac{16}{(\Delta t)^2}U^n + \frac{8}{\Delta t}\dot{U}^n + \ddot{U}^n\right] \tag{6}$$

The second step is to apply the BDF2 method at t^{n+1} using the $t^{n+1/2}$ and t^n steps as input to the 2-step method. The time derivatives are computed as

$$\dot{U}^{n+1} = \frac{1}{\Delta t}\left(3U^{n+1} - 4U^{n+1/2} + U^n\right), \quad \ddot{U}^{n+1} = \frac{1}{\Delta t}\left(3\dot{U}^{n+1} - 4\dot{U}^{n+1/2} + \dot{U}^n\right) \tag{7}$$

The resulting linear system for U^{n+1} is

$$\left(\frac{9}{(\Delta t)^2}M + K\right)U^{n+1} = F^{n+1} + M\left[\frac{12}{(\Delta t)^2}U^{n+1/2} - \frac{3}{(\Delta t)^2}U^n + \frac{4}{\Delta t}\dot{U}^{n+1/2} - \frac{1}{\Delta t}\dot{U}^n\right] \tag{8}$$

3.3 Newmark methods

The Newmark family of methods is a two parameter family of methods where

$$U^{n+1} = U^n + (\Delta t)\dot{U}^n + \frac{(\Delta t)^2}{2}\left\{(1-2\beta)\ddot{U}^n + 2\beta\ddot{U}^{n+1}\right\} \quad (9)$$

$$\dot{U}^{n+1} = \dot{U}^n + (\Delta t)\left\{(1-\gamma)\ddot{U}^n + \gamma\ddot{U}^{n+1}\right\} \quad (10)$$

The resulting linear system for U^{n+1} is then

$$\left(\frac{1}{\beta(\Delta t)^2}M + K\right)U^{n+1} = F^{n+1} + M\left(\frac{1}{\beta(\Delta t)^2}U^n + \frac{1}{\beta\Delta t}\dot{U}^n + \frac{1-2\beta}{2\beta}\ddot{U}^n\right) \quad (11)$$

From the solution to this equation, one can obtain the velocity and accelerations at t^{n+1} from the previous equations.

Note: I came across a paper on the web by Kane,Marsden,Ortiz and West that states Newmark methods are variational and thus symplectic and momentum preserving. They also have good energy behavior. Although it is a non-canonical symplectic form and a non-standard momenta.

3.3.1 Average acceleration

The average acceleration method, also known as the trapezoidal rule, is a member of the Newmark family of integration schemes. It is unconditionally stable and second order accurate [2]. The parameters $\beta = 1/4$ and $\gamma = 1/2$ in (9) to obtain the trapezoidal rule.

3.3.2 Linear acceleration

The trapezoidal rule, also known as the average acceleration method, is a member of the Newmark family of integration schemes. It is conditionally stable and second order accurate [2]. The parameters $\beta = 1/6$ and $\gamma = 1/2$ in (9) to obtain the linear acceleration method.

3.3.3 Central difference (explicit)

The explicit central difference formula is a member of the Newmark family. The choice $\beta = 0, \gamma = 1/2$ combined with mass-lumping yield a second order

accurate explicit time integration method:

$$
\begin{aligned}
U^{n+1} &= U^n + (\Delta t)\dot{U}^n + \frac{(\Delta t)^2}{2}\ddot{U}^n, && (12)\\
\ddot{U}^{n+1} &= (M_L)^{-1}(F^{n+1} - KU^{n+1}), && (13)\\
\dot{U}^{n+1} &= \dot{U}^n + \frac{\Delta t}{2}\left(\ddot{U}^{n+1} + \ddot{U}^n\right). && (14)
\end{aligned}
$$

Note: CDF is also known as explicit Newmark as well as velocity-Verlet.

3.4 Generalized α-methods

Generalized α-methods [3] use the Newmark time derivative approximations (9). However, two additional parameters $\{\alpha_m, \alpha_f\}$ are introduced as the governing equations are rewritten as

$$
M\left[(1-\alpha_m)\ddot{U}^{n+1} + \alpha_m\ddot{U}^n\right] + K\left[(1-\alpha_f)U^{n+1} + \alpha_f U^n\right] = (1-\alpha_f)F^{n+1} + \alpha_f F^n. \tag{15}
$$

Substituting the time derivative formulae from the Newmark methods into (15), we arrive at the following discrete equation:

$$
\begin{aligned}
\left[\frac{1-\alpha_m}{\beta(\Delta t)^2}M + (1-\alpha_f)K\right]U^{n+1} &= (1-\alpha_f)F^{n+1} + \alpha_f F^n - \alpha_f KU^n \\
&\quad + M\left[\frac{1-\alpha_m}{\beta(\Delta t)^2}U^n + \frac{1-\alpha_m}{\beta(\Delta t)}\dot{U}^n + \frac{1-\alpha_m-2\beta}{2\beta}\ddot{U}^n\right]. \quad (16)
\end{aligned}
$$

Second order accuracy and unconditional stability are obtained if the following are satisfied:

$$
\alpha_m < \alpha_f \leq \frac{1}{2}, \quad \gamma = \frac{1}{2} - \alpha_m + \alpha_f, \quad \beta_n \geq \frac{1}{4} + \frac{1}{2}(\alpha_f - \alpha_m). \tag{17}
$$

We can satisfy all of these by choosing all four parameters $\{\alpha_m, \alpha_f, \beta, \gamma\}$ based on a single parameters ρ_∞ as

$$
\begin{aligned}
\alpha_f &= \rho_\infty/(1+\rho_\infty), && (18)\\
\alpha_m &= (2\rho_\infty - 1)/(1+\rho_\infty), && (19)\\
\beta &= (1-\alpha_m+\alpha_f)^2/4, && (20)\\
\gamma &= 1/2 - \alpha_m + \alpha_f. && (21)
\end{aligned}
$$

3.4.1 Hilber-Hughes-Taylor α-method

HHT-α is obtained by setting $\alpha_m = 0$ in the generalized α-method. If the parameters are selected such that [2]

$$\alpha_f \in \left[0, \frac{1}{3}\right], \quad \gamma = \frac{1 + 2\alpha_f}{2}, \quad \beta = \frac{(1 + \alpha_f)^2}{4}, \tag{22}$$

an unconditionally stable, second-order accurate scheme results. With these choices, selecting $\alpha_f = 0$ results in the trapezoidal rule.

Note: Hughes has a "Comparison of Algorithms" section, but he looks at spectral radii of various methods with approximately the same computational cost.

Note 2: The α in [2] is actually $-\alpha_f$ here. STM has adjusted the current equations appropriately.

3.4.2 Wood-Bossak-Zienkiewicz α-method

WBZ-α is obtained by setting $\alpha_f = 0$ in the generalized α-method. If the parameters are selected such that [2]

$$\alpha_m \in \left[0, \frac{1}{3}\right], \quad \gamma = \frac{1 + 2\alpha_m}{2}, \quad \beta = \frac{(1 + \alpha_m)^2}{4}, \tag{23}$$

an unconditionally stable, second-order accurate scheme results.

3.5 Forward Euler

The simplest explicit method is the forward Euler method. We solve the equation

$$M\ddot{U}^{n+1} = F^n - KU^n, \tag{24}$$

with the time derivatives approximated as in (3). We shall also replace the consistent mass matrix given in (2) with a lumped mass matrix M_L given by (see, e.g., [2] or [4])

$$(M_L)_{ij} = \begin{cases} \sum_j \int_V \hat{u}_i \rho \hat{u}_j \, dV & i = j, \\ 0 & i \neq j. \end{cases} \tag{25}$$

That is, M_L is a diagonal matrix with positive non-zero entries that can be trivially inverted.

The forward Euler method updates the displacement as

$$U^{n+1} = (\Delta t)^2 (M_L)^{-1} K U^n + U^n + (\Delta t)\dot{U}^n. \tag{26}$$

3.6 Explicit generalized α-method

From [5]:

$$M[(1-\alpha_m)\ddot{U}^{n+1} + \alpha_m \ddot{U}^n] = F^n - KU^n, \tag{27}$$

$$U^{n+1} = U^n + (\Delta t)\dot{U}^n + \frac{(\Delta t)^2}{2}\left((1-2\beta)\ddot{U}^n + 2\beta\ddot{U}^{n+1}\right), \tag{28}$$

$$\dot{U}^{n+1} = \dot{U}^n + (\Delta t)((1-\gamma)\ddot{U}^n + \gamma\ddot{U}^{n+1}), \tag{29}$$

where the constants can be defined as a one-parameter family in terms of ρ_b as

$$\alpha_m = \frac{2\rho_b - 1}{1+\rho_b}, \quad \beta = \frac{5-3\rho_b}{(1+\rho_b)^2(2-\rho_b)}, \quad \gamma = \frac{3}{2} - \alpha_m. \tag{30}$$

4 Two-field continuous Galerkin formulation

Spatial discretization via continuous finite elements yield the semi-discrete equation

$$M\dot{U} + KU = F, \tag{31}$$

where

$$U = [U \ V]^{\mathrm{T}}, \tag{32}$$

and the matrices are given by

$$M = \begin{bmatrix} \int_V \hat{u}_i \hat{u}_j \, dV & 0 \\ 0 & \int_V \hat{v}_i \rho \hat{v}_j \, dV \end{bmatrix},$$

$$K = \begin{bmatrix} 0 & -\int_V \hat{u}_i \hat{v}_j \, dV \\ \int_V \nabla \hat{v}_i \mathbb{C}[\mathrm{Sym}(\nabla \hat{u}_j)] \, dV & 0 \end{bmatrix},$$

$$F = \begin{bmatrix} 0 \\ \int_V \hat{v}_i \rho b_i \, dV \end{bmatrix} \tag{33}$$

As in the one-field formulation, the lumped mass matrix M_L is obtained by summing the row entries and placing the result on the diagonal.

4.1 Backward Euler

Backward Euler timestepping yields

$$\dot{U}^{n+1} = \frac{U^{n+1} - U^n}{\Delta t}. \tag{34}$$

Substituting into (31), we have

$$\left(\frac{1}{(\Delta t)}M + K\right) U^{n+1} = F^{n+1} + M\left(\frac{U^n}{(\Delta t)}\right). \tag{35}$$

4.2 Forward Euler

The simplest explicit method is the forward Euler method. We solve the equation

$$M\dot{U}^{n+1} = F^n - KU^n, \tag{36}$$

with the time derivatives approximated as in (3). We shall also replace the consistent mass matrix given in (2) with a lumped mass matrix M_L whose off-diagonal entries are zero. That is, M_L is a diagonal matrix with positive non-zero entries that can be trivially inverted.

The forward Euler method explicitly updates the solution vector of displacements and velocities as

$$U^{n+1} = (\Delta t)(M_L)^{-1}(F^n - KU^n) + U^n \tag{37}$$

4.3 Symplectic Euler

Symplectic (or semi-implicit) Euler is a staggered/segregated solution algorithm where first the velocity and then the displacement are updated, sequentially.

Ignoring bad notation for a second, we can write the algorithm as:

1.
$$V^{n+1} = V^n + (\Delta t)(M_{22})_L{}^{-1}(F^n - K_{21}U^n), \tag{38}$$

2.
$$U^{n+1} = U^n + (\Delta t) - (M_{11})_L{}^{-1}K_{12}V^{n+1}. \tag{39}$$

5 Exact solutions

5.1 $C^\infty(\Omega)$ solution

The exact solutions for linearized elastodynamics take the form:

$$u_0 = A_0 \sin(m_0 x_0) \sin(\alpha t), \tag{40}$$

$$\begin{aligned} u_0 &= A_0 \sin(m_0 x_0) \sin(m_1 x_1) \sin(\alpha t), && \text{(41a)} \\ u_1 &= A_1 \cos(m_0 x_0) \cos(m_1 x_1) \sin(\alpha t), && \text{(41b)} \end{aligned}$$

$$\begin{aligned} u_0 &= A_0 \sin(m_0 x_0) \sin(m_1 x_1) \sin(m_2 x_2) \sin(\alpha t), && \text{(42a)} \\ u_1 &= A_1 \cos(m_0 x_0) \cos(m_1 x_1) \sin(m_2 x_2) \sin(\alpha t), && \text{(42b)} \\ u_2 &= A_2 \cos(m_0 x_0) \sin(m_1 x_1) \cos(m_2 x_2) \sin(\alpha t), && \text{(42c)} \end{aligned}$$

for $d = 1, 2, 3$, respectively. Let f_i and L_i be the frequencies and domain length in direction $i \in \{1, 2, 3\}$. The values m_i are obtained,

$$m_i = \frac{2\pi f_i}{L_i} \tag{43}$$

The eigen solutions, solutions with zero body force that satisfy the equations of motion, represent either a dilatational wave or a shear wave. The relation between A_i are as follow:

$$A_0 \neq 0 \tag{44}$$

$$\begin{aligned} A_0 = f_0, \quad A_1 = -f_1, \quad & \text{dilatational wave} && \text{(45a)} \\ A_0 = f_1, \quad A_1 = f_0, \quad & \text{shear wave} && \text{(45b)} \end{aligned}$$

$$\begin{aligned} A_0 = f_0, \quad A_1 = -f_1, \quad A_2 = -f_2, \quad & \text{dilatational wave} && \text{(46a)} \\ A_0 = f_1, \quad A_1 = f_0, \quad A_2 = 0, \quad & \text{shear wave (option 1)} && \text{(46b)} \\ A_0 = f_2, \quad A_1 = 0, \quad A_2 = f_0, \quad & \text{shear wave (option 2)} && \text{(46c)} \\ A_0 = f_0 f_2, \quad A_1 = -f_1 f_2, \quad A_2 = f_0^2 + f_1^2, \quad & \text{shear wave (option 3)} && \text{(46d)} \end{aligned}$$

for $d = 1, 2, 3$, respectively. The temporal coefficient α satisfies,

$$\alpha = \sqrt{\Sigma_{i=0}^{d} m_i^2} c \tag{47}$$

where $c = c_{\mathrm{d}}$ for dilatational wave and $c = c_{\mathrm{s}}$ for shear wave, respectively.

I am doing studies for only the dilatational wave for all dimensions. Thus the wave speed is given by (I use plane-strain model for $d = 2$),

$$c_{\mathrm{d}} = \begin{cases} 1 & d = 1 \\ \sqrt{\frac{1-\nu}{(1+\nu)(1-2\nu)} \frac{E}{\rho}} & d = 2, 3 \end{cases} \tag{48}$$

For the material properties I use ($E = \rho = 1, \nu = 0.3$), the wave speed is 1 and ≈ 1.16 for $d = 1$ and $d > 1$, respectively. In addition I use $L_i = 1$ for all dimensions. To ensure a full cycle in time the final time, T, would is equal to $2\pi/\alpha$. Based on the form of the eigen solutions for longitudinal wave we get $\alpha = 2\pi\sqrt{d}c$ which yields $T = \frac{1}{\sqrt{d}c}$.

The errors I propose to use are:

- Dissipation: It is our familiar integral of energy flux on the boundaries of the domain.
- L^2 and energy errors of the solution with respect to the exact solution. The energy errors are the integrals of internal and kinetic energy corresponding to the error in the displacement field.

Special care should be given to integrals on boundaries that correspond to initial and boundary conditions. For initial condition strain energy is zero and kinetic energy is given by,

$$K = \frac{1}{2} \int_{[0\ 1]^d} \rho |v|^2 \mathrm{d}A = \frac{\alpha^2 d}{2^{(d+1)}} = \frac{(\pi c d)^2}{2^{(d-1)}} \tag{49}$$

Computing the energy flux integral on the boundary of the domain is more tricky. First, some fluxes are prescribed and their energy adjoint quantities are obtained from the interior trace that in turn is computed from the discrete solution. In all my examples I have solved Dirichlet boundary condition (all velocities are determined based on the exact solutions given above). Accordingly, stress is obtained from FEM solution. Second, special care should be given to integration of the energy flux. In my setting, target velocity is

a trigonometric function while stress is a polynomial of order $p-1$ (p being the order of displacement interpolants). Regardless of the order of Gauss quadrature, there would be some error in computing these integrals in such boundaries. The final boundary of the domain that we should consider is the causal outflow boundary. As all target values are obtained from discrete solution, a quadrature order of $2(p-1)$ ensures exact evaluation of energy flux.

The function void SLSubConfig::setSpecifiedLoadFlags() can be used in the evaluation of equations 4 to 8 and the function bool SLPhysics::computeExactSolutionIndividua is used to compute the displacement field (equations 1 to 3) and the corresponding velocity, strain, linear momentum density, and stress fields.

5.2 $C^0(\Omega)$ solution with weak shocks

As suggested in [6], we can consider a one-dimensional bar impacting a wall.

Let $\Omega := \{x | x \in [0, L]\}$ with bar length $L = 4$, density $\rho = 1$, elastic modulus $\lambda = 1$. The time interval is $0 \leq t \leq 4$. Boundary conditions are

$$u(0,t) = t, \quad v(0,t) = 1, \quad u(4,t) = 0, \quad v(4,t) = 0. \tag{50}$$

The initial conditions are all zero.

The analytical solution is given as

$$u(x,t) = \begin{cases} t - x & \text{for } t \geq x, \\ 0 & \text{for } t \leq x. \end{cases} \tag{51}$$

Note: Idesman also has cases with weak shocks in higher dimensions. We should look at his paper and maybe plot some of the same stuff?

6 Conclusions

References

[1] Klaus-Jürgen Bathe. Conserving energy and momentum in nonlinear dynamics: a simple implicit time integration scheme. *Computers & Structures*, 85(7):437–445, 2007.

[2] Thomas JR Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Dover Publications, 2012.

[3] J Chung and GM Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method. *Journal of applied mechanics*, 60(2):371–375, 1993.

[4] RD Cook, DS Malkus, ME Plesha, and RJ Witt. *Concepts and Applications of Finite Element Analysis*. John Wiley & Sons, Inc, 4 edition, 2001.

[5] Gregory M. Hulbert and Jintai Chung. Explicit time integration algorithms for structural dynamics with optimal numerical dissipation. *Computer Methods in Applied Mechanics and Engineering*, 137(2):175 – 188, 1996.

[6] A. Idesman and D. Pham. Finite element modeling of linear elastodynamics problems with explicit time-integration methods and linear elements with the reduced dispersion error. *Computer Methods in Applied Mechanics and Engineering*, 271(0):86 – 108, 2014.