Some notes on complexity of structured and unstructured finite element methods for elastic wave propagation

Omar Ghattas and David O'Hallaron

1. Background

We are interested in numerically solving Navier's equations of elastodynamics

$$(\mu + \lambda)\nabla(\nabla \cdot \mathbf{u}) + \mu\nabla^2\mathbf{u} = \rho\frac{\partial^2\mathbf{u}}{\partial t^2}, \tag{1}$$

where $\mathbf{u}$ is the displacement vector field, ρ is the density, and λ and μ are elastic material constants. Discretization by finite elements produces a system of ordinary differential equations of the form

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}. \tag{2}$$

Here $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ is the damping matrix associated with the absorbing boundary and material damping, and $\mathbf{K}$ is the stiffness matrix. Given appropriate initial conditions, this system of ODEs can be integrated in time using central differences, yielding the explicit method

$$\left(\mathbf{M} + \frac{\Delta t}{2}\mathbf{C}\right)\mathbf{u}_{t+\Delta t} = \Delta t^2\mathbf{f}_t - \left(\Delta t^2\mathbf{K} - 2\mathbf{M}\right)\mathbf{u}_t - \left(\mathbf{M} - \frac{\Delta t}{2}\mathbf{C}\right)\mathbf{u}_{t-\Delta t}. \tag{3}$$

Since $\mathbf{M}$ and $\mathbf{C}$ are diagonal matrices, inversion of the time stepping operator requires only a scaling of the righthand side of (3), which is carried out just once prior to time stepping. Thus, we avoid solving linear systems at each time step, as required by an implicit method. On the other hand, the explicit method is only conditionally stable, so we can expect to have to take a larger number of time steps.

2. Storage complexity

In this section we characterize the storage required by structured and unstructured mesh methods, in terms of the number of nodes N_n and the average nodal adjacency N_a (i.e. the number of nodes that are coupled to a typical node). Assumptions are that

- $\mathbf{M}$ and $\mathbf{C}$ are diagonal,
- $O(1)$ storage is ignored,
- Δt is constant,
- $N_n \gg 1$, and

- symmetry is exploited in matrix storage.

Storage requirements for the different components of each method are summarized in Table 1.

Table 1: Words of storage required for structured and unstructured mesh methods

item	structured	unstructured
$\mathbf{f}$	$3N_n$	$3N_n$
$\mathbf{u}_t$	$3N_n$	$3N_n$
$\mathbf{u}_{t-\Delta t}$	$3N_n$	$3N_n$
$\mathbf{M} + \frac{\Delta t}{2}\mathbf{C}$	$3N_n$	$3N_n$
$\mathbf{M} - \frac{\Delta t}{2}\mathbf{C}$	$3N_n$	$3N_n$
$\Delta t^2\mathbf{K} - 2\mathbf{M}$	–	$4.5N_aN_n$
material type	N_n	–
adjacency info.	–	N_aN_n
total	$18N_n$	$5.5N_aN_n + 15N_n$

So we see that, for the same number of nodes, storage for an unstructured mesh method can be much greater than that of the structured method. However, we expect that an unstructured mesh will contain many fewer nodes, because the mesh size can be tailored to the scale of the local physics.

3. Arithmetic complexity

In this section we characterize the number of operations per time step required by both structured and unstructured mesh methods to integrate the equations of elastodynamics (2) by the explicit method (3). Our parameters again are N_n and N_a. Additional assumptions include

- the cost of setting up matrices and vectors in (3) is negligible, and
- the cost of evaluating $\mathbf{f}_t$ at each time step is negligible.

The results are summarized in Table 2. The explicit algorithm (3) thus requires two vector multiplications, one sparse matrix-vector multiplication, and two vector additions. Both structured and unstructured methods require the same work, so no distinction in made in table between them.

Table 2: Number of floating point operations per explicit time step

item	quantity	total flops
vector multiply	two	$6N_n$
matrix-vector multiply	one	$18N_aN_n$
vector add	two	$6N_n$
total		$18N_aN_n + 12N_n$

4. Communication complexity

Because of the large values of N_n needed for seismic wave propagation in large basins, parallel computers are indicated. In this section we characterize the communication complexity in terms of the number of nodes N_n and the number of processors P. We make two basic assumptions in addition to those made already:

- the number of nodes per processor, N_n/P, is constant, and
- the mesh partitions are cube-shaped.

The former is not difficult to achieve using contemporary mesh partitioners. On the other hand, the latter is a crude but currently necessary assumption for unstructured meshes. Once 3D unstructured meshes of portions of our target basin—Los Angeles—become available, we can use actual numbers corresponding to specific basins.

Communication is necessary because the displacement vector $\mathbf{u}$ as well as entries of the stiffness matrix $\mathbf{K}$ and the diagonal damping and mass matrices $\mathbf{C}$ and $\mathbf{M}$ are distributed. Under our assumptions, each cubic partition has $(N_n/P)^{\frac{1}{3}} \times (N_n/P)^{\frac{1}{3}} \times (N_n/P)^{\frac{1}{3}}$ nodes. Boundary nodes are shared among processors, and communication is required between processors for each shared a node.

There are three types of boundary nodes: those that lie on one of the six faces, those that lie on one of the twelve edges, and those that lie at the eight vertices. Face nodes are shared by two processors, edge nodes by four, and vertex nodes by eight. For each shared boundary node, a processor must both send and receive three displacements to every other processor that shares that node. Thus, if we ignore vertex nodes (of which there are very few when N_n is large), the total communication per processor per time step is

$$216 \left(\frac{N_n}{P}\right)^{\frac{1}{3}} + 36 \left(\frac{N_n}{P}\right)^{\frac{2}{3}} \text{ words.}$$

Since the amount of computation per processor per time step is $(18N_nN_a + 12N_n)/P$, the computation to communication ratio is

$$\frac{(1.5N_a + 1)\left(\frac{N_n}{P}\right)^{\frac{2}{3}}}{18 + 3\left(\frac{N_n}{P}\right)^{\frac{1}{3}}}.$$

5. Some common finite elements

In this section we will specialize the storage and work estimates to some common finite elements. In addition to the assumptions above, we use the idea that

- the average adjacency for an unstructured mesh is the same as that of a structured mesh,

to help us find N_a for a given element type. The results are summarized in Table 3 below.

Table 3: Storage and work estimates for some common elements.

element	N_a	struc storage	unstruc storage	flops/time step
linear tetrahedron	15	$18N_n$	$97.5N_n$	$282N_n$
trilinear hexahedron	27	$18N_n$	$163.5N_n$	$498N_n$
triquadratic hexahedron	64	$18N_n$	$367N_n$	$1164N_n$

6. Costs of realistic problems

In this final section we relate the storage, work, and communication estimates of the previous sections to our scientific goal, the prediction of earthquake-induced ground motion in large basins, in particular the Greater Los Angeles Basin. A knowledge of ground motion is essential to analyzing the safety of existing buildings as well as designing new ones.

In wave propagation problems, the node spacing is given by

$$h = \frac{v_s T}{N_\lambda},$$

where v_s is the shear wave velocity that characterizes a material, T is the period of excitation, and N_λ is the number of nodal points needed to resolve a wavelength, usually between 4 and 20, depending on the order of the element. So if the domain were homogeneous and shaped like a cube of length L, a total of $(L/h)^3$ nodes would be needed. For 3D wave propagation problems, the total number of nodes thus varies as the inverse of the cube of shear wave velocity and the excitation period. The cubic dependence is a result of the dimensionality

of the problem. Halving the period of incoming waves or the shear wave velocity of the soil results in an eightfold increase in problem size N_n. Since storage and number of flops per time step depend linearly on N_n, we can say that

$$\text{storage or flops/time step} = O(v_s^{-3}T^{-3}).$$

Thus the size of a problem is completely specified by the parameters v_s and T, the dimensions of the basin and surrounding rock, and the element order. Most basins are highly heterogeneous, containing many layers of differing material. Unstructured meshes allow us to adapt the mesh to the wavelength in each layer. So we estimate the required number of nodes layer by layer and sum. The Greater LA Basin contains soil deposits in the upper layers that have shear wave velocities as low as 200m/s. Furthermore, excitations with periods as low as 0.5s have to be resolved. Using these parameters, we arrive at a value of N_n on the order of 10^8. Thus, assuming 8 byte words and linear tetrahedra, we need about 100Gb of primary storage in order to predict ground motion in the LA Basin for realistic parameters. This is beyond the capacity of existing machines. Thus we are memory bound; this is typical of unstructured domain methods such as finite elements.

Due to these memory limitations, present simulations are limited to faster shear wave velocities and longer periods than those needed for a complete analysis. As bigger machines become available, T and v_s are reduced towards their target values. We see, therefore, that for the foreseeable future, our simulation will fill the entire available memory of any given machine. We can use this fact to simplify the complexity expressions (let us assume linear tetrahedra). Let M represent the available memory per processor in words. Thus the total available memory is MP words, and the total number of nodes N_n is $MP/97.5$. We can restate the complexities in terms of M:

- computation per processor per time step = $2.9M$ flops
- communication per processor per time step = $1.7M^{\frac{2}{3}} + 47M^{\frac{1}{3}}$ words
- computation to communication ratio = $\frac{1.1M^{\frac{2}{3}}}{18+0.65M^{\frac{1}{3}}}$ flops/words

The computation to communication ratio is approximately equal to $1.7M^{\frac{1}{3}}$ for large enough M, say 8 megawords.

Finally, we note that the required time steps number in the thousands to tens of thousands for realistic earthquakes. Since multiple earthquake scenarios have to be considered, it is desirable that each simulation be completed in less than an hour.

7. Validating the estimates

We can validate the communication estimates with several moderately-sized unstructured 3D meshes we have generated. They correspond to coarse and fine meshes for a half-moon shaped

basin. The coarse mesh has 21,062 elements and 4220 nodes; the fine mesh has 117,044 elements and 22,119 nodes. The partitioning of the coarse mesh results in a maximum communication per processor of 1446 words per time step. The partitioned fine mesh results in a maximum communication per processor of 2736 words per time step. Our model predicts communication of 1460 (fine) and 3290 (coarse) words per processor per time step, which is a reasonable prediction considering the the structured mesh assumed in the model. Also these meshes have $N_a = 20.3$ (coarse) and $N_a = 17.5$ (fine) compared to the value of 15 characteristic of a structured mesh.