(1.) We start in fe_mainExplicit. We need to calculate critical time step dT. To do this we use fe_getTimeStep to find deltaT_crit.

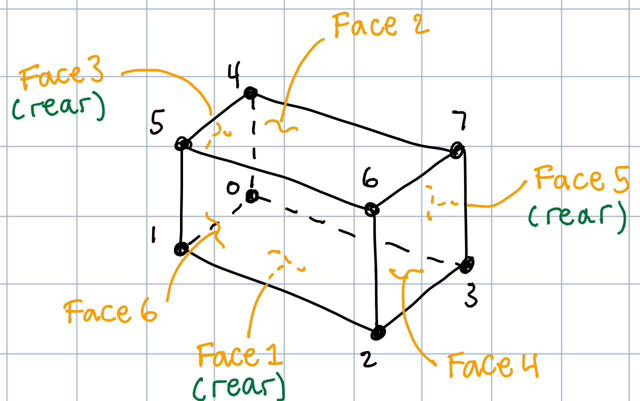
(2.) We start by extracting nodes and elements from mesh[0]. In addition, we also extract the system-level displacement vector U, which was recently read into mesh[0]. Finally, we use the method getElementCharacteristicPointer from the Mesh class. This method returns element_characteristic_pointer which is defined as &element_characteristic in the other method preprocessMesh. The method preprocessMesh is called at the end of fe_mainRead in a loop over all mesh objects, within preprocessMesh another method, calculateElementCharacteristic, is called, which returns a nel x 1 vector element_characteristic for each mesh object. Each component of element_characteristic is the volume of the corresponding element.

Note:
In conclusion, element_volumes in (fe_getTimeStep) is a pointer to a vector of element initial volumes.

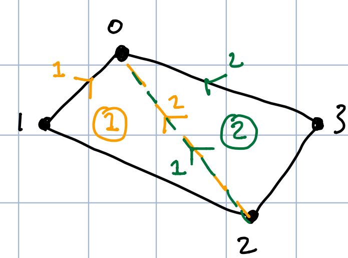
(3.) Next, we run a loop over nel. First, we extract the x, y, z coordinates for the element from nodes. Then we use the system-level displacement vector U and the element connectivity from elements to generate the element-level 24x1 vector u_e using fe_gather_pbr.

④ While still in the loop over nel we use fe_calTimeStep to populate the nel x 1 vector deltaT_element. In fe_calTimeStep, we start off by using u_e to calcultate the current x,y,z coordinates of the element nodes. Then we use fe_calVolume to calculate volume_current. Next, we use fe_minElementLength to calculate the critical length lc.

⑤ In fe_minElementLength we first calculate the area of each face on the current hex element using fe_calArea_4. The area values are stored in 6 x 1 vector face_area. Next, fe_calVolume is used to

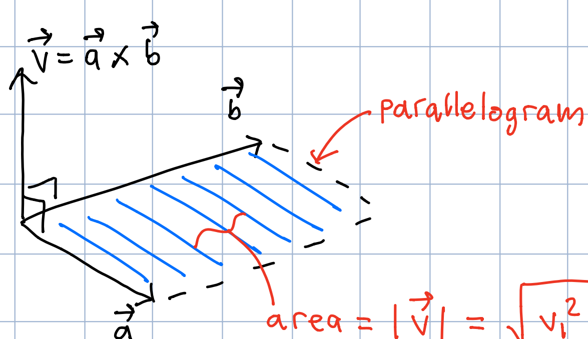


Hexahedral Element



Face 1

Pg. 369



area $= |\vec{V}| = \sqrt{V_1^2 + V_2^2 + V_3^2}$

Note:
Area of triangle is half of Parallelogram.

calculate the current volume_element. Next, we extract largest_face from face_area. Finally, we calculate ℓc.

Pg. 25-1 (LS-DYNA)

$$\ell c = \frac{\text{volument\_element}}{\text{largest\_face}}$$

volume

area

(6.) The next step in fe_calTimeStep is to calculate c_wave using fe_calWaveSpeed. First, we use fe_get_model to extract the material name mechanical_mat_model per the material_id associated with the particular element. Then we extract material properties E, nu, rho for model = simple_elastic and K, rho for model = mooney-rivlin_hyperelastic.

Pg. 335

stable time step

reduction factor (courant number)

$$\Delta t = \alpha \, \Delta t_{crit}$$

critical time step

Restrictions:

- constant strain elements
- rate-independent materials

characteristic length of the element

$$\Delta t_{crit} = \frac{2}{\omega_{max}} < \min_{e, I} \frac{2}{\omega_I^e} = \min_{e} \frac{\ell_e}{c_e}$$

(courant condition)

current wave speed in the element

max frequency of the linearized system (highest eigenvalue)

Pg. 74 (2-node element with diagonal mass)

initial length

For small deformation, E^{PF} corresponds to Young's modulus.

$$\Delta t_{crit} = \frac{l_0}{C_0} \qquad \text{where} \qquad c_0^2 = \frac{E^{PF}}{\rho_0}$$

wave speed

Note:

Large density and small stiffness are good.

Pg. 25 (constitutive equations)

nominal stress tensor

$$P(X,t) = E^{PF}\varepsilon(X,t) = E^{PF}(F(X,t)-1)$$ linear elastic material

functional relating P and F

Pg. 25-1 (LS-DYNA)

$$c = \sqrt{\frac{E(1-\nu)}{(1+\nu)(1-2\nu)\rho}}$$ elastic materials with a bulk modulus

Per code: (no source)

Bulk Modulus

$$c\_wave = \sqrt{\frac{volume\_current}{volume\_initial}} \times \sqrt{\frac{K}{rho}} \times 1.3$$

Pg. 206

$$K = B = \frac{E}{3(1-2\nu)}$$ (bulk modulus)

$$G = \frac{E}{2(1+\nu)}$$ (shear modulus)

Note:

- small-strain theory
- linear elastic material

7. The last step in fe_calTimeStep is $deltaT = \frac{lc}{c\_wave}$. Then, back in fe_getTimeStep, we find deltaT_crit; which is the smallest value in the nel x 1 vector deltaT_element. Next, we update deltaT_crit by multiplying it by reduction, which is a global variable.

8. Finally, we compare deltaT_crit to failure_time_step, which is defined in fe_mainEXPLICIT, in order to make sure the new time step is not too small. Lastly, we compare deltaT_critical to dt_min, in order to make sure the new time step is not too big. Note that dt_min is a global variable and is defined in fe_mainRead from the input file.