# A non-linear anisotropic Finite Element Method for surgical simulation

Stephen J. Jeffrey

*Advanced Computational Modelling Centre*

*University of Queensland, Brisbane, Queensland 4072, Australia*

This report describes a real-time finite element method (FEM) for simulating the deformation of soft tissue. The algorithm is non-linear, anisotropic and invariant to rigid rotations and translations.

## 1 Introduction to elasticiy modeling with FEM

A high-fidelity surgical simulator must accurately reproduce the movement and deformation of tissue and organs, typically in response to user manipulation. The physical response of biological tissue can be modeled using linear elasticity providing the deformation remains relatively small [1]. In the following subsections we outline an elasticity model developed by Picinbono and coworkers, which they presented in a series of papers [2], [3] and [4].

### 1.1 Linear Elasticity and an FEM implementation

The deformation of an organ can be represented by the action of a deformation field $\mathbf{\Phi}$ on the model, $\mathcal{M}$. If the deformation results in some or all components of the model being displaced by an amount $\mathbf{U}(x, y, z)$, the resulting model is given by: $\mathcal{M}_{deformed} = \mathcal{M} + \mathbf{U}(x, y, z)$. The elastic energy arising from the deformation can be quantified by the Cauchy-Green deformation tensor:

$$C = \nabla\mathbf{\Phi}^T \nabla\mathbf{\Phi} \tag{1}$$

or the Green-St. Venant deformation tensor:

$$E = \frac{1}{2}(C - I) = \frac{1}{2}\left(\nabla\mathbf{U}^T + \nabla\mathbf{U} + \nabla\mathbf{U}^T\nabla\mathbf{U}\right), \tag{2}$$

where $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)$ is the gradient operator, $I$ is the $3 \times 3$ identity matrix and $\nabla\mathbf{\Phi} = I + \nabla\mathbf{U}$. If we consider only the linear components of $E$:

$$E_l = \frac{1}{2}\left(\nabla\mathbf{U}^T + \nabla\mathbf{U}\right), \tag{3}$$

the elastic energy is given by [2]:

$$W_{linear} = \frac{\lambda}{2}(\text{tr } E_l)^2 + \mu \text{tr } E_l^2 = \frac{\lambda}{2}(\text{div}\mathbf{U})^2 + \mu \|\nabla \mathbf{U}\|^2 - \frac{\mu}{2}\|\text{rot}\mathbf{U}\|^2, \tag{4}$$

where $\lambda$ and $\mu$ are the Lamé coefficients, tr $E_l$ is the trace of the $3 \times 3$ linearised deformation tensor, div$\mathbf{U} = \nabla.\mathbf{U}$ is the divergence of $\mathbf{U}$ and rot$\mathbf{U} = \nabla \times \mathbf{U}$ is the rotation of $\mathbf{U}$.

To compute the forces arising from the deformation using the finite element method, we subdivide the volume comprising $\mathcal{M}$ into a conformal tetrahedral mesh. Each tetrahedron $\mathcal{T}_i$ is defined by 4 nodes $\mathbf{P}_j, j = 0, \ldots, 3$ and the interpolation functions are linear:

$$\begin{aligned}\boldsymbol{\alpha}_j = \frac{(-1)^j}{6V(\mathcal{T}_i)}\big(\mathbf{P}_{(j+1)mod4} \times \mathbf{P}_{(j+2)mod4} &+ \mathbf{P}_{(j+2)mod4} \times \mathbf{P}_{(j+3)mod4} \\ &+ \mathbf{P}_{(j+3)mod4} \times \mathbf{P}_{(j+1)mod4}\big),\end{aligned} \tag{5}$$

where *mod* denotes modulo and $V(\mathcal{T}_i)$ is the volume of tetrahedron $\mathcal{T}_i$. The interpolation functions allow one to calculate the value of some property, given the corresponding values at the nodal positions. For example, the displacement $\mathbf{U}(\mathbf{X})$ of point $\mathbf{X}$ is given by:

$$\mathbf{U}(\mathbf{X}) = \sum_{j=0}^{3} \Lambda_j(\mathbf{X})\mathbf{U}_j \tag{6}$$

where $\Lambda_j(\mathbf{X}), j = 0, \ldots, 3$ are the barycentric coordinates of $\mathbf{X}$, $\Lambda_j(\mathbf{X}) = \boldsymbol{\alpha}_j.\mathbf{X} + \beta_j$. ($B_j$ is a constant, see [5].) When the above expression for $\mathbf{U}$ is substituted into Eqn 4 the linear elastic energy is given by:

$$W_{linear}(\mathcal{T}_i) = \sum_{j,k=0}^{3} \mathbf{U}_j^T \left[\mathcal{B}_{jk}^{\mathcal{T}_i}\right] \mathbf{U}_k, \tag{7}$$

where

$$\mathcal{B}_{jk}^{\mathcal{T}_i} = \frac{\lambda}{2}(\boldsymbol{\alpha}_j \otimes \boldsymbol{\alpha}_k) + \frac{\mu}{2}\left[(\boldsymbol{\alpha}_k \otimes \boldsymbol{\alpha}_j) + \boldsymbol{\alpha}_j.\boldsymbol{\alpha}_k I\right]. \tag{8}$$

$\mathcal{B}_{jk}^{\mathcal{T}_i}$ is a $3 \times 3$ matrix that encapsulates the contribution from tetrahedron $\mathcal{T}_i$ to the stiffness tensor for the edge between nodes $P_j$ and $P_k$, or the contribution from node $P_j$ if $j = k$.

The force exerted on node $p$ is obtained by differentiating the elastic energy with respect to nodal position:

$$\mathbf{F}_p^{linear}(\mathcal{T}_i) = \frac{\partial W_{linear}}{\partial \mathbf{P}_p} = 2\sum_{j=0}^{3} \left[\mathcal{B}_{pj}^{\mathcal{T}_i}\right] \mathbf{U}_j. \tag{9}$$

The total force experienced by node $p$ is given by summing the contributions from all tetrahedra that share that node. The movement of the simulated organ is given by numerically propagating the position of each node in the model $\mathcal{M}$. A Newtonian integration scheme is used [6]:

$$m_i \frac{d^2\mathbf{P}_i}{dt^2} = \mathbf{F}_i - \gamma_i \frac{d\mathbf{P}_i}{dt} \tag{10}$$

where $m_i$ is the mass associated with node $i$ and $\gamma$ is the damping coefficient. Given the position of each node at times $t$ and $t-1$, the position at time $t+1$ can be computed using an explicit integration scheme:

$$\left(\frac{2m_i}{\eta(\eta+1)\Delta t^2} + \frac{\gamma_i}{\eta(\eta+1)\Delta t}\right)\mathbf{P}_i^{t+1} = \mathbf{F}_i + \left(\frac{2m_i}{\eta\Delta t^2} - \frac{\gamma_i(\eta-1)}{\eta\Delta t}\right)\mathbf{P}_i^{t} - \left(\frac{2m_i}{(\eta+1)\Delta t^2} - \frac{\gamma_i\eta}{(\eta+1)\Delta t}\right)\mathbf{P}_i^{t-1}, \tag{11}$$

where $\mathbf{P}^{t-1}$, $\mathbf{P}^{t}$ and $\mathbf{P}^{t+1}$ are the positions of the node at times $t-1$, $t$ and $t+1$, respectively. $\Delta t$ is the timestep between times $t-1$ and $t$ and $\eta\Delta t$ is the timestep between times $t$ and $t+1$ (See Appendix A). The preceding development (with the exception of Eqns 10 and 11) was presented by Cotin et al in [2].

## 1.2 Non-linear elasticity

The linear elasticity algorithm just presented can be used to model small deformations in biomechanical materials. As the magnitude of deformations increase, the linearised Green-St. Venant deformation tensor progressively becomes less accurate due to the omission of the quadratic terms (cf. Eqns 2 and 3). If the quadratic terms are retained ie. the Green-St. Venant deformation tensor is used, the expression for the elastic energy becomes [5]:

$$\begin{aligned} W = {} & \tfrac{\lambda}{2} \left[(\mathrm{div}\mathbf{U}) + \frac{1}{2}\left\|\nabla\mathbf{U}\right\|^2\right]^2 + \mu\left\|\nabla\mathbf{U}\right\|^2 - \frac{\mu}{2}\left\|\mathrm{rot}\mathbf{U}\right\|^2 \\ + {} & \mu\left(\nabla\mathbf{U} : \nabla\mathbf{U}^T\nabla\mathbf{U}\right) + \frac{\mu}{4}\left\|\nabla\mathbf{U}^T\nabla\mathbf{U}\right\|^2, \end{aligned} \tag{12}$$

where $A\!:\!B$ denotes the scalar product of two matrices $A$ and $B$, defined as: $A\!:\!B = \sum_{ij} A_{ij}B_{ij}$. To compute the elastic force using the finite element method, we convert the above expression to tensor representation, proceeding as we did for the linearised case. The result is:

$$\begin{aligned} \mathbf{F}_p(\mathcal{T}_i) = {} & 2\sum_{j=0}^{3}\left[\mathcal{B}_{pj}^{\mathcal{T}_i}\right]\mathbf{U}_j + \sum_{j,k=0}^{3} 2\left(\mathbf{U}_k \otimes \mathbf{U}_j\right)\mathcal{C}_{jkp}^{\mathcal{T}_i} + \left(\mathbf{U}_j.\mathbf{U}_k\right)\mathcal{C}_{pjk}^{\mathcal{T}_i} \\ + {} & 4\sum_{j,k,l=0}^{3}\mathcal{D}_{jklp}^{\mathcal{T}_i}\mathbf{U}_l\mathbf{U}_k^T\mathbf{U}_j \end{aligned} \tag{13}$$

where the $\mathcal{C}_{jkl}$ terms are vectors:

$$\mathcal{C}^{\mathcal{T}_i}_{jkl} = \frac{\lambda}{2}\boldsymbol{\alpha}_j\left(\boldsymbol{\alpha}_k.\boldsymbol{\alpha}_l\right) + \frac{\mu}{2}\left[\boldsymbol{\alpha}_l\left(\boldsymbol{\alpha}_j.\boldsymbol{\alpha}_k\right) + \boldsymbol{\alpha}_k\left(\boldsymbol{\alpha}_j.\boldsymbol{\alpha}_l\right)\right], j,k,l = 0,\ldots,3 \tag{14}$$

and the $\mathcal{D}_{jklm}$ terms are scalars:

$$\mathcal{D}^{\mathcal{T}_i}_{jklm} = \frac{\lambda}{8}\left(\boldsymbol{\alpha}_j.\boldsymbol{\alpha}_k\right)\left(\boldsymbol{\alpha}_l.\boldsymbol{\alpha}_m\right) + \frac{\mu}{4}\left(\boldsymbol{\alpha}_j.\boldsymbol{\alpha}_m\right)\left(\boldsymbol{\alpha}_k.\boldsymbol{\alpha}_l\right), j,k,l,m = 0,\ldots,3 \tag{15}$$

The above expression for the non-linear elastic force was presented by Picinbono et al in [4].

## 1.3 Non-linear anisotropic elasticity

Many biomechanical materials are not isotropic. Indeed many tissues, such as ligaments, exhibit a strong anisotropy and consequently the isotropic models of the preceding sections are not valid. Picinbono et al developed an expression for the elastic energy arising from deformation of materials that are transversally isotropic [4]. The material is parameterised by Lamé coefficients $\lambda^L$ and $\mu^L$, which characterise the biomechanical behaviour in the direction of the unit vector $\mathbf{a}_0$. In the plane orthogonal to $\mathbf{a}_0$, the material is characterised by Lamé coefficients $\lambda$ and $\mu$. Shearing and cross-stretching terms were neglected as they were considered to be small relative to the stretching terms. The resulting expression for the non-linear, transversally isotropic force [4] is the same as Eqn 13, but the stiffness parameters $\mathcal{B}_{jk}$, $\mathcal{C}_{jkl}$ and $\mathcal{D}_{jklm}$ now have both isotropic and transversally isotropic components. $\mathcal{B}_{jk}$ is a $3 \times 3$ symmetric matrix:

$$\begin{aligned}\mathcal{B}^{\mathcal{T}_i}_{jk} = & \tfrac{\lambda}{2} \quad \left(\boldsymbol{\alpha}_j \otimes \boldsymbol{\alpha}_k\right) + \frac{\mu}{2}\left[\left(\boldsymbol{\alpha}_k \otimes \boldsymbol{\alpha}_j\right) + \boldsymbol{\alpha}_j.\boldsymbol{\alpha}_k I\right] \\ + & \left(\frac{\lambda^L - \lambda}{2} + \mu^L - \mu\right)\left(\mathbf{a}_0 \otimes \mathbf{a}_0\right)\left(\boldsymbol{\alpha}_j \otimes \boldsymbol{\alpha}_k\right)\left(\mathbf{a}_0 \otimes \mathbf{a}_0\right),\end{aligned} \tag{16}$$

$\mathcal{C}_{jkl}$ is a vector:

$$\begin{aligned}\mathcal{C}^{\mathcal{T}_i}_{jkl} = & \tfrac{\lambda}{2} \quad \boldsymbol{\alpha}_j\left(\boldsymbol{\alpha}_k.\boldsymbol{\alpha}_l\right) + \frac{\mu}{2}\left[\boldsymbol{\alpha}_l\left(\boldsymbol{\alpha}_j.\boldsymbol{\alpha}_k\right) + \boldsymbol{\alpha}_k\left(\boldsymbol{\alpha}_j.\boldsymbol{\alpha}_l\right)\right] \\ + & \left(\frac{\lambda^L - \lambda}{2} + \mu^L - \mu\right)\left(\mathbf{a}_0 \otimes \mathbf{a}_0\right)\left(\boldsymbol{\alpha}_j \otimes \boldsymbol{\alpha}_k\right)\left(\mathbf{a}_0 \otimes \mathbf{a}_0\right)\boldsymbol{\alpha}_l, j,k,l = 0,\ldots,3,\end{aligned} \tag{17}$$

and $\mathcal{D}_{jklm}$ is a scalar:

$$\begin{aligned}\mathcal{D}^{\mathcal{T}_i}_{jklm} = & \tfrac{\lambda}{8} \quad \left(\boldsymbol{\alpha}_j.\boldsymbol{\alpha}_k\right)\left(\boldsymbol{\alpha}_l.\boldsymbol{\alpha}_m\right) + \frac{\mu}{4}\left(\boldsymbol{\alpha}_j.\boldsymbol{\alpha}_m\right)\left(\boldsymbol{\alpha}_k.\boldsymbol{\alpha}_l\right) \\ + & \left(\frac{\lambda^L - \lambda}{8} + \frac{\mu^L - \mu}{4}\right)\left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)\left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right)\left(\mathbf{a}_0.\boldsymbol{\alpha}_l\right)\left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right), j,k,l,m = 0,\ldots,3.\end{aligned} \tag{18}$$

## 1.4 Non-linear isotropic elasticity - invariant to rigid translation

The elasticity models presented in the preceding sections were based on the displacements of nodes from their initial positions. In the case of a rigid translation, all nodes would be displaced and would require numerical propagation to compute their new position. The elastic energy should be invariant to rigid translation, so numerical propagation should not be required in this case. To overcome this problem Picinbono developed an expression for the elastic energy that was based on the variation in the edge lengths of the tetrahedral elements, as opposed to nodal displacements. In the edge length formulation, the non-linear isotropic elastic energy is given by [5]:

$$\begin{aligned}\mathbf{F}_p^{iso}(\mathcal{T}_i) = \frac{1}{162V^4}\Bigg\{ & (\lambda+\mu) \left[\sum_{\substack{\text{Edge}\\ j\text{–}k}} \left(L_{jk}^2 - l_{jk}^2\right)\mathcal{K}_{jk}\right]\left(\sum_{\substack{\text{Edge}\\ j-p}} \overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\mathcal{K}_{jp}\right) \\ + & \ \mu\Bigg[\sum_{\substack{\text{Edge}\\ j-p}} \left(L_{jp}^2 - l_{jp}^2\right)\mathcal{A}_p^2\left(\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\mathcal{A}_j^2 - \overrightarrow{\mathbf{Q}_k\mathbf{Q}_p}\mathcal{K}_{jk} - \overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}\mathcal{K}_{jm}\right) \\ + & \sum_{\substack{\text{Edge}\\ j-k\\ j\neq k}} \left(L_{jk}^2 - l_{jk}^2\right)\left(\overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}\mathcal{K}_{jkmp} - \overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\mathcal{A}_j^2\mathcal{K}_{kp} - \overrightarrow{\mathbf{Q}_k\mathbf{Q}_p}\mathcal{A}_k^2\mathcal{K}_{jp}\right)\Bigg]\Bigg\}\end{aligned} \tag{19}$$

where

$$\mathcal{K}_{jk} = \frac{l_{lm}^2\left(l_{jl}^2 + l_{kl}^2 + l_{jm}^2 + l_{km}^2 - 2l_{jk}^2 - l_{lm}^2\right) - \left(l_{jl}^2 - l_{jm}^2\right)\left(l_{kl}^2 - l_{km}^2\right)}{16} \tag{20}$$

and

$$\mathcal{K}_{jkmp} = \mathcal{K}_{jp}\mathcal{K}_{km} + \mathcal{K}_{jm}\mathcal{K}_{kp} - \mathcal{K}_{jk}\mathcal{K}_{mp}. \tag{21}$$

$l_{jk}$ is the initial length of edge $j-k$, $L_{jk}$ is the length of deformed edge $j-k$, $\mathcal{A}_j$ is the area of the face opposite node $j$ and $\mathbf{Q}_j$ is the position of node $j$ after the deformation, $\mathbf{P}_j \xrightarrow{\mathbf{\Phi}} \mathbf{Q}_j$. It should also be noted that $\mathcal{K}_{jk} = \mathcal{K}_{kj}$ and $\mathcal{K}_{jkmp} = \mathcal{K}_{kjmp} = \mathcal{K}_{jkpm} = \mathcal{K}_{kjpm} = \mathcal{K}_{mpjk} = \mathcal{K}_{mpkj} = \mathcal{K}_{pmjk} = \mathcal{K}_{pmkj}$.

Picinbono cites the primary motivation for deriving the translation invariant form to be the desire for an expression that explicitly removes all computation when the translation is entirely rigid. In this case $L_{jk} = l_{jk}$ for all edges, and the computation can be avoided. In practice we have found the original formulation based on nodal displacements to be less stable than the translation invariant form. Under substantial deformation, tetrahedra may reach meta-stable conformations and not return to their original conformation after the deformation force has been removed. The edge length formulation overcomes this problem.

The translation invariant expression for the non-linear isotropic elastic force (Eqn 19) is derived in [5], but does not appear to have been published elsewhere. The extension to anisotropic materials was not given in [5] and likewise, does not appear to have been published elsewhere. As noted above, the formulation based on edge length variation has two major advantages when compared to the original form based on nodal displacements. Firstly, the modified algorithm is invariant to rigid translation, and secondly, it overcomes the stability problem previously described. These facts motivated us to develop the edge length formulation for transversally isotropic materials.

## 2 Translation invariant formulation for non-linear, transversally isotropic elasticity

The equation relating the non-linear isotropic elastic force to variations in edge length (Eqn 19) was obtained by differentiating the elastic energy with respect to nodal position, $\mathbf{F}_p = \partial W / \partial \mathbf{Q}_p$ where:

$$W_{iso} = \frac{1}{324V^4} \left\{ \left( \frac{\lambda + \mu}{2} \right) \left[ \sum_{\substack{\text{Edge} \\ j\text{–}k}} \left( L_{jk}^2 - l_{jk}^2 \right) \mathcal{K}_{jk} \right]^2 + \mu \tilde{\Delta} \right\} \tag{22}$$

and

$$\begin{aligned} \tilde{\Delta} = \frac{1}{2} \quad & \textstyle\sum_{\substack{\text{Edge} \\ j\text{–}k}} \quad \left( L_{jk}^2 - l_{jk}^2 \right)^2 \mathcal{A}_j^2 \mathcal{A}_k^2 \\ - \quad & \textstyle\sum_{\substack{\text{Edge pair} \\ j\text{–}k,j\text{–}m}} \quad \left( L_{jk}^2 - l_{jk}^2 \right) \left( L_{jm}^2 - l_{jm}^2 \right) \mathcal{A}_j^2 \mathcal{K}_{km} \\ + \quad & \textstyle\sum_{\substack{\text{Edge pair} \\ j\text{–}k,m\text{–}n}} \quad \left( L_{jk}^2 - l_{jk}^2 \right) \left( L_{mn}^2 - l_{mn}^2 \right) \left( \mathcal{K}_{jn}\mathcal{K}_{km} + \mathcal{K}_{jm}\mathcal{K}_{kn} - \mathcal{K}_{jk}\mathcal{K}_{mn} \right). \end{aligned} \tag{23}$$

For the details leading to Eqn 22, see [5].

To develop an expression for the elastic energy in non-linear anisotropic materials, we separate the isotropic and anisotropic components:

$$W_a = W_{iso} + \left( \frac{\Delta\lambda}{2} + \Delta\mu \right) \left( \mathbf{a}_0^T E \mathbf{a}_0 \right)^2 \tag{24}$$

where $\Delta\lambda = \lambda^T - \lambda$ and $\Delta\mu = \mu^T - \mu$. It should be noted that all shearing and cross-stretching terms have been neglected, as was the case when the original formulation was developed (see

Section 1.3). The $\left(\mathbf{a}_0^T E \mathbf{a}_0\right)^2$ term can be expanded as follows. Firstly we use $E = \frac{1}{2}(C - I)$ to replace the Green-St. Venant tensor:

$$\left(\mathbf{a}_0^T E \mathbf{a}_0\right)^2 = \frac{1}{4}\mathbf{a}_0^T C \left(\mathbf{a}_0 \otimes \mathbf{a}_0\right) C \mathbf{a}_0 - \frac{1}{2}\mathbf{a}_0^T C \mathbf{a}_0 + \frac{1}{4} \tag{25}$$

and then express the Cauchy-Green deformation tensor, $C$, in terms of the edge lengths:

$$C = -\frac{1}{2}\sum_{\substack{j,k=0\\ j\neq k}}^{3} L_{jk}^2 \left(\boldsymbol{\alpha}_j \otimes \boldsymbol{\alpha}_k\right) \tag{26}$$

to obtain:

$$\begin{aligned}\left(\mathbf{a}_0^T E \mathbf{a}_0\right)^2 = \frac{1}{2} \quad & \sum\nolimits_{\substack{\text{Edge}\\ j\text{–}k}} \left(L_{jk}^2 - l_{jk}^2\right)^2 \left[\frac{1}{2}\left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right)^2\right] \\ - \quad & \sum\nolimits_{\substack{\text{Edge pair}\\ j\text{–}k,j\text{–}m}} \left(L_{jk}^2 - l_{jk}^2\right)\left(L_{jm}^2 - l_{jm}^2\right) \left[-\frac{1}{2}\left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right)\left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right)\right] \\ + \quad & \sum\nolimits_{\substack{\text{Edge pair}\\ j\text{–}k,m\text{–}n}} \left(L_{jk}^2 - l_{jk}^2\right)\left(L_{mn}^2 - l_{mn}^2\right) \left[\frac{1}{2}\left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)\left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right)\left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right)\left(\mathbf{a}_0.\boldsymbol{\alpha}_n\right)\right]\end{aligned} \tag{27}$$

For the details, see Appendix B.

Equations 23 and 27 are of the same form and can be combined:

$$\begin{aligned}\tilde{\Delta}_a &= \frac{\mu}{324V^4}\tilde{\Delta} + \left(\frac{\Delta\lambda}{2} + \Delta\mu\right)\left(\mathbf{a}_0^T E \mathbf{a}_0\right)^2 \\ &= \frac{1}{2}\sum_{\substack{\text{Edge}\\ j\text{–}k}} \left(L_{jk}^2 - l_{jk}^2\right)^2 \left[\frac{\mu}{324V^4}\mathcal{A}_j^2\mathcal{A}_k^2 + \left(\frac{\Delta\lambda}{2} + \Delta\mu\right)\frac{1}{2}\left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2\left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right)^2\right] \qquad (28) \\ &\quad - \sum_{\substack{\text{Edge pair}\\ j\text{–}k,j\text{–}m}} \left(L_{jk}^2 - l_{jk}^2\right)\left(L_{jm}^2 - l_{jm}^2\right)\left[\frac{\mu}{324V^4}\mathcal{A}_j^2\mathcal{K}_{km}\right. \\ &\qquad\qquad \left. - \left(\frac{\Delta\lambda}{2} + \Delta\mu\right)\frac{1}{2}\left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2\left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right)\left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right)\right] \\ &\quad + \sum_{\substack{\text{Edge pair}\\ j\text{–}k,m\text{–}n}} \left(L_{jk}^2 - l_{jk}^2\right)\left(L_{mn}^2 - l_{mn}^2\right)\left[\frac{\mu}{324V^4}\left(\mathcal{K}_{jn}\mathcal{K}_{km} + \mathcal{K}_{jm}\mathcal{K}_{kn} - \mathcal{K}_{jk}\mathcal{K}_{mn}\right)\right. \\ &\qquad\qquad \left. + \left(\frac{\Delta\lambda}{2} + \Delta\mu\right)\frac{1}{2}\left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)\left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right)\left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right)\left(\mathbf{a}_0.\boldsymbol{\alpha}_n\right)\right].\end{aligned}$$

Substituting Eqn 22 into Eqn 24 yields an expression for the elastic energy:

$$\begin{aligned}W_a &= \frac{1}{324V^4}\left(\frac{\lambda+\mu}{2}\right)\left[\sum_{\substack{\text{Edge}\\ j\text{–}k}}\left(L_{jk}^2 - l_{jk}^2\right)\mathcal{K}_{jk}\right]^2 + \frac{\mu}{324V^4}\tilde{\Delta} + \left(\frac{\Delta\lambda}{2} + \Delta\mu\right)\left(\mathbf{a}_0^T E \mathbf{a}_0\right)^2 \\ &= \frac{1}{324V^4}\left(\frac{\lambda+\mu}{2}\right)\left[\sum_{\substack{\text{Edge}\\ j\text{–}k}}\left(L_{jk}^2 - l_{jk}^2\right)\mathcal{K}_{jk}\right]^2 + \tilde{\Delta}_a \qquad (29)\end{aligned}$$

where $\tilde{\Delta}_a$ is given in Eqn 28. The elastic force is obtained by differentiating the energy and the result is (see Appendix C):

$$\begin{aligned}
\mathbf{F}^a_p(\mathcal{T}_i) = \quad & \frac{\lambda+\mu}{162V^4}\left[\sum_{\substack{\text{Edge}\\ j\text{–}k}}\left(L^2_{jk}-l^2_{jk}\right)\mathcal{K}_{jk}\right]\left[\sum_{\substack{\text{Edge}\\ j\text{–}p}}\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\mathcal{K}_{jp}\right] \qquad (30)\\
+ \quad & \frac{\mu}{162V^4}\mathcal{A}^2_p\sum_{\substack{\text{Edge}\\ j\text{–}p}}\left(L^2_{jp}-l^2_{jp}\right)\left[\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\mathcal{A}^2_j-\overrightarrow{\mathbf{Q}_k\mathbf{Q}_p}\mathcal{K}_{jk}-\overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}\mathcal{K}_{jm}\right]\\
+ \quad & \left(\frac{\Delta\lambda}{2}+\Delta\mu\right)(\mathbf{a}_0.\boldsymbol{\alpha}_p)^2\sum_{\substack{\text{Edge}\\ j\text{–}p}}(\mathbf{a}_0.\boldsymbol{\alpha}_j)\left(L^2_{jp}-l^2_{jp}\right)\\
& \qquad\left[\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_j)+\overrightarrow{\mathbf{Q}_k\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_k)+\overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_m)\right]\\
+ \quad & \frac{\mu}{162V^4}\sum_{\substack{\text{Edge}\\ j\text{–}k\\ j,k\neq p}}\left(L^2_{jk}-l^2_{jk}\right)\left[\overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}\mathcal{K}_{jkmp}-\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\mathcal{A}^2_j\mathcal{K}_{kp}-\overrightarrow{\mathbf{Q}_k\mathbf{Q}_p}\mathcal{A}^2_k\mathcal{K}_{jp}\right]\\
+ \quad & \left(\frac{\Delta\lambda}{2}+\Delta\mu\right)(\mathbf{a}_0.\boldsymbol{\alpha}_p)\sum_{\substack{\text{Edge}\\ j\text{–}k\\ j,k\neq p}}\left(L^2_{jk}-l^2_{jk}\right)(\mathbf{a}_0.\boldsymbol{\alpha}_j)(\mathbf{a}_0.\boldsymbol{\alpha}_k)\\
& \qquad\left[\overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_m)+\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_j)+\overrightarrow{\mathbf{Q}_k\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_k)\right].
\end{aligned}$$

The above expression computes the non-linear elastic force required to deform an anisotropic material. The formulation is based on the variation in the edge lengths of the tetrahedral finite elements and is thus invariant to rigid rotations and translations. Comparing the above expression with Eqn 19 shows the anisotropic force is related to the isotropic force as follows:

$$\begin{aligned}
\mathbf{F}^a_p(\mathcal{T}_i) = \quad & \mathbf{F}^{iso}_p(\mathcal{T}_i)\\
+ \quad & \left(\frac{\Delta\lambda}{2}+\Delta\mu\right)(\mathbf{a}_0.\boldsymbol{\alpha}_p)^2\sum_{\substack{\text{Edge}\\ j\text{–}p}}(\mathbf{a}_0.\boldsymbol{\alpha}_j)\left(L^2_{jp}-l^2_{jp}\right)\\
& \qquad\left[\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_j)+\overrightarrow{\mathbf{Q}_k\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_k)+\overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_m)\right]\\
+ \quad & \left(\frac{\Delta\lambda}{2}+\Delta\mu\right)(\mathbf{a}_0.\boldsymbol{\alpha}_p)\sum_{\substack{\text{Edge}\\ j\text{–}k\\ j,k\neq p}}\left(L^2_{jk}-l^2_{jk}\right)(\mathbf{a}_0.\boldsymbol{\alpha}_j)(\mathbf{a}_0.\boldsymbol{\alpha}_k)\\
& \qquad\left[\overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_m)+\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_j)+\overrightarrow{\mathbf{Q}_k\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_k)\right].
\end{aligned}$$

The expression for $\mathbf{F}_p$ in Eqn. 30 consists of summations that collect terms associated with the 3 $j$–$p$ edges (edges radiating from node $p$) and the 3 $j$–$k$ edges (edges not connected with node $p$). The summation over edges $j-k$ can be converted to a summation over edges $j-p$ using a simple transformation of indices. The two summations can then be combined and all terms can be packed into a single summation:

$$
\begin{aligned}
\mathbf{F}^a_p(\mathcal{T}_i) = \sum_{\substack{\text{Edge}\\ j\text{–}p}} \quad & \overrightarrow{\mathbf{Q}_j\mathbf{Q}_p} \quad \Bigg[ \left(\frac{\Delta\lambda}{2} + \Delta\mu\right) (\mathbf{a}_0.\boldsymbol{\alpha}_p)(\mathbf{a}_0.\boldsymbol{\alpha}_j) \\
& \qquad \left\{ \left(L^2_{jp} - l^2_{jp}\right)(\mathbf{a}_0.\boldsymbol{\alpha}_p)(\mathbf{a}_0.\boldsymbol{\alpha}_j) + \left(L^2_{mk} - l^2_{mk}\right)(\mathbf{a}_0.\boldsymbol{\alpha}_m)(\mathbf{a}_0.\boldsymbol{\alpha}_k) \right\} \\
& + \frac{\mu}{162V^4} \left\{ \left(L^2_{jp} - l^2_{jp}\right) \mathcal{A}^2_p \mathcal{A}^2_j + \left(L^2_{mk} - l^2_{mk}\right) \mathcal{K}_{mkjp} \right\} + \frac{\lambda+\mu}{162V^4} \left( \sum_{\substack{\text{Edge}\\ j\text{–}k}} \left(L^2_{jk} - l^2_{jk}\right) \mathcal{K}_{jk} \right) \mathcal{K}_{jp} \Bigg] \\
& + \overrightarrow{\mathbf{Q}_k\mathbf{Q}_p} \quad \Bigg[ \left(\frac{\Delta\lambda}{2} + \Delta\mu\right) (\mathbf{a}_0.\boldsymbol{\alpha}_p)(\mathbf{a}_0.\boldsymbol{\alpha}_k) \\
& \qquad \left\{ \left(L^2_{jp} - l^2_{jp}\right)(\mathbf{a}_0.\boldsymbol{\alpha}_p)(\mathbf{a}_0.\boldsymbol{\alpha}_j) + \left(L^2_{mk} - l^2_{mk}\right)(\mathbf{a}_0.\boldsymbol{\alpha}_m)(\mathbf{a}_0.\boldsymbol{\alpha}_k) \right\} \\
& - \frac{\mu}{162V^4} \left\{ \left(L^2_{jp} - l^2_{jp}\right) \mathcal{A}^2_p \mathcal{K}_{jk} + \left(L^2_{mk} - l^2_{mk}\right) \mathcal{A}^2_k \mathcal{K}_{mp} \right\} \Bigg] \\
& + \overrightarrow{\mathbf{Q}_m\mathbf{Q}_p} \quad \Bigg[ \left(\frac{\Delta\lambda}{2} + \Delta\mu\right) (\mathbf{a}_0.\boldsymbol{\alpha}_p)(\mathbf{a}_0.\boldsymbol{\alpha}_m) \\
& \qquad \left\{ \left(L^2_{jp} - l^2_{jp}\right)(\mathbf{a}_0.\boldsymbol{\alpha}_p)(\mathbf{a}_0.\boldsymbol{\alpha}_j) + \left(L^2_{mk} - l^2_{mk}\right)(\mathbf{a}_0.\boldsymbol{\alpha}_m)(\mathbf{a}_0.\boldsymbol{\alpha}_k) \right\} \\
& - \frac{\mu}{162V^4} \left\{ \left(L^2_{jp} - l^2_{jp}\right) \mathcal{A}^2_p \mathcal{K}_{jm} + \left(L^2_{mk} - l^2_{mk}\right) \mathcal{A}^2_m \mathcal{K}_{kp} \right\} \Bigg]
\end{aligned}
\tag{31}
$$

In practice we have found that the form consisting of the single summation (Eqn 31) can be evaluated approximately 35% faster than the form consisting of two summations (Eqn. 30).

Given the expression for $\mathbf{F}^a_p(\mathcal{T}_i)$ (Eqn 31), the nodal positions can be propagated using the Newtonian scheme in Eqn 10. For further details, we refer the reader to Appendices A and D.

## 3 Implementation and Demonstration

The algorithm was implemented in a prototype simulator consisting of two PCs connected by a network and an Immersion Laparoscopic Surgical Workstation (LSW)[7]. The simulator was logically separated into two components: one PC hosting the simulation and the other hosting the LSW. At the start of each iteration the simulation would send a request to the machine hosting the LSW for the current position and orientation of the input device. Given the position and orientation of the input device, the simulation machine would then compute the forces arising from the user intervention and update the visual display accordingly. The simulation machine would then send the force information to the LSW-host machine to update the LSW.

The LSW would then exert the appropriate forces on the user, providing tactile feedback. The simulation executed with an update frequency that was determined by the complexity of the simulation algorithm and the complexity of the deformable models. The simulation frequency should be maintained at about 50Hz so the visual display is acceptable. In contrast, the haptic device should be updated at a frequency no less than about 300 - 500Hz to ensure the tactile feedback is continuous and realistic. To overcome the disparity in update frequencies a force extrapolation scheme [3] was used.

The algorithm was demonstrated using a model of a muscle exhibiting strong anisotropy. The model consisted of 2552 tetrahedra and 861 nodes and had the following material properties: in the direction along the length of the muscle, Young's modulus was 150kPa and the Poisson ratio was 0.4. In the plane transverse to the longitudinal axis the Young's modulus was 20kPa and the Poisson ratio was 0.4. The simulation was tested on a Linux-based 2.80GHz Pentium 4 PC with an nVidia GeForce4 Ti4600 graphics card. The LSW was hosted on an SGI 330 PC running MS Windows 2000. Unix and MS Windows sockets were used for network communication between the simulation and LSW-host machines. A deformation action is shown in Figure 1. As the number of nodes requiring numerical propagation increased, the update frequency decreased, as shown in Figure 2. The timing results shown in Figure 3 indicated that the application was fill-rate limited for small simulation volumes. In other words, the simulation frequency was limited by the rate at which the host machine's graphics card could refresh the frame buffer. The application became CPU limited when the simulation volume contained more than about 2000 tetrahedra. To maintain the update frequency at an acceptable level, nodes that were subject to a force having magnitude less than a predetermined threshold were not propagated. The update frequency primarily depended on the number of tetrahedra that were being deformed in each iteration. However the computational load depended on both the complexity of the algorithm being used to model the deformation, and also the configuration of the underlying tetrahedral mesh. A hierarchical tetrahedral mesh could have been used to reduce the number of finite elements in non-essential areas and an adapative algorithm could have been used to reduce the computational cost associated with the algorithm. An adaptive algorithm would seek to simulate peripheral regions using a low cost algorithm such as a mass-spring model, while areas of interest would be simulated with a combination of linear, non-linear, isotropic and anisotropic FEM algorithms, as required.

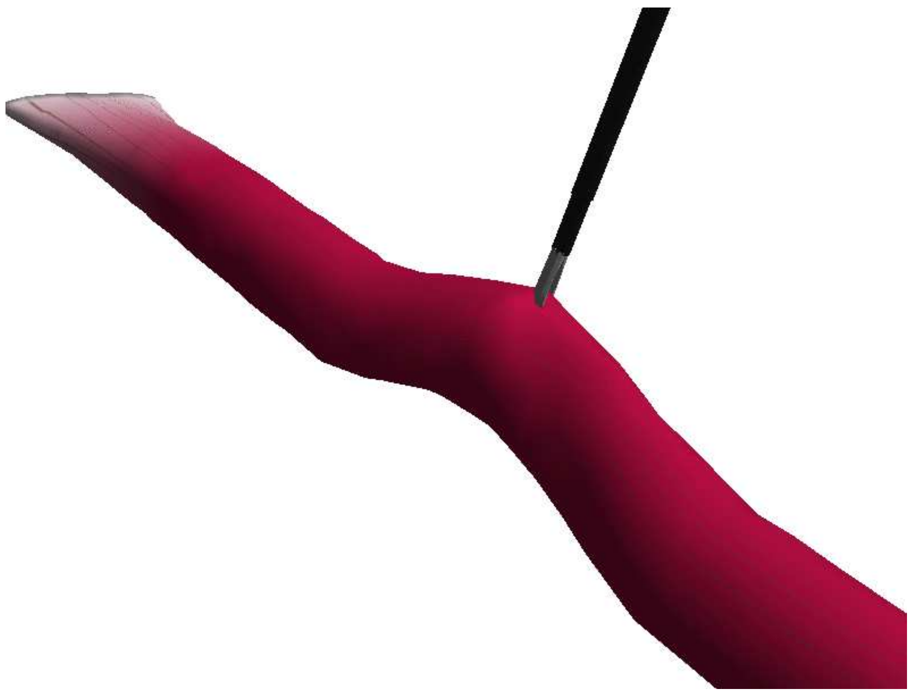


Figure 1: Deformation of an anisotropic muscle. The deformation shown required the numerical propagation of 161 nodes.

Preliminary analysis indicated the performance of the application was limited by cache misses. Furthermore, the various expressions for $\mathbf{F}_p$ (eg. Eqn. 31) consist of a large number of multiply-and-add operations, which could be efficiently evaluated using Single Instruction Multiple Data (SIMD) instructions. We therefore expect basic code tuning could substantially improve the performance.

In Section 1.4 we noted that the edge length formulation of the FEM algorithm overcomes the stability problems inherent in Picinbono's original algorithm that was based on nodal displacements. Stability is compromised when tetrahedra are contorted into metastable conformations and cannot return to their original configurations when the deforming force is removed. In Figures 4a-d we show the deformation of a $3 \times 3 \times 3$ block using Picinbono's non-linear anisotropic algorithm (Eqns 13, 16, 17 and 18). Figure 4e shows the configuration after the deforming force has been removed and allowed to stabilise. The volume does not return to its original shape. The same sequence of deformations using our non-linear anisotropic algorithm based on edge length variation is shown in Figures 4f-i. Figure 4j shows the volume immediately after the deforming force has been removed. Given sufficient time to overcome the damping

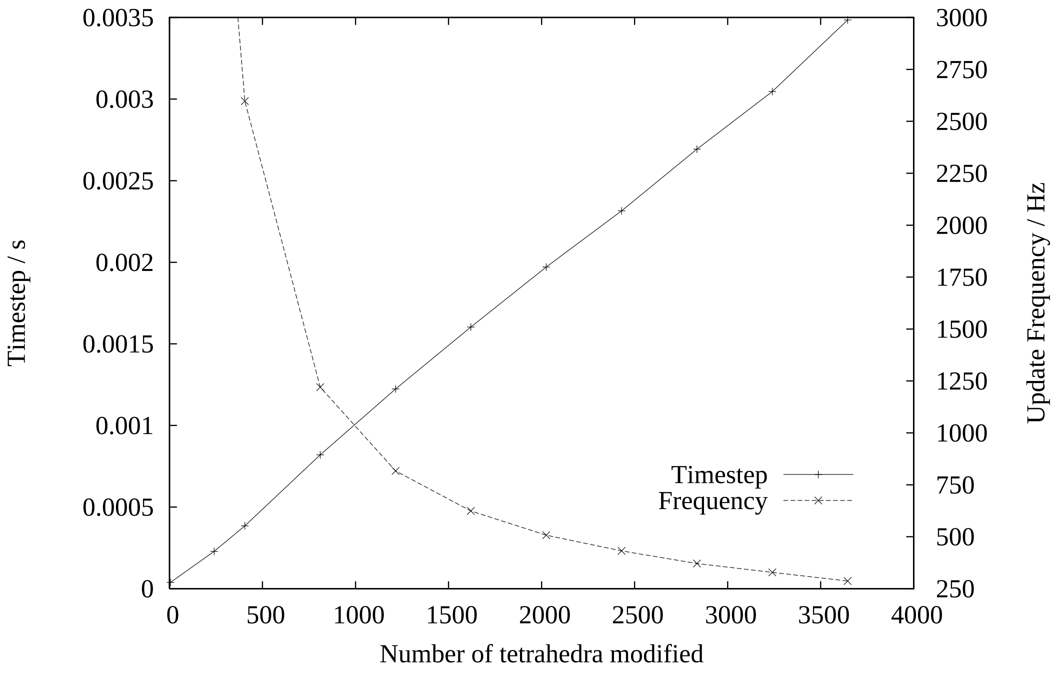


Figure 2: Performance of the non-linear anisotropic algorithm. The timing results were generated by applying a sinusoidal deformation to a rectangular mesh of dimension 10 $\times$ 10 $\times$ $n$, where the mesh height, $n$, was chosen to generate the required number of tetrahedra.

force, the volume returns to its original configuration. The deformation sequences demonstrate that the algorithm based on edge length variations is more stable than the original algorithm based on nodal displacements.

## 4 Summary

We have presented an algorithm for modeling soft tissue deformation in real time. The algorithm can model large deformations in inhomogeneous materials at interactive rates and therefore has the potential to advance the development of high-fidelity surgical simulators.

The algorithm is non-linear, anisotropic and invariant to rigid rotations and translations. It uses the finite element method with tetrahedral elements to compute the elastic force that results when the target model is deformed.

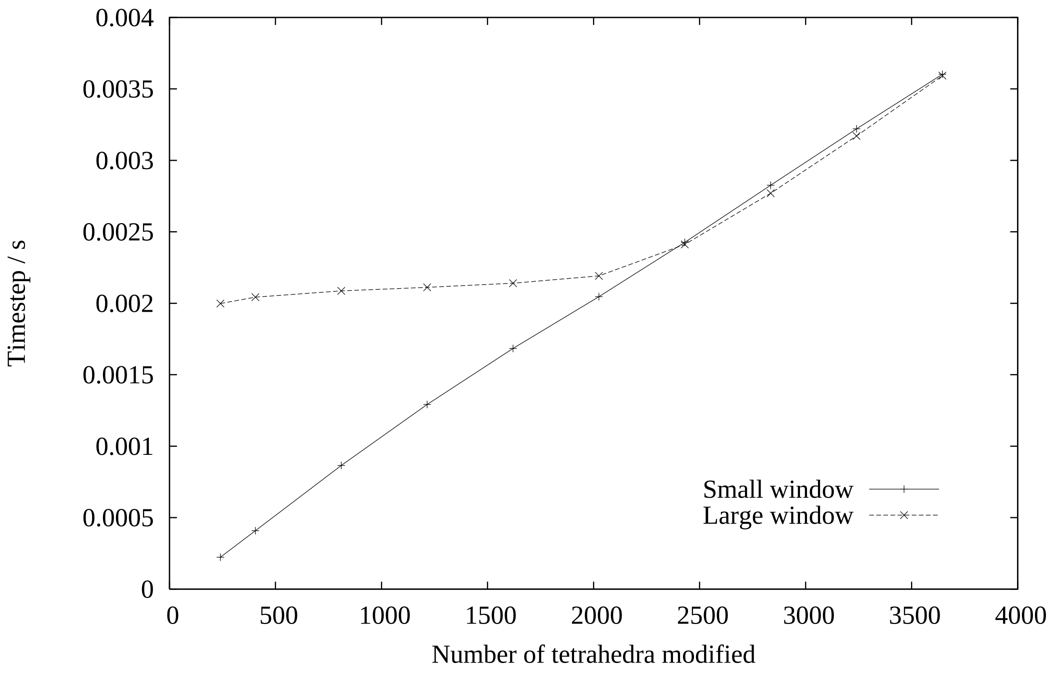


Figure 3: Timing results for the non-linear anisotropic algorithm indicated the application was fill-rate limited until there were about 2000 tetrahedra being updated in each iteration. Small window: 38 x 101 pixels, large window: 1270 x 944 pixels.

## 5 Acknowledgments

This work was supported by the Australian Partnership for Advanced Computing, a Pilot Project Grant from the Queensland Parallel Supercomputing Foundation and the Advanced Computational Modelling Centre at the University of Queensland. The author would like to thank surgeons David Nicol and Andrew Stevenson for supporting the project.

## References

[1] Y-C. Fung, Biomechanics: Mechanical Properties of Living Tissues, 2nd edition, Springer-Verlag (New York, 1993).

[2] S. Cotin, H. Delingette, N. Ayache, A hybrid elastic model for real-time cutting, deformations, and force feedback for surgery training and simulation, The Visual Computer 16 (2000) 437-452.

[3] G. Picinbono, J-C. Lombardo, H. Delingette, N. Ayache, Improving realism of a surgery simulator: linear anisotropic elasticity, complex interactions and force extrapolation, The Journal of Visualization and Computer Animation 13 (2002) 147-167.

[4] G. Picinbono, H. Delingette, N. Ayache, Non-linear anisotropic elasticity for real-time surgery simulation, Graphical Models 65 (2003) 305-321.

[5] G. Picinbono, Modèles géométriques et physiques pour la simulation d'interventions chirurgicales, Thèse de sciences, université de Nice Sophia-Antipolis (2001).

[6] K-L. Bathe, Finite element procedures in engineering analysis, Prentice-Hall (Englewood Cliffs, New Jersey, 1982).

[7] Laparoscopic Surgical Workstation, Immersion Medical, Gaithersburg USA. See http://www.immersion.com/medical/.

[8] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, Numerical recipes in C++ : the art of scientific computing, Cambridge University Press (Cambridge, 2002).

[9] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, Springer Series in Computational Maths No. 14, Springer-Verlag (Heidelberg, 2010).

[10] U.M. Ascher, L.R. Petzold, Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations, Society for Industrial and Applied Mathematics (Philadelphia, 1998).

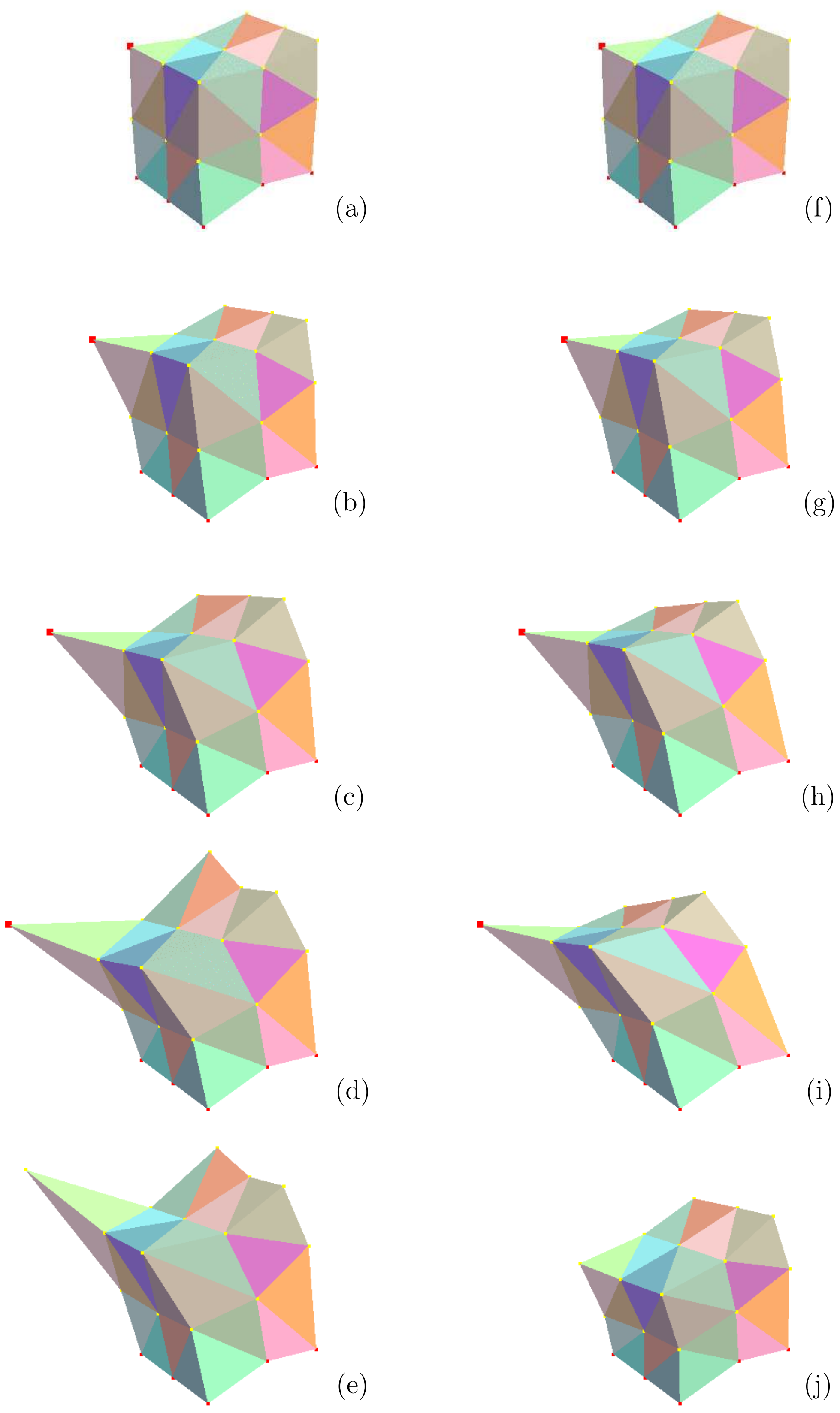


Figure 4: Deformation sequence modelled using non-linear anisotropic FEM. The sequence in (a-e) used Picinbono's algorithm based on nodal displacements, and the sequence in (f-j) used our modified algorithm based on edge length variation. The nodes at the bottom of the grid were fixed in position. The top left node was being manipulated by the user.

## A Newtonian integration scheme

At each time step in the simulation, the nodal positions are propagated using the Newtonian scheme in Eqn 10, which leads to the explicit integration scheme in Eqn 11. The expression for $\mathbf{P}_i^{t+1}$ (Eqn 11) is obtained by expanding both $\mathbf{P}_i^{t+1}$ and $\mathbf{P}_i^{t-1}$ using Taylor series expansions:

$$\begin{aligned}
\mathbf{P}_{t+1} &= \mathbf{P}_t + \eta\Delta t\frac{d\mathbf{P}_t}{dt} + \frac{(\eta\Delta t)^2}{2}\frac{d^2\mathbf{P}_t}{dt^2} + O\left((\eta\Delta t)^3\right), && \text{(A.1)}\\
\mathbf{P}_{t-1} &= \mathbf{P}_t - \Delta t\frac{d\mathbf{P}_t}{dt} + \frac{(\Delta t)^2}{2}\frac{d^2\mathbf{P}_t}{dt^2} - O\left(\Delta t^3\right) && \text{(A.2)}
\end{aligned}$$

where the timestep between times $t-1$ and $t$ is $\Delta t$ and the timestep between times $t$ and $t+1$ is $\eta\Delta t$. Rearranging and substituting terms in Eqns A.1 and A.2 yields expressions for the velocity and acceleration:

$$\begin{aligned}
\frac{d\mathbf{P}_t}{dt} &= \frac{\mathbf{P}_{t+1} + (\eta^2-1)\mathbf{P}_t - \eta^2\mathbf{P}_{t-1}}{\eta(\eta+1)\Delta t},\\
\frac{d^2\mathbf{P}_t}{dt^2} &= \frac{2}{\Delta t^2}\left[\frac{\eta-1}{\eta(\eta^2-1)}\mathbf{P}_{t+1} - \frac{1}{\eta}\mathbf{P}_t + \frac{\eta-1}{\eta^2-1}\mathbf{P}_{t-1}\right]
\end{aligned}$$

which when substituted into Eqn 10 yields the expression for the numerical propagation, Eqn 11.

As an alternative to explicit Euler integration, one could numerically integrate Eqn 10 using more sophisticated methods. Explicit fourth order [8] and implicit fifth order [9] Runge-Kutta algorithms with adaptive step size were tested. Both algorithms enabled a larger step size to be used compared to the explicit Euler method, despite the additional computational cost incurred when recomputing the forces at the intermediate steps. We recommend the implicit method as it appears to be more stable the explicit methods. The implemention details are provided in Appendix D.

It should be noted that Picinbono et al [2], [3], [4] use the following scheme:

$$m_i\frac{d^2\mathbf{P}_i}{dt^2} = \mathbf{F}_i + \gamma_i\frac{d\mathbf{P}_i}{dt} \tag{A.3}$$

where the sign on the damping force has been changed. The second order system in Eqn A.3 can be converted to two first order equations of the form:

$$\begin{aligned}
\frac{d\mathbf{P}}{dt} &= v,\\
\frac{d\mathbf{v}}{dt} &- \gamma v = 0
\end{aligned} \tag{A.4}$$

where we have set $m = 1$ and neglected the force term for convenience. Eqn A.4 has a solution of the form:

$$v = v_o e^{\gamma t}$$

which appears to be divergent when the damping coefficient $\gamma$ is chosen to be positive. In practice we have found this implementation to be unstable.

Picinbono et al also assume equal timesteps between successive iterations. In other words, they do not differentiate between the timesteps between times $t-1$ and $t$, and times $t$ and $t+1$ *ie.* $\eta = 1$. The resulting integration scheme is ([2], [3], [4]):

$$\left(\frac{m_i}{\Delta t^2} - \frac{\gamma_i}{2\Delta t}\right)\mathbf{P}_i^{t+1} = \mathbf{F}_i + \frac{2m_i}{\Delta t^2}\mathbf{P}_i^t - \left(\frac{m_i}{\Delta t^2} + \frac{\gamma_i}{2\Delta t}\right)\mathbf{P}_i^{t-1}. \tag{A.5}$$

In general, a constant timestep will not be appropriate for a real time simulation, so Eqn A.5 is an approximation.

## B Derivation of Eqn 27

The expression for $\left(\mathbf{a}_0^T E \mathbf{a}_0\right)^2$ in terms of edge lengths was obtained by substituting the Cauchy-Green deformation tensor for the Green-St. Venant tensor and simplifying:

$$\begin{aligned}\left(\mathbf{a}_0^T E \mathbf{a}_0\right)^2 &= \mathbf{a}_0^T E \mathbf{a}_0 \; \mathbf{a}_0^T E \mathbf{a}_0 \\ &= \mathbf{a}_0^T \left[\frac{1}{2}(C-I)\right] \mathbf{a}_0 \; \mathbf{a}_0^T \left[\frac{1}{2}(C-I)\right] \mathbf{a}_0 \\ &= \frac{1}{4}\mathbf{a}_0^T C \mathbf{a}_0 \; \mathbf{a}_0^T C \mathbf{a}_0 - \frac{1}{2}\mathbf{a}_0^T C \mathbf{a}_0 \left(\mathbf{a}_0 . \mathbf{a}_0\right) + \frac{1}{4}\left(\mathbf{a}_0 . \mathbf{a}_0\right)^2 \\ &= \frac{1}{4}\mathbf{a}_0^T C \left(\mathbf{a}_0 \otimes \mathbf{a}_0\right) C \mathbf{a}_0 - \frac{1}{2}\mathbf{a}_0^T C \mathbf{a}_0 + \frac{1}{4}\end{aligned} \tag{B.1}$$

where the normality of the vectors specifying the direction of the anisotropy was used:

$$\left(\mathbf{a}_0 . \mathbf{a}_0\right) = \left\|\mathbf{a}_0\right\|^2 = 1. \tag{B.2}$$

The Cauchy-Green deformation tensor can be expressed in terms of the edge lengths (for details see [5]):

$$C = -\frac{1}{2}\sum_{\substack{j,k=0 \\ j \neq k}}^{3} L_{jk}^2 \left(\boldsymbol{\alpha}_j \otimes \boldsymbol{\alpha}_k\right) \tag{B.3}$$

and substituting this expression into Eqn B.1 we obtain:

$$\begin{aligned}\left(\mathbf{a}_0^T E \mathbf{a}_0\right)^2 = \ & \tfrac{1}{4} \ \mathbf{a}_0^T \left[-\frac{1}{2}\sum_{\substack{j,k=0 \\ j \neq k}}^{3} L_{jk}^2 \left(\boldsymbol{\alpha}_j \otimes \boldsymbol{\alpha}_k\right)\right] \left(\mathbf{a}_0 \otimes \mathbf{a}_0\right) \left[-\frac{1}{2}\sum_{\substack{m,n=0 \\ m \neq n}}^{3} L_{mn}^2 \left(\boldsymbol{\alpha}_m \otimes \boldsymbol{\alpha}_n\right)\right] \mathbf{a}_0 \\ - \ & \tfrac{1}{2} \ \mathbf{a}_0^T \left[-\frac{1}{2}\sum_{\substack{j,k=0 \\ j \neq k}}^{3} L_{jk}^2 \left(\boldsymbol{\alpha}_j \otimes \boldsymbol{\alpha}_k\right)\right] \mathbf{a}_0 + \frac{1}{4} \\ = \ & \tfrac{1}{16} \ \sum_{\substack{j,k=0 \\ j \neq k}}^{3} \sum_{\substack{m,n=0 \\ m \neq n}}^{3} L_{jk}^2 L_{mn}^2 \left(\mathbf{a}_0 . \boldsymbol{\alpha}_j\right)\left(\mathbf{a}_0 . \boldsymbol{\alpha}_k\right)\left(\mathbf{a}_0 . \boldsymbol{\alpha}_m\right)\left(\mathbf{a}_0 . \boldsymbol{\alpha}_n\right) \\ + \ & \tfrac{1}{4} \ \sum_{\substack{j,k=0 \\ j \neq k}}^{3} L_{jk}^2 \left(\mathbf{a}_0 . \boldsymbol{\alpha}_j\right)\left(\mathbf{a}_0 . \boldsymbol{\alpha}_k\right) + \frac{1}{4}\end{aligned} \tag{B.4}$$

The summations in Eqn B.4 can be split as follows:

$$\begin{aligned}
\left(\mathbf{a}_0^T E \mathbf{a}_0\right)^2 = \; & \tfrac{1}{16} \left[ 4 \sum_{\text{Edge } j\text{–}k} L_{jk}^4 \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right)^2 \right. \\
& + 8 \sum_{\substack{\text{Edge pair} \\ j\text{–}k, j\text{–}m}} L_{jk}^2 L_{jm}^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right) \\
& \left. + 8 \sum_{\substack{\text{Edge pair} \\ j\text{–}k, m\text{–}n}} L_{jk}^2 L_{mn}^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_n\right) \right] \qquad \text{(B.5)} \\
+ \; & \frac{1}{4} \left[ 2 \sum_{\text{Edge } j\text{–}k} L_{jk}^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \right] + \frac{1}{4} \qquad \text{(B.6)}
\end{aligned}$$

and substituting:

$$L_{jk}^2 = \left(L_{jk}^2 - l_{jk}^2\right) + l_{jk}^2 \qquad \text{(B.7)}$$

we obtain:

$$\begin{aligned}
\left(\mathbf{a}_0^T E \mathbf{a}_0\right)^2 = \; & \tfrac{1}{4} \sum_{\text{Edge } j\text{–}k} \left[\left(L_{jk}^2 - l_{jk}^2\right) + l_{jk}^2\right]^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right)^2 \\
+ \; & \tfrac{1}{2} \sum_{\substack{\text{Edge pair} \\ j\text{–}k, j\text{–}m}} \left[\left(L_{jk}^2 - l_{jk}^2\right) + l_{jk}^2\right] \left[\left(L_{jm}^2 - l_{jm}^2\right) + l_{jm}^2\right] \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right) \\
+ \; & \tfrac{1}{2} \sum_{\substack{\text{Edge pair} \\ j\text{–}k, m\text{–}n}} \left[\left(L_{jk}^2 - l_{jk}^2\right) + l_{jk}^2\right] \left[\left(L_{mn}^2 - l_{mn}^2\right) + l_{mn}^2\right] \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_n\right) \\
+ \; & \tfrac{1}{2} \sum_{\text{Edge } j\text{–}k} \left[\left(L_{jk}^2 - l_{jk}^2\right) + l_{jk}^2\right] \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) + \frac{1}{4} \qquad \text{(B.8)}
\end{aligned}$$

Expanding Eqn B.8 and separating terms yields:

$$
\begin{aligned}
\left(\mathbf{a}_0^T E \mathbf{a}_0\right)^2 = \; & \tfrac{1}{2} \sum_{\text{Edge } j\text{–}k} \left(L_{jk}^2 - l_{jk}^2\right)^2 \left[\tfrac{1}{2} \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right)^2\right] \\
& - \sum_{\substack{\text{Edge pair} \\ j\text{–}k,j\text{–}m}} \left(L_{jk}^2 - l_{jk}^2\right)^2 \left(L_{jm}^2 - l_{jm}^2\right)^2 \left[-\tfrac{1}{2} \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right)\right] \\
& + \sum_{\substack{\text{Edge pair} \\ j\text{–}k,m\text{–}n}} \left(L_{jk}^2 - l_{jk}^2\right)^2 \left(L_{mn}^2 - l_{mn}^2\right)^2 \left[\tfrac{1}{2} \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_n\right)\right] \qquad \Bigg] \; T1 \\
& + \tfrac{1}{2} \Bigg[ \tfrac{1}{2} \sum_{\text{Edge } j\text{–}k} l_{jk}^4 \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right)^2 \\
& \qquad + \sum_{\substack{\text{Edge pair} \\ j\text{–}k,j\text{–}m}} l_{jk}^2 l_{jm}^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right) \\
& \qquad + \sum_{\substack{\text{Edge pair} \\ j\text{–}k,m\text{–}n}} l_{jk}^2 l_{mn}^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_n\right) \Bigg] \qquad \Bigg] \; T2 \\
& + \tfrac{1}{2} \Bigg[ \sum_{\text{Edge } j\text{–}k} \left(L_{jk}^2 - l_{jk}^2\right) l_{jk}^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right)^2 \\
& \qquad + \sum_{\text{Edge } j\text{–}k} \left(L_{jk}^2 - l_{jk}^2\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \\
& \qquad + \sum_{\substack{\text{Edge pair} \\ j\text{–}k,j\text{–}m}} \left[\left(L_{jk}^2 - l_{jk}^2\right) l_{jm}^2 + \left(L_{jm}^2 - l_{jm}^2\right) l_{jk}^2\right] \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right) \\
& \qquad + \sum_{\substack{\text{Edge pair} \\ j\text{–}k,m\text{–}n}} \left[\left(L_{jk}^2 - l_{jk}^2\right) l_{mn}^2 + \left(L_{mn}^2 - l_{mn}^2\right) l_{jk}^2\right] \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_n\right) \Bigg] \qquad \Bigg] \; T3 \\
& + \tfrac{1}{2} \sum_{\text{Edge } j\text{–}k} l_{jk}^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \qquad \Big] \; T4 \\
& + \tfrac{1}{4} \qquad \Big] \; T5
\end{aligned}
\tag{B.9}
$$

To proceed we note the following simplifications. Term $T2$ in Eqn B.9 reduces to $\frac{1}{2}$:

$$
\begin{aligned}
& \tfrac{1}{2} \sum_{\text{Edge } j\text{–}k} l_{jk}^4 \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right)^2 \\
+ & \sum_{\substack{\text{Edge pair} \\ j\text{–}k,j\text{–}m}} l_{jk}^2 l_{jm}^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right) \\
+ & \sum_{\substack{\text{Edge pair} \\ j\text{–}k,m\text{–}n}} l_{jk}^2 l_{mn}^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_n\right) = \frac{1}{2},
\end{aligned}
$$

term $T3$ reduces to zero:

$$\begin{aligned}
&\sum_{\text{Edge } j\text{–}k} \left(L_{jk}^2 - l_{jk}^2\right) l_{jk}^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right)^2 \\
&\sum_{\text{Edge } j\text{–}k} \left(L_{jk}^2 - l_{jk}^2\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \\
&\sum_{\substack{\text{Edge pair}\\ j\text{–}k,j\text{–}m}} \left[\left(L_{jk}^2 - l_{jk}^2\right) l_{jm}^2 + \left(L_{jm}^2 - l_{jm}^2\right) l_{jk}^2\right] \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right) \\
&\sum_{\substack{\text{Edge pair}\\ j\text{–}k,m\text{–}n}} \left[\left(L_{jk}^2 - l_{jk}^2\right) l_{mn}^2 + \left(L_{mn}^2 - l_{mn}^2\right) l_{jk}^2\right] \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_n\right) \;=\; 0
\end{aligned}$$

and term $T4$ reduces to -1:

$$\sum_{\text{Edge } j\text{–}k} l_{jk}^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \;=\; -1.$$

Therefore the contribution from terms $T2$, $T3$, $T4$ and $T5$ in Eqn B.9 sum to zero and Eqn B.9 reduces to:

$$\begin{aligned}
\left(\mathbf{a}_0^T E \mathbf{a}_0\right)^2 = \;& \tfrac{1}{2} \sum_{\text{Edge } j\text{–}k} \left(L_{jk}^2 - l_{jk}^2\right)^2 \left[\tfrac{1}{2}\left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right)^2\right] \\
& - \sum_{\substack{\text{Edge pair}\\ j\text{–}k,j\text{–}m}} \left(L_{jk}^2 - l_{jk}^2\right) \left(L_{jm}^2 - l_{jm}^2\right) \left[-\tfrac{1}{2}\left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2 \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right)\right] \qquad \text{(B.10)} \\
& + \sum_{\substack{\text{Edge pair}\\ j\text{–}k,m\text{–}n}} \left(L_{jk}^2 - l_{jk}^2\right) \left(L_{mn}^2 - l_{mn}^2\right) \left[\tfrac{1}{2}\left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right) \left(\mathbf{a}_0.\boldsymbol{\alpha}_n\right)\right]
\end{aligned}$$

## C Derivation of Eqn 30

The non-linear anisotropic force is given by differentiating the corresponding elastic energy term, $W_a$ (Eqn 29):

$$
\begin{aligned}
W_a = {} & \frac{1}{324V^4}\left(\frac{\lambda+\mu}{2}\right)\left[\sum_{\substack{\text{Edge}\\ j\text{--}k}}\left(L_{jk}^2-l_{jk}^2\right)\mathcal{K}_{jk}\right]^2 && \Big]\, T1\\
& +\frac{1}{2}\sum_{\substack{\text{Edge}\\ j\text{--}k}}\left(L_{jk}^2-l_{jk}^2\right)^2\left[\frac{\mu}{324V^4}\mathcal{A}_j^2\mathcal{A}_k^2+\left(\frac{\Delta\lambda}{2}+\Delta\mu\right)\frac{1}{2}\left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2\left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right)^2\right] && \Big]\, T2\\
& -\sum_{\substack{\text{Edge pair}\\ j\text{--}k,j\text{--}m}}\left(L_{jk}^2-l_{jk}^2\right)\left(L_{jm}^2-l_{jm}^2\right)\Big[\tfrac{\mu}{324V^4}\mathcal{A}_j^2\mathcal{K}_{km} && \\
& \qquad -\left(\tfrac{\Delta\lambda}{2}+\Delta\mu\right)\tfrac{1}{2}\left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2\left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right)\left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right)\Big] && \Big]\, T3\\
& +\sum_{\substack{\text{Edge pair}\\ j\text{--}k,m\text{--}n}}\left(L_{jk}^2-l_{jk}^2\right)\left(L_{mn}^2-l_{mn}^2\right)\Big[\tfrac{\mu}{324V^4}\left(\mathcal{K}_{jn}\mathcal{K}_{km}+\mathcal{K}_{jm}\mathcal{K}_{kn}-\mathcal{K}_{jk}\mathcal{K}_{mn}\right) && \\
& \qquad +\left(\tfrac{\Delta\lambda}{2}+\Delta\mu\right)\tfrac{1}{2}\left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)\left(\mathbf{a}_0.\boldsymbol{\alpha}_k\right)\left(\mathbf{a}_0.\boldsymbol{\alpha}_m\right)\left(\mathbf{a}_0.\boldsymbol{\alpha}_n\right)\Big]\,. && \Big]\, T4
\end{aligned}
\tag{C.1}
$$

In differentiating the various terms in Eqn C.1 we will make use of the following:

$$
\frac{\partial}{\partial\mathbf{Q}_p}\left(L_{jk}^2-l_{jk}^2\right)k_{jk}=\begin{cases}2\overrightarrow{\mathbf{Q}_k\mathbf{Q}_p}k_{pk} & : \ p=j\\ 2\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}k_{jp} & : \ p=k\\ \mathbf{0} & : \ p\neq j, p\neq k\end{cases}
\tag{C.2}
$$

where $k_{jk}$ is a constant and $L_{jk}=\left\|\overrightarrow{\mathbf{Q}_k\mathbf{Q}_j}\right\|$. Differentiating term $T1$ yields:

$$
\frac{\partial\, T1}{\partial\mathbf{Q}_p}=\frac{\lambda+\mu}{162V^4}\left[\sum_{\substack{\text{Edge}\\ j\text{--}k}}\left(L_{jk}^2-l_{jk}^2\right)\mathcal{K}_{jk}\right]\left[\sum_{\substack{\text{Edge}\\ j\text{--}p}}\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\mathcal{K}_{jp}\right]
\tag{C.3}
$$

where:

$$
\frac{\partial}{\partial\mathbf{Q}_p}\left[\sum_{\substack{\text{Edge}\\ j\text{--}k}}\left(L_{jk}^2-l_{jk}^2\right)\mathcal{K}_{jk}\right]^2=2\left[\sum_{\substack{\text{Edge}\\ j\text{--}k}}\left(L_{jk}^2-l_{jk}^2\right)\mathcal{K}_{jk}\right]\left[\sum_{\substack{\text{Edge}\\ j\text{--}p}}\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\mathcal{K}_{jp}.\right]
\tag{C.4}
$$

Proceeding in a similar manner for terms $T2$, $T3$ and $T4$ we obtain:

$$
\frac{\partial\, T2}{\partial\mathbf{Q}_p}=\sum_{\substack{\text{Edge}\\ j\text{--}p}}\left(L_{jp}^2-l_{jp}^2\right)\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\left[\frac{\mu}{162V^4}\mathcal{A}_p^2\mathcal{A}_j^2+\left(\frac{\Delta\lambda}{2}+\Delta\mu\right)\left(\mathbf{a}_0.\boldsymbol{\alpha}_j\right)^2\left(\mathbf{a}_0.\boldsymbol{\alpha}_p\right)^2\right],
\tag{C.5}
$$

$$\frac{\partial\, T3}{\partial \mathbf{Q}_p} = \sum_{\substack{\text{Edge pair}\\ j\text{–}p,k\text{–}p}} \left[\left(L_{jp}^2 - l_{jp}^2\right)\overrightarrow{\mathbf{Q}_k\mathbf{Q}_p} + \left(L_{kp}^2 - l_{kp}^2\right)\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\right] \left[\frac{\mu}{162V^4}\mathcal{A}_p^2\mathcal{K}_{jk} - \left(\frac{\Delta\lambda}{2} + \Delta\mu\right)(\mathbf{a}_0.\boldsymbol{\alpha}_p)^2(\mathbf{a}_0.\boldsymbol{\alpha}_j)(\mathbf{a}_0.\boldsymbol{\alpha}_k)\right] + \sum_{\substack{\text{Edge pair}\\ j\text{–}p,j\text{–}k}} \left(L_{jk}^2 - l_{jk}^2\right)\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\left[\frac{\mu}{162V^4}\mathcal{A}_j^2\mathcal{K}_{kp} - \left(\frac{\Delta\lambda}{2} + \Delta\mu\right)(\mathbf{a}_0.\boldsymbol{\alpha}_j)^2(\mathbf{a}_0.\boldsymbol{\alpha}_k)(\mathbf{a}_0.\boldsymbol{\alpha}_p)\right] \tag{C.6}$$

and

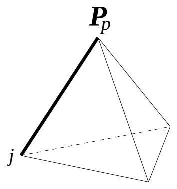
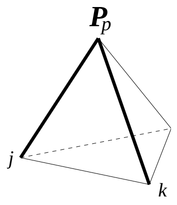
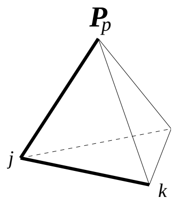
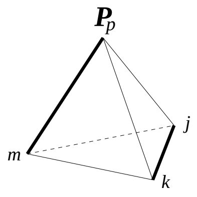
$$\frac{\partial\, T4}{\partial \mathbf{Q}_p} = \sum_{\substack{\text{Edge pair}\\ j\text{–}k,m\text{–}p}} \left(L_{jk}^2 - l_{jk}^2\right)\overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}\left[\frac{\mu}{162V^4}\left(\mathcal{K}_{jp}\mathcal{K}_{km} + \mathcal{K}_{jm}\mathcal{K}_{kp} - \mathcal{K}_{jk}\mathcal{K}_{mp}\right) + \left(\frac{\Delta\lambda}{2} + \Delta\mu\right)(\mathbf{a}_0.\boldsymbol{\alpha}_j)(\mathbf{a}_0.\boldsymbol{\alpha}_k)(\mathbf{a}_0.\boldsymbol{\alpha}_m)(\mathbf{a}_0.\boldsymbol{\alpha}_p)\right]. \tag{C.7}$$

Combining all terms $\frac{\partial\, T1}{\partial \mathbf{Q}_p}$, $\frac{\partial\, T2}{\partial \mathbf{Q}_p}$, $\frac{\partial\, T3}{\partial \mathbf{Q}_p}$ and $\frac{\partial\, T4}{\partial \mathbf{Q}_p}$ yields the equation for the non-linear anisotropic force:

$$\begin{aligned}\mathbf{F}_p^a(\mathcal{T}_i) = {} & \frac{\lambda+\mu}{162V^4}\left[\sum_{\substack{\text{Edge}\\ j\text{–}k}} \left(L_{jk}^2 - l_{jk}^2\right)\mathcal{K}_{jk}\right]\left[\sum_{\substack{\text{Edge}\\ j\text{–}p}} \overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\mathcal{K}_{jp}\right] \qquad \text{(C.8)}\\ & + \sum_{\substack{\text{Edge}\\ j\text{–}p}} \left(L_{jp}^2 - l_{jp}^2\right)\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\left[\frac{\mu}{162V^4}\mathcal{A}_p^2\mathcal{A}_j^2 + \left(\frac{\Delta\lambda}{2} + \Delta\mu\right)(\mathbf{a}_0.\boldsymbol{\alpha}_j)^2(\mathbf{a}_0.\boldsymbol{\alpha}_p)^2\right]\\ & - \sum_{\substack{\text{Edge pair}\\ j\text{–}p,k\text{–}p}} \left[\left(L_{jp}^2 - l_{jp}^2\right)\overrightarrow{\mathbf{Q}_k\mathbf{Q}_p} + \left(L_{kp}^2 - l_{kp}^2\right)\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\right]\\ & \qquad \left[\frac{\mu}{162V^4}\mathcal{A}_p^2\mathcal{K}_{jk} - \left(\frac{\Delta\lambda}{2} + \Delta\mu\right)(\mathbf{a}_0.\boldsymbol{\alpha}_p)^2(\mathbf{a}_0.\boldsymbol{\alpha}_j)(\mathbf{a}_0.\boldsymbol{\alpha}_k)\right]\\ & - \sum_{\substack{\text{Edge pair}\\ j\text{–}p,j\text{–}k}} \left(L_{jk}^2 - l_{jk}^2\right)\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\left[\frac{\mu}{162V^4}\mathcal{A}_j^2\mathcal{K}_{kp} - \left(\frac{\Delta\lambda}{2} + \Delta\mu\right)(\mathbf{a}_0.\boldsymbol{\alpha}_j)^2(\mathbf{a}_0.\boldsymbol{\alpha}_k)(\mathbf{a}_0.\boldsymbol{\alpha}_p)\right]\\ & + \sum_{\substack{\text{Edge pair}\\ j\text{–}k,m\text{–}p}} \left(L_{jk}^2 - l_{jk}^2\right)\overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}\left[\frac{\mu}{162V^4}\left(\mathcal{K}_{jp}\mathcal{K}_{km} + \mathcal{K}_{jm}\mathcal{K}_{kp} - \mathcal{K}_{jk}\mathcal{K}_{mp}\right) + \right.\\ & \qquad \left.\left(\frac{\Delta\lambda}{2} + \Delta\mu\right)(\mathbf{a}_0.\boldsymbol{\alpha}_j)(\mathbf{a}_0.\boldsymbol{\alpha}_k)(\mathbf{a}_0.\boldsymbol{\alpha}_m)(\mathbf{a}_0.\boldsymbol{\alpha}_p)\right].\end{aligned}$$

The above expression can be cast in a form similar to the expression for the non-linear isotropic force in Eqn 19. To proceed we observe that the summations in Eqn C.8 generate the edge terms listed in Table C.1. The edge terms arising from a summation of the form (*cf* lines 3

Table C.1: Edge terms generated by summations

| Summation | Interpretation | No. of terms |
|---|---|---|
| $\sum\limits_{\substack{\text{Edge}\\ j\text{–}p}}$ |  | 3 |
| $\sum\limits_{\substack{\text{Edge pair}\\ j\text{–}p,k\text{–}p}}$ |  | 3 |
| $\sum\limits_{\substack{\text{Edge pair}\\ j\text{–}p,j\text{–}k}}$ |  | 6 $j$ and $k$ may be interchanged |
| $\sum\limits_{\substack{\text{Edge pair}\\ j\text{–}k,m\text{–}p}}$ |  | 3 |

and 4, Eqn C.8):

$$\sum_{\substack{\text{Edge pair}\\ j\text{–}p,k\text{–}p}} \left[\left(L_{jp}^2 - l_{jp}^2\right)\overrightarrow{\mathbf{Q}_k\mathbf{Q}_p} + \left(L_{kp}^2 - l_{kp}^2\right)\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\right]\mathcal{K}_{jk} \tag{C.9}$$

can be more conveniently expressed as:

$$\sum_{\substack{\text{Edge}\\ j\text{–}p}} \left(L_{jp}^2 - l_{jp}^2\right)\left[\overrightarrow{\mathbf{Q}_k\mathbf{Q}_p}\mathcal{K}_{jk} + \overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}\mathcal{K}_{jm}\right] \tag{C.10}$$

where the indices $k$ and $m$ are complementary to indices $j$ and $p$. When the summation in lines 3 and 4 of Eqn C.8 is cast in the latter form, it can be combined with the summation in line 2 and the result is:

$$\frac{\mu}{162V^4}\mathcal{A}_p^2\sum_{\substack{\text{Edge}\\ j\text{–}p}} \left(L_{jp}^2 - l_{jp}^2\right)\left[\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\mathcal{A}_j^2 - \overrightarrow{\mathbf{Q}_k\mathbf{Q}_p}\mathcal{K}_{jk} - \overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}\mathcal{K}_{jm}\right] +$$
$$\left(\frac{\Delta\lambda}{2} + \Delta\mu\right)(\mathbf{a}_0.\boldsymbol{\alpha}_p)^2\sum_{\substack{\text{Edge}\\ j\text{–}p}}(\mathbf{a}_0.\boldsymbol{\alpha}_j)\left(L_{jp}^2 - l_{jp}^2\right)\left[\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_j) + \overrightarrow{\mathbf{Q}_k\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_k) + \overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_m)\right]. \tag{$T5$}$$

The edge terms arising from a summation of the form (*cf* line 5, Eqn C.8):

$$\sum_{\substack{\text{Edge pair}\\ j\text{–}p,j\text{–}k}} \left(L_{jk}^2 - l_{jk}^2\right)\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\mathcal{A}_j^2\mathcal{K}_{kp} \tag{C.11}$$

can be more conveniently expressed as:

$$\sum_{\substack{\text{Edge}\\ j\text{–}k\\ j,k\neq p}} \left(L_{jk}^2 - l_{jk}^2\right)\left[\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\mathcal{A}_j^2\mathcal{K}_{kp} + \overrightarrow{\mathbf{Q}_k\mathbf{Q}_p}\mathcal{A}_k^2\mathcal{K}_{jp}\right]. \tag{C.12}$$

The edge terms arising from a summation of the form (*cf* lines 6 and 7, Eqn C.8):

$$\sum_{\substack{\text{Edge pair}\\ j\text{–}k,m\text{–}p}} \left(L_{jk}^2 - l_{jk}^2\right)\overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}\left(\mathcal{K}_{jp}\mathcal{K}_{km} + \mathcal{K}_{jm}\mathcal{K}_{kp} - \mathcal{K}_{jk}\mathcal{K}_{mp}\right) \tag{C.13}$$

can be more conveniently expressed as:

$$\sum_{\substack{\text{Edge}\\ j\text{–}k\\ j,k\neq p}} \left(L_{jk}^2 - l_{jk}^2\right)\overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}\left(\mathcal{K}_{jp}\mathcal{K}_{km} + \mathcal{K}_{jm}\mathcal{K}_{kp} - \mathcal{K}_{jk}\mathcal{K}_{mp}\right) \tag{C.14}$$

where index $m$ is complementary to indices $j$, $k$ and $p$. Using the above results, we can combine the two summations appearing in lines 5, 6 and 7 of Eqn C.8. The result is:

$$
\begin{aligned}
&\frac{\mu}{162V^4}\sum_{\substack{\text{Edge}\\ j\text{–}k\\ j,k\neq p}}\left(L_{jk}^2-l_{jk}^2\right)\left[\overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}\mathcal{K}_{jkmp}-\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\mathcal{A}_j^2\mathcal{K}_{kp}-\overrightarrow{\mathbf{Q}_k\mathbf{Q}_p}\mathcal{A}_k^2\mathcal{K}_{jp}\right]+\\
&\left(\frac{\Delta\lambda}{2}+\Delta\mu\right)(\mathbf{a}_0.\boldsymbol{\alpha}_p)\sum_{\substack{\text{Edge}\\ j\text{–}k\\ j,k\neq p}}\left(L_{jk}^2-l_{jk}^2\right)(\mathbf{a}_0.\boldsymbol{\alpha}_j)(\mathbf{a}_0.\boldsymbol{\alpha}_k)\\
&\qquad\qquad\left[\overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_m)+\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_j)+\overrightarrow{\mathbf{Q}_k\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_k)\right],
\end{aligned}
\tag{$T6$}
$$

where $\mathcal{K}_{jkmp}$ is defined in Eqn 21. The final expression for the elastic force is obtained by combining the (unmodified) summation in line 1 of Eqn C.8 and terms $T5$ and $T6$:

$$
\begin{aligned}
\mathbf{F}_p(\mathcal{T}_i)=\;&\frac{\lambda+\mu}{162V^4}\left[\sum_{\substack{\text{Edge}\\ j\text{–}k}}\left(L_{jk}^2-l_{jk}^2\right)\mathcal{K}_{jk}\right]\left[\sum_{\substack{\text{Edge}\\ j\text{–}p}}\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\mathcal{K}_{jp}\right]\\
+\;&\frac{\mu}{162V^4}\mathcal{A}_p^2\sum_{\substack{\text{Edge}\\ j\text{–}p}}\left(L_{jp}^2-l_{jp}^2\right)\left[\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\mathcal{A}_j^2-\overrightarrow{\mathbf{Q}_k\mathbf{Q}_p}\mathcal{K}_{jk}-\overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}\mathcal{K}_{jm}\right]\\
+\;&\left(\frac{\Delta\lambda}{2}+\Delta\mu\right)(\mathbf{a}_0.\boldsymbol{\alpha}_p)^2\sum_{\substack{\text{Edge}\\ j\text{–}p}}(\mathbf{a}_0.\boldsymbol{\alpha}_j)\left(L_{jp}^2-l_{jp}^2\right)\\
&\qquad\left[\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_j)+\overrightarrow{\mathbf{Q}_k\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_k)+\overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_m)\right]\\
+\;&\frac{\mu}{162V^4}\sum_{\substack{\text{Edge}\\ j\text{–}k\\ j,k\neq p}}\left(L_{jk}^2-l_{jk}^2\right)\left[\overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}\mathcal{K}_{jkmp}-\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}\mathcal{A}_j^2\mathcal{K}_{kp}-\overrightarrow{\mathbf{Q}_k\mathbf{Q}_p}\mathcal{A}_k^2\mathcal{K}_{jp}\right]\\
+\;&\left(\frac{\Delta\lambda}{2}+\Delta\mu\right)(\mathbf{a}_0.\boldsymbol{\alpha}_p)\sum_{\substack{\text{Edge}\\ j\text{–}k\\ j,k\neq p}}\left(L_{jk}^2-l_{jk}^2\right)(\mathbf{a}_0.\boldsymbol{\alpha}_j)(\mathbf{a}_0.\boldsymbol{\alpha}_k)\\
&\qquad\left[\overrightarrow{\mathbf{Q}_m\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_m)+\overrightarrow{\mathbf{Q}_j\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_j)+\overrightarrow{\mathbf{Q}_k\mathbf{Q}_p}(\mathbf{a}_0.\boldsymbol{\alpha}_k)\right].
\end{aligned}
\tag{C.15}
$$

## D Implicit Runge-Kutta implementation details

The Newtonian scheme in Eqn 10 can be integrated using a variety of solvers. For stability purposes we use a 3-stage, 5th order Radau IIA implicit Runge-Kutta algorithm with an adaptive step size. The implementation details provided in this Appendix follow the development set out in [9], with some modifications as noted below.

An $s$-stage implicit Runge-Kutta algorithm for propagating $\mathbf{y}$ from $t_0 \to t_1$ has the general form:

$$\mathbf{g}_i = \mathbf{y}_{t0} + h\sum_{j=1}^{s} a_{ij}\mathbf{f}(x_o + c_j h, \mathbf{g}_j) \quad i = 1, ..., s \tag{D.1a}$$

$$\mathbf{y}_{t1} = \mathbf{y}_{t0} + h\sum_{j=1}^{s} b_j\mathbf{f}(x_o + c_j h, \mathbf{g}_j) \tag{D.1b}$$

The $\mathbf{g}_j$ are the intermediate stages to be determined, and once known, Eqn D.1b provides an explicit formula for $\mathbf{y}_{t1}$.

The Radau IIA method is a collocation method based on Radua quadrature with *Butcher* tableau:

$$\begin{array}{c|ccc}
\frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\
\frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\
1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\
\hline
 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9}
\end{array}$$

where the $c_i$ are in the leftmost column, the $b_j$ are in the bottom row, and the remaining terms are the $a_{ij}$.

To reduce numerical error we work with $z_i$:

$$\mathbf{z}_i = \mathbf{g}_i - \mathbf{y}_{t0} \tag{D.2}$$

and consequently Eqn D.1a becomes:

$$\mathbf{z}_i = h\sum_{j=1}^{s} a_{ij}\mathbf{f}(x_o + c_j h, \mathbf{y}_{t0} + \mathbf{z}_j). \tag{D.3}$$

If Eqn D.3 is rewritten as:

$$\begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_s \end{bmatrix} = A \begin{bmatrix} h\mathbf{f}(x_0 + c_1 h, \mathbf{y}_{t0} + \mathbf{z}_1) \\ \vdots \\ h\mathbf{f}(x_0 + c_s h, \mathbf{y}_{t0} + \mathbf{z}_s) \end{bmatrix} \tag{D.4}$$

we can see that Eqn D.1b is equivalent to:

$$\mathbf{y}_{t1} = \mathbf{y}_{t0} + h \sum_{i=1}^{s} d_i \mathbf{z}_i \tag{D.5}$$

where

$$(d_1, \ldots, d_s) = (b_1, \ldots, b_s)\, A^{-1}. \tag{D.6}$$

If $A$ is the matrix of $a_{ij}$ coefficients in the *Butcher* tableau above, it is non-singular and $\mathbf{d} = (0, 0, 1)$. Consequently, $\mathbf{y}_{t1} = \mathbf{y}_{t0} + \mathbf{z}_3$, and the challenge now is how to compute the $\mathbf{z}_i, i = 1, \ldots, s$. To proceed, we use the following:

1. for some non-linear equation $\mathbf{g}(\mathbf{z}) = \mathbf{0}$, the $k$-th Newton iteration estimate for $\mathbf{z}$ is given by:

$$\mathbf{z}^{k+1} = \mathbf{z}^k - \left( \frac{\partial \mathbf{g}}{\partial \mathbf{z}} \left(\mathbf{z}^k\right) \right)^{-1} \mathbf{g}(\mathbf{z}^k), \quad k = 0, 1, \ldots \tag{D.7}$$

2. to propagate some arbitrary (possibly non-linear) system $\mathbf{z}' = \mathbf{f}(\mathbf{z})$ over the interval $[\mathbf{x}_{n-1}, \mathbf{x}_n]$, we can use implicit differencing:

$$\mathbf{z}_n = \mathbf{z}_{n-1} + h\mathbf{f}(\mathbf{z}_n). \tag{D.8}$$

The implicit system (Eqn D.8) can be solved using Newton iteration. Setting:

$$\mathbf{g}(\mathbf{z}_n) = \mathbf{z}_n - \mathbf{z}_{n-1} - h\mathbf{f}(x_n, \mathbf{z}_n) = \mathbf{0}. \tag{D.9}$$

and using the result in Eqn D.7 to solve for $\mathbf{z}_n$ yields:

$$\begin{aligned} \mathbf{z}_n^{k+1} &= \mathbf{z}_n^k - \left( \frac{\partial \mathbf{g}}{\partial \mathbf{z}} \right)^{-1} \mathbf{g}(\mathbf{z}_n^k) \qquad \text{(D.10)} \\ &= \mathbf{z}_n^k - \left( I - h \frac{\partial \mathbf{f}}{\partial \mathbf{z}} \right)^{-1} \left( \mathbf{z}_n^k - \mathbf{z}_{n-1} - h\mathbf{f}(x_n, \mathbf{z}_n^k) \right), \quad k = 0, 1, \ldots \end{aligned}$$

Rearranging we obtain:

$$\left( I - h \frac{\partial \mathbf{f}}{\partial \mathbf{z}} \right) \left( \mathbf{z}_n^{k+1} - \mathbf{z}_n^k \right) = -\mathbf{z}_n^k + \mathbf{z}_{n-1} + h\mathbf{f}(x_n, \mathbf{z}_n^k). \tag{D.11}$$

Returning to our original system, we now use the result in Eqn D.11 to propagate forward with timesteps $c_1 h$, $c_2 h$,... $c_s h$, to obtain the collocation points $\mathbf{z}_1, \mathbf{z}_2, \dots \mathbf{z}_s$. In matrix notation we have:

$$\begin{bmatrix} I - ha_{11}J_1 & -ha_{12}J_2 & \dots & -ha_{1s}J_s \\ -ha_{21}J_1 & I - ha_{22}J_2 & \dots & -ha_{2s}J_s \\ \vdots & \vdots & \ddots & \vdots \\ -ha_{s1}J_1 & I - ha_{s2}J_2 & \dots & -ha_{ss}J_s \end{bmatrix} \begin{bmatrix} \mathbf{z}_1^{k+1} - \mathbf{z}_1^k \\ \mathbf{z}_2^{k+1} - \mathbf{z}_2^k \\ \vdots \\ \mathbf{z}_s^{k+1} - \mathbf{z}_s^k \end{bmatrix} = - \begin{bmatrix} \mathbf{z}_1^k - \mathbf{z}_0 - h \sum\limits_{j=1}^{s} a_{1j} \mathbf{f}\left(x_0 + c_1 h, \mathbf{y}_{t0} + \mathbf{z}_j^k\right) \\ \mathbf{z}_2^k - \mathbf{z}_0 - h \sum\limits_{j=1}^{s} a_{2j} \mathbf{f}\left(x_0 + c_2 h, \mathbf{y}_{t0} + \mathbf{z}_j^k\right) \\ \vdots \\ \mathbf{z}_s^k - \mathbf{z}_0 - h \sum\limits_{j=1}^{s} a_{sj} \mathbf{f}\left(x_0 + c_s h, \mathbf{y}_{t0} + \mathbf{z}_j^k\right) \end{bmatrix} \tag{D.12}$$

with Jacobians $J_i = \frac{\partial f}{\partial y}\left(x_0 + c_i h, \mathbf{y}_{t0} + \mathbf{z}_i\right)$. More succinctly:

$$\left(I - hA \otimes J\right) \Delta \mathbf{Z}^k = -\mathbf{Z}^k + h\left(A \otimes I\right) \mathbf{F}(\mathbf{Z}^k) \tag{D.13}$$

where $\otimes$ is the Kronecker or direct product of two matrices, and $\mathbf{z}_0 = \mathbf{0}$. Eqn D.13 is the expression we are seeking: by solving for $\Delta\mathbf{Z}^k$ we obtain $\mathbf{Z}^{k+1} = \mathbf{Z}^k + \Delta\mathbf{Z}^k$, enabling us to iteratively compute an approximation for $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_s)$. Given the $\mathbf{z}_i$ we can then compute $\mathbf{y}_{t1}$ via Eqn D.5.

The linear system in Eqn D.13 has dimension $n.s$, where $n$ is the dimension of $\mathbf{y}$, and each Newton iteration requires $s$ evaluations of $f$. To reduce the numerical cost we replace the $J_i$ with an approximation:

$$J \approx \frac{\partial \mathbf{f}}{\partial \mathbf{y}}(x_0, \mathbf{y}_{t0}) \tag{D.14}$$

so the matrix $(I - hA \otimes J)$ is the same for all Newton iterations. The approximate Jacobian does not affect overall accuracy providing the Newton iteration can still converge [10].

To further improve computational efficiency we reduce the $3n$ dimensional system $(s = 3)$ to a pure-real system of dimension $n$ and a complex system also of dimension $n$. To proceed we note the $s \times s$ matrix $A^{-1}$ has eigenvalues $\hat{\gamma}, \hat{\alpha} \pm i\hat{\beta}$ and choose a transformation $T$ such that:

$$T^{-1} A^{-1} T = \Lambda \tag{D.15}$$

is diagonal, block diagonal, triangular or in Jordan canonical form. With the transformation $W^k = \left(T^{-1} \otimes I\right) Z^k$, Eqn D.13 becomes:

$$\begin{aligned} \left(h^{-1}\Lambda \otimes I - I \otimes J\right) \Delta \mathbf{W}^k &= -h^{-1}\left(\Lambda \otimes I\right) \mathbf{W}^k + \left(T^{-1} \otimes I\right) \mathbf{F}\left((T \otimes I)\mathbf{W}^k\right) \\ \mathbf{W}^{k+1} &= \mathbf{W}^k + \Delta \mathbf{W}^k. \end{aligned} \tag{D.16}$$

The matrix $(h^{-1}\Lambda \otimes I - I \otimes J)$ is block diagonal:

$$\left[\begin{array}{c:cc} \gamma I - J & 0 & 0 \\ \hdashline 0 & \alpha I - J & -\beta I \\ 0 & \beta I & \alpha I - J \end{array}\right] \tag{D.17}$$

with $\gamma = \frac{\hat{\gamma}}{h}$ and $\beta = \frac{\hat{\beta}}{h}$, enabling the $3n$ dimensional system to be split into two sub-systems: one of dimension $n$ and the other of dimension $2n$. The $2n$ dimensional system can be reduced to an $n$-dimensional complex system of the form $((\alpha + i\beta)I - J)$. Hairer *et al* [9] recommend using $LU$ decomposition to solve Eqn D.16 for $\Delta\mathbf{W}^k$, however in our case we can exploit the special structure of $J$ and use a simple back substitution scheme. For further details, see Appendix E.

Given the preceeding development, we now summarise the 3-stage ($s = 3$) algorithm as follows:

**Step 0.** Initialise

- Compute the initial step size $h$. Note: *multiple Runge-Kutta steps may be required to propagate over the interval $x_0 \to x_0 + \Delta t$. The step size for each individual Runge-Kutta step ($h$) is adjusted at the end of each step according to the propagation error, to minimise the total number of Runge-Kutta steps required.*
- Compute the derivatives at the start of the timestep, $\mathbf{F}(x_0, \mathbf{y}_{t0})$:

$$\begin{aligned} \frac{d\mathbf{p}_i}{dt} &= \mathbf{v}_i, \qquad i = 1, \ldots, n \\ \frac{d\mathbf{v}_i}{dt} &= \mathcal{F}_i - \frac{\gamma_i \mathbf{v}_i}{m_i} \end{aligned} \tag{D.18}$$

where for each node $i$, $\mathbf{p}_i$ is the position, $\mathbf{v}_i$ is the velocity, $\mathcal{F}_i$ is the force (obtained from Eqn 31), $\gamma_i$ is the damping coefficient and $m_i$ is the mass.

**Step 1.** Main loop (a series of small time steps $h$ until the total $\Delta t$ is reached)

*For each Runge-Kutta step of size $h$:*

- Compute the starting values for the Newton iteration:

$$\mathbf{z}_i^0 = \begin{cases} \mathbf{0} & : \quad k = 0 \\ \mathbf{q}(1 + wc_i) + \mathbf{y}_{t0} - \mathbf{y}_{t1} & : \quad k > 0 \end{cases} \tag{D.19}$$

for $i = 1, \ldots, s$, where $q$ is the Newton form of the interpolating polynomial (see below) and $w = \frac{h_{prev}}{h}$ is the ratio of the previous and current time steps.

- Compute $\mathbf{W}^0 = (T^{-1} \otimes I)\,\mathbf{Z}^0$
- Newton iteration, $k = 0, \ldots$ (to iteratively compute the intermediate points, $\mathbf{z}_i, i = 1, \ldots, 3$)
  1. Compute the intermediate points, $\mathbf{y}_{t0} + \mathbf{z}_i, i = 1, \ldots, 3$
  2. Evaluate the derivatives at the intermediate points:

$$\begin{aligned}\left\{\mathbf{f}\left(x_0 + c_1 h, \mathbf{y}_{t0} + \mathbf{z}_1^k\right), \mathbf{f}\left(x_0 + c_2 h, \mathbf{y}_{t0} + \mathbf{z}_2^k\right), \mathbf{f}\left(x_0 + c_3 h, \mathbf{y}_{t0} + \mathbf{z}_3^k\right)\right\} &= \mathbf{F}\left(\mathbf{Z}^k\right) \quad \text{(D.20)} \\ &= \mathbf{F}\left((T \otimes I)\mathbf{W}^k\right)\end{aligned}$$

  3. Solve Eqn D.16 for $\Delta\mathbf{W}^k$ using back substituion (Appendix E).
  4. Compute the $k + 1^{th}$ estimate, $\mathbf{Z}^{k+1} = (T \otimes I)\,\mathbf{W}^{k+1}$.

  Continue until the Newton iteration has converged. See [9] for the convergence criteria and scheme for adaptively modifying $h$.
- Estimate the extrapolation error:

$$\boldsymbol{\epsilon} = \left(I - \frac{h}{\hat{\gamma}} J\right)^{-1} (\hat{\mathbf{y}}_{t1} - \mathbf{y}_{t1}) \qquad \text{(D.21)}$$

  where

$$\hat{\mathbf{y}}_{t1} - \mathbf{y}_{t1} = \frac{h}{\hat{\gamma}} \mathbf{f}(x_0, \mathbf{y}_{t0}) + e_1\mathbf{z}_1 + e_2\mathbf{z}_2 + e_3\mathbf{z}_3 \qquad \text{(D.22)}$$

  and $(e_1, e_2, e_3) = \frac{1}{3\hat{\gamma}}\left(-13 - 7\sqrt{6}, -13 + 7\sqrt{6}, -1\right)$. We use simple back substitution to solve Eqn D.21 for the error, $\boldsymbol{\epsilon}$. If the error norm exceeds a given threshold, the Runge-Kutta step is rejected and restarted using a smaller step size $h$. Again, see [9] for the algorithms used in dynamically adjusting the step size.
- If the step is accepted:
  1. Compute $\mathbf{y}_{t1}$ using Eqn D.5 (given $\mathbf{d} = (0, 0, 1)$, only $\mathbf{z}_3$ is required).
  2. Compute the derivatives $\mathbf{F}\left(x_0 + h, \mathbf{y}_{t1}\right)$ so they are ready for use in the next step. See Eqn D.18.
  3. Compute the Newton coefficients used for constructing the initial estimate $\mathbf{z}_i^0$ in the next step:

$$\mathbf{z}_i^0 = \mathbf{y}_{t0} - \mathbf{y}_{t1} + \mathbf{q}(1 + wc_i), \qquad i = 1, \ldots 3 \qquad \text{(D.23)}$$

  with interpolation points $\mathbf{q}(0) = \mathbf{0}$ and $\mathbf{q}_i(c_i) = \mathbf{z}_i, i = 1, \ldots 3$. Adapting the general Newton form:

$$\begin{aligned} q(x) &= [z_0] + [z_0, z_1](x - c_1) + [z_0, z_1, z_2](x - c_1)(x - c_2) \\ &+ [z_0, z_1, z_2, z_3](x - c_1)(x - c_2)(x - c_3) \end{aligned} \tag{D.24}$$

we have:

$$\begin{aligned} \mathbf{q}(1 + wc_i) &= [\mathbf{z}_0] + [\mathbf{z}_0, \mathbf{z}_1]\,(1 + wc_i - c_1) \\ &+ [\mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2]\,(1 + wc_i - c_1)(1 + wc_i - c_2) \\ &+ [\mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3]\,(1 + wc_i - c_1)(1 + wc_i - c_2)(1 + wc_i - c_3) \end{aligned} \tag{D.25}$$

with coefficients:

$$\begin{aligned} [\mathbf{z}_0] &= \mathbf{0} \\ [\mathbf{z}_0, \mathbf{z}_1] &= \frac{\mathbf{z}_1 - \mathbf{0}}{c_1 - 0} \\ [\mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2] &= \frac{\frac{\mathbf{z}_2 - \mathbf{z}_1}{c_2 - c_1} - \frac{\mathbf{z}_1}{c_1}}{c_2 - 0} \\ [\mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3] &= \frac{\frac{\frac{\mathbf{z}_3 - \mathbf{z}_2}{c_3 - c_2} - \frac{\mathbf{z}_2 - \mathbf{z}_1}{c_2 - c_1}}{c_3 - 1} - \frac{\frac{\mathbf{z}_2 - \mathbf{z}_1}{c_2 - c_1} - \frac{\mathbf{z}_1 - \mathbf{0}}{c_1 - c_0}}{c_2 - 0}}{c_3 - 0}. \end{aligned} \tag{D.26}$$

## E Structure of the Jacobian, $J$

The Jacobian can be constucted either numerically, using difference approximations, or analytically. The numerical approximation is simple to compute, but numerically expensive and consequently not suited for real time applications where rapid evaluation is critical. To investigate the analytical construction we consider the fundamental equations for the $x$-component of node $i$:

$$\begin{aligned} p'_{ix} &= v_{ix} \\ v'_{ix} &= \frac{F_{ix}}{m} - \frac{\gamma v_{ix}}{m} \end{aligned} \tag{E.1}$$

where $p$ is position, $v$ is velocity, $m$ is mass, $\gamma$ is the damping and $F$ is the force computed via Eqn 31. The elements of the Jacobian are then:

$$\frac{\partial p'_{ix}}{\partial p_{ix}} = 0 \qquad \frac{\partial p'_{ix}}{\partial v_{ix}} = 1$$

$$\frac{\partial v'_{ix}}{\partial p_{ix}} = \frac{1}{m}\frac{\partial F_{ix}}{\partial p_{ix}} \qquad \frac{\partial v'_{ix}}{\partial v_{ix}} = -\frac{\gamma}{m}$$

There are 3 position ($x$, $y$, $z$) components and 3 velocity ($x$, $y$, $z$) components for each node, so $J$ is a $6n \times 6n$ matrix. If the force terms terms $\frac{1}{m}\frac{\partial F}{\partial p}$ were omitted, $J$ would be very sparse and Eqns D.16 and D.21 could be solved rapidly by back substition.

The force terms $\frac{1}{m}\frac{\partial F_{i;x,y,z}}{\partial p_{j;x,y,z}}$ will be non-zero for any nodes $i$ and $j$ that are connected. Consequently $J$ will not be sparse and full $LU$ decomposition will be required to solve Eqns D.16 and D.21. Numerical testing has shown that the Newton iteration will converge if the force terms can be omitted, albeit with a performance penalty. Figures 5 shows the magnitude of the components of J for a simple $4 \times 4 \times 4$ grid consisting of 64 nodes and 135 tetrahedra. The nodes on the bottom row of the grid are constrained, leaving 48 free nodes and the Jacobian has dimension $288 \times 288$ (6 terms for each node). The quantity shown in the figure is $sign\left(J_{ij}\right).log\left(|J_{ij}|\right)$. The components of the Jacobian computed using difference approximations are shown in green (positive values) and red (negative values). The components of the simplified analytical Jacobian (*i.e.* force terms omitted) are shown in blue.

The convergence properties of the Runge-Kutta propagation was tested using the numerical and analytical methods for computing the Jacobian. The tests were conducted on a $4 \times 4 \times 4$

Table E.1: Comparison of convergence rates with alternative methods for constructing $J$

| Method | Mean time step, $h$ (s) | No. of R-K steps[1] | No. of $J$ evaluations[2] | No. of Newton iterations[3] |
|---|---|---|---|---|
| Numerical | 0.0200 | 2.5 | 2.4 | 7.7 |
| Analytical | 0.0336 | 1.5 | 1.5 | 4.0 |
| Simplified analytical | 0.0052 | 9.5 | Nil | 49.6 |

Notes:

1. Number of Runge-Kutta steps of size $h$ required to propagate over the full time interval $\Delta t = 0.05$ seconds.
2. Average number of times the Jacobian is evaluated for each propagation step, $\Delta t$. The Jacobian is not explicitly constructed when using the simplified analytical method - the necessary terms are computed when required (during back substitution)
3. Average number of Newton iterations required for each propagation step, $\Delta t$.

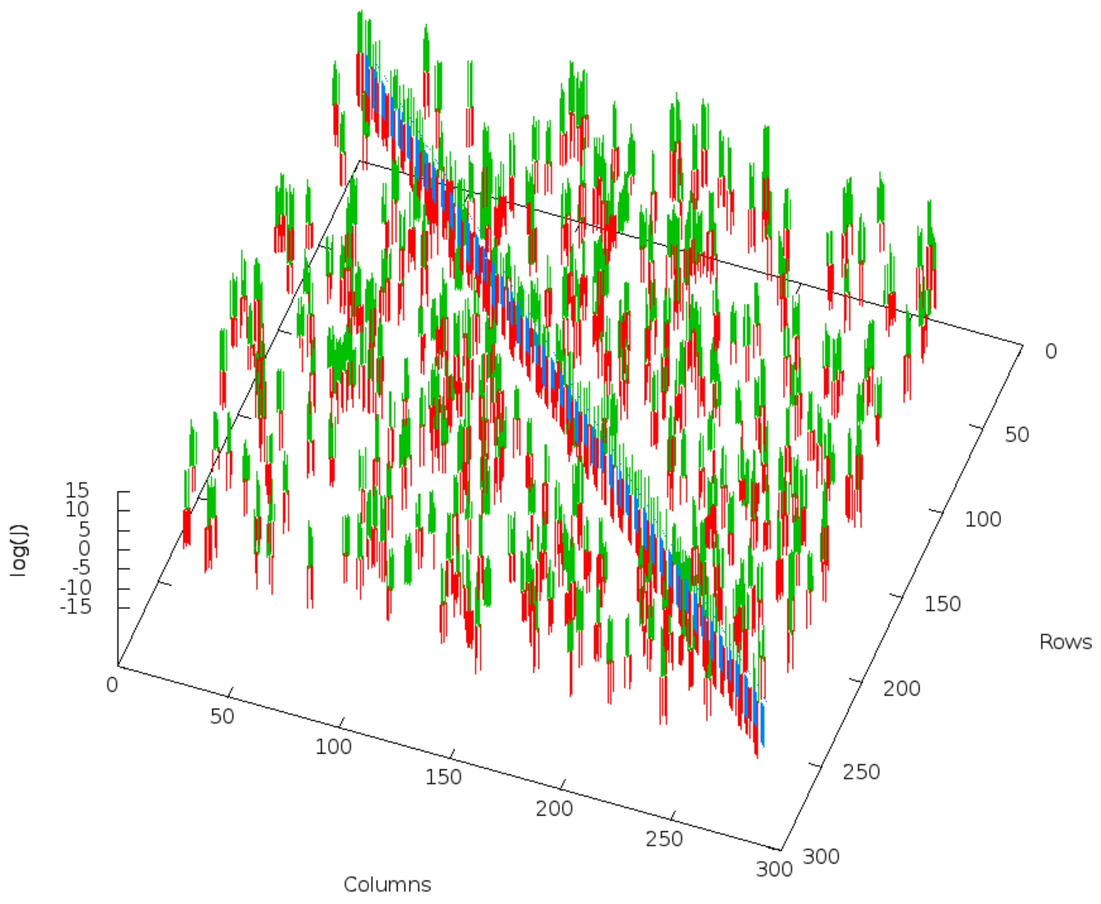


Figure 5: Components of the Jacobian computed for a $4 \times 4 \times 4$ grid. Quantity shown is $sign\left(J_{ij}\right).log\left(|J_{ij}|\right)$. Analytic evaluation of $J$ is shown in blue, and numerical evaluation shown in green (positive terms) and red (negative terms).

and the results are summarised in Table E.1. The data in Table E.1 are expressed as the average number of operations required for each call to the Runge-Kutta algorithm with a total propagation interval $\Delta t = 0.05$ seconds. The total interval $\Delta t$ may be propagated using a series of smaller Runge-Kutta timesteps, $h$. The results show that a much smaller step size is required when the force terms are ommitted. In summary:

- the simplified analytical method is very fast as $J$ does not need to be constructed, but small step sizes $h$ are required to compensate for the omission of the force terms.

- a larger step size $h$ can be obtained when using the numerical method (compared to the simplified analytical method), however each step takes longer to compute due to the cost of: (i) evaluating $J$, and (ii) the $LU$ decomposition of $J$.

- the largest step size $h$ can be obtained when using the exact Jacobian (computed using an analytical expression), reducing the number of R-K steps, but again requiring $LU$ decomposition of $J$.

All work to date has used the simplified analytical method as the numerical approach is not fast enough for real-time application. Future work will investigate whether $LU$ decomposition of the exact Jacobian (constructed from an analytical expression) can be performed in real time for large systems.