Interactive real-time physics
An intuitive approach to form-finding and structural analysis for design and education

Gennaro Senatore [a,*], Daniel Piker [b]

[a] University College London, Expedition Engineering, United Kingdom
[b] Foster + Partners, McNeel & Associates, United Kingdom

HIGHLIGHTS

- Unified physics engine for both static–dynamic load analysis, structural optimization and form finding.
- The implementation combines dynamic relaxation and co-rotational formulation.
- Interactive structural analysis as a teaching and design aid.
- Interactive structural form-finding.

ARTICLE INFO

ABSTRACT

Real-time physics simulation has been extensively used in computer games, but its potential has yet to be fully realized in design and education. We present an interactive 3D physics engine with a wide variety of applications.

In common with traditional FEM, the use of a local element stiffness matrix is retained. However, unlike typical non-linear FEM routines elements forces, moments and inertia are appropriately lumped at nodes following the dynamic relaxation method. A semi-implicit time integration scheme updates linear and angular momentum, and subsequently the local coordinate frames of the nodes. A co-rotational approach is used to compute the resultant field of displacements in global coordinates including the effect of large deformations. The results obtained compare well against established commercial software.

We demonstrate that the method presented allows the making of interactive structural models that can be used in teaching to develop an intuitive understanding of structural behaviour. We also show that the same interactive physics framework allows real-time optimization that can be used for geometric and structural design applications.

© 2014 The Authors. Published by Elsevier Ltd.
This is an open access article under the CC BY license
(http://creativecommons.org/licenses/by/3.0/).

1. Introduction

This paper describes the formulation of a physics engine which combines the principle aspects of Dynamic Relaxation [1] and those of the co-rotational formulation for the Finite Elements Method [2]. Our formulation draws on the work done by Williams [3] and shares similarities with recently published works in the literature [4]. Our contribution is to frame dynamic relaxation within the co-rotational approach in order to enable both the static and dynamic simulation of structures with geometrically non-linear behaviour (i.e. small strains, finite displacements and rotations). Such an approach aims at building a unified engine that can use linear formulations for conventional beams, shells/plates and 3D elements whilst being able to compute accurately displacement and stress fields in both static and dynamic problems.

The formulation lies at the core of software, implemented by the authors, which has been published online as a java applet (http://expeditionworkshed.org/) in 2011 and since then adopted in both design and education by several universities in the UK. A description of their applications such as geometric and structural optimization tasks and the development of interactive structural models for teaching aid are given in the last section of the paper.

* Corresponding author.
E-mail addresses: gennarosenatore@gmail.com (G. Senatore), danielpiker@gmail.com (D. Piker).
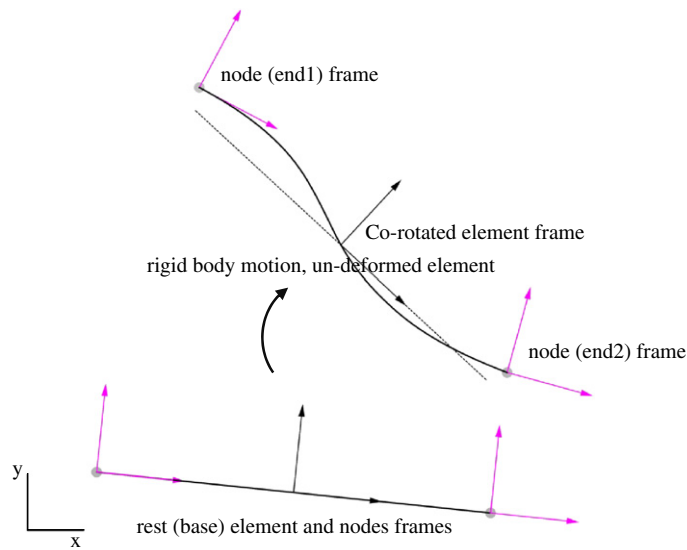
Fig. 1. Co-rotated element (beam) kinematics.

1.1. Dynamic relaxation (DR)

Computational mechanics based on particle systems have been extensively used in computer graphics for realistic simulation of rigid and deformable bodies, elastoplastic deformation, fractures and fluids [5,6]. These applications frequently make use of the Verlet integration scheme [7], a method originally developed for simulation of molecular dynamics. This scheme can be shown to be very closely linked to the method of *dynamic relaxation*, a numerical method formulated as an alternative to other relaxation techniques such as Gauss–Seidel or successive over relaxation for solving systems of partial differential equations using finite difference approximation. Newmark [8] developed a method that used implicit integration and global stiffness, mass and damping matrices for both the static and dynamic solutions of structures with both material and geometric non-linear behaviour. Day [1] implemented an explicit solution method for the analysis of portal frames and plates and Otter for the analysis of pre-stressed concrete pressure vessels [9]. Later, these methods were adapted to finite elements problems [10] finally assuming the popular formulation in which equilibrium and compatibility relations are separated and no overall stiffness matrix of the structure is assembled. Successively, it was further improved by other authors amongst with Barnes [11] and Wakefield [12].

Dynamic Relaxation is based on the discretization of the continuum into a set of concentrated masses (called particles or nodes) linked by elements (which can be one, two or three dimensional) defining how forces and moments propagate through the system. The method has been mainly used to solve static problems which are converted into dynamic problems with the use of fictitious masses and viscous damping at nodes.

Non-linear FEM routines are usually implemented using implicit Newton–Raphson methods. Compared to Newton–Raphson methods dynamic relaxation needs less memory because there is no need to store and update the global structure stiffness matrix ($N \times N$) but only the internal forces of the elements ($N \times 1$) where N is the number of the nodal degrees of freedom. Since there is only need to derive the internal force vectors and the elements linear stiffness matrix, it is easier to include the higher order terms of the strain–displacement relations for the element [13]. In addition, it can be proven that for NR implicit methods the operation count is proportional to the number of elements $O(ne^{7/3})$ while for DR is $O(ne^{4/3})$ (ne is the number of elements) thus making the latter better suited for systems with larger number of elements [14]

cited in [4]. On the other hand, DR is conditionally stable and requires pre-computation to estimate optimum parameters such as the integration time step to ensure convergence and stability.

DR has been used for the design of grid-shells and tension structures such as those built with cable or fabrics, and more generally for the static solution of structures exhibiting material and geometrical non-linearity. A good review of applications of DR on building scale projects is given in [15]. Other recent examples of structural form-finding can be found in [16,17]. On-going work that adopts a force-based approach for purely geometric related optimization applications is the software called Kangaroo developed by Piker [18].

1.2. Co-rotational formulation

The co-rotational formulation was developed for the analysis of geometrically non-linear structures under the assumption of small strains but arbitrarily large displacements and rotations [19,20]. It can be considered the most recent kinematics description after the Total Lagrangian and Updated Lagrangian which have been used extensively in the FEM analysis of structures with non-linear behaviour.

The main idea behind the co-rotational approach is to separate the motion of the element into a rigid motion and a purely deformational part. Each element has a base (also called initial or rest) reference frame and its own co-rotated frame obtained as a rigid body motion (translations and rotations) of the base frame. When rotational degrees of freedom are involved, the nodes of the element have their own rigidly attached frames which move with the element. Deformations are measured with respect to the element co-rotated frame (Fig. 1).

The node frames are effectively utilized to describe the element orientations at its ends. During load analysis, the difference between the orientations at the element's ends and the co-rotated element frame represent the rotational part of the deformation. In form-finding or geometric optimization, the node orientations can be used to measure the difference with respect to target orientations specified beforehand. Fig. 1 shows the different reference frames for a beam element.

Fig. 1 shows the kinematics for a co-rotated beam element in two dimensions; in Section 2.5 the main steps of the formulation for beam elements in 3 dimensions (6 degrees of freedom) are given; the same approach can be taken for 2D elements (i.e. plates, shells) [21] and 3D elements (i.e. tetrahedrons, bricks) [22].

Due to the fact that each element has its own co-rotated frame and that deformational displacements and rotations are measured with respect to it, for sufficiently small element sizes the small strains assumption holds while both rotations and displacements can be arbitrarily large [23].

1.3. Integration with standard FEM element libraries

Dynamic Relaxation is well suited to problems that are solved with the FEM formulation and can be implemented using standard element libraries [10].

The co-rotational formulation qualifies as "*element independent*". This definition comes from the fact that the relations between the local and global element deformational displacements and rotations are defined by a transformation matrix and consequently the stiffness and the local forces and moments depend on the choice of the local coordinate system. However, the relation between the local and global deformation is independent of the particular strain definition used [24,21]. In simpler words, the addition and removal of the rigid body motion part from translational and rotational displacements can be implemented separately from the element and the solver sub-routines. In addition, the fact that each element has its own co-rotated frame allows keeping deformations small with respect to the frame. For these reasons, standard linear or geometric-non-linear FEM libraries can be adopted immediately without the need of new implementation.

A disadvantage of the co-rotational formulation is the involvement of relatively complex mathematics when rotational degrees of freedom are involved especially considering that finite rotations are not commutative and generally non-additive. However, several parameterizations of rotations are available that can be integrated in both explicit and implicit integration stepping routines [25].

2. Methodology

2.1. A force based approach, towards a unified solver

The main idea is to "follow" the movements of the nodes caused by the out of balance forces. Elements' masses and rotational inertia are lumped at nodes. Information regarding position, velocity and acceleration of each node is computed iteratively. The system converges to an equilibrium position around which it oscillates and eventually settles when the out of balance forces and moments become very small/null. The simplest way to achieve convergence is to damp the nodal movements using artificial viscous damping.

Because each step is relatively simple to compute (and for moderate numbers of elements, many steps can be performed per second on consumer level computers) intermediate steps towards convergence correspond to actual physical dynamics, they can be shown. Thus a similar framework can be used as in an interactive physics engine, such as those used in games.

For some optimization and analysis purposes it is only the final equilibrium result which is of interest, and the intermediate steps need not be shown. In these cases the damping and mass coefficients may be chosen purely for the sake of convergence. A kinetic damping scheme [26] may also be used, which zeros the velocities whenever kinetic energy peaks are reached, achieving fast convergence, but more removed from the actual dynamic behaviour. However, if appropriate damping and mass values are set, the same system can also be used to model realistic dynamics [27].

The method can be considered as a simplified form of a standard FEM using a lumped mass matrix. It can be shown that the computation of the fundamental modes using a lumped mass matrix gives underestimated natural frequencies and vice versa for

the consistent mass matrix. By increasing the number of elements of the mesh, the computation of the natural frequencies using both lumped and consistent mass matrices converges to the same result for simple elements (2nd order) [28].

2.2. Forces equilibrium and translational motion

For discretized systems the equation of motion can be written as:

$$M_{ij} a_{ij}^t + C v_{ij}^t + K_{ij} \delta_{ij}^t = P_{ij} \tag{1}$$

$$j = (x, y, z); \qquad v = \dot{\delta}; \qquad a = \ddot{\delta}$$

where the subscript i refers to the ith node and the j to the jth direction (x, y, z) in global coordinates and the super script t is time. M is the mass of the element lumped at node and C is the damping factor. K_{ij} and P_{ij} are the stiffness and the external loads respectively of the ith node in the jth direction. The total translational displacement, velocity and acceleration of the ith node in the jth direction j at time step t are represented by the symbols δ_{ij}^t, v_{ij}^t and a_{ij}^t. By defining the residuals as out of balance forces F_{ij}^t resulting from the difference between the internal forces $K_{ij} \delta_{ij}^t$ and the external applied load vector P_{ij}, Eq. (1) can be rewritten as:

$$F_{ij}^t = M_{ij} a_{ij}^t + C v_{ij}^t. \tag{2}$$

The velocities and position of the nodes can be obtained by an explicit, implicit or semi-implicit time step integration scheme. Using Newton's 2nd law and a simple modification of the forward Euler integration scheme, it is possible to obtain a simple yet very stable and 1st order accurate integration scheme called semi-implicit or symplectic Euler:

$$v_{ij}^{t+\Delta t} = v_{ij}^t + \Delta t \frac{F_{ij}^t}{M_i} \tag{3}$$

$$r_{ij}^{t+\Delta t} = r_{ij}^t + \Delta t\, v_{ij}^{t+\Delta t} \tag{4}$$

where Δt is the time step and r_{ij}^t is the jth coordinate position of the ith node. The integration scheme is semi-implicit because it uses forward Euler to compute the velocity but backward Euler to obtain the position using the velocity at $t + \Delta t$. For forces that do not depend on velocity (non-dissipative) the symplectic Euler is 2nd order accurate like the central difference or Stoermer–Verlet with global error $O(\Delta t^2)$. Energy preservation features of the symplectic Euler and comparison with the velocity-Verlet scheme are discussed in Section 2.4. Eqs. (2)–(4) show that the velocity and position of the ith node at next time step only depend on its own previous time step velocity and residuals. The system is decoupled meaning that the update of a node position at the next time step does not perturb the updates of the others.

Once the updated nodal positions are computed, it is then possible to separate the rigid from the deformational part of the displacements to obtain the residuals at next time step:

$$F_{ij}^{t+\Delta t} = P_{ij} - \sum_m K_{ijm} \delta_{ijm}^{t+\Delta t} \tag{5}$$

where m is the number of elements connecting to the ith node. Equilibrium of forces Eq. (5) is indeed separated from the compatibility conditions Eq. (4) which is one of the key features of the DR scheme. The updated internal forces $K_{ij} \delta_{ij}^{t+\Delta t}$ are obtained from the element stresses at time t which are computed using the element stiffness matrix in local coordinates and then transforming those forces into global (nodal) coordinates (virtual work equivalence in the local and global systems). The resolved components of the residuals are then added as nodal forces. In Section 2.5 the computation of the nodal forces for a 6DOFs beam element is described in more details and references are given for its derivation and extension to other elements.

2.3. Moments equilibrium and rotational motion, a co-rotational approach

Following the co-rotational approach, when rotational degrees of freedom are involved the rigid translational motion of the nodes is de-coupled from the rotational part. As discussed in Section 1.2 each element has its own co-rotated frame and the nodes have their attached frames to compute the rotational part of the deformation with respect to the element's co-rotated frame. In a DR scheme it is only necessary to keep track of the nodes' orientations. In fact, once these are updated it is possible to compute the current co-rotated element orientation from the nodal position (within the small strains assumption).

A parameterization for finite rotations must be chosen to handle the rigid body part of the displacement. An intuitive parameterization is to describe a rotation from axis (unit vector $\boldsymbol{n}$) and angle, often referred as rotational vector. That is the axis of the vector is the axis of the rotation and its magnitude represents the angle (Fig. 2):

$$\boldsymbol{\theta} = \theta \boldsymbol{n}; \qquad \theta = \sqrt{\boldsymbol{\theta}^T \boldsymbol{\theta}}. \tag{6}$$

The relation between the rotational vector and its corresponding rotation matrix are given by the Rodriguez's formula:

$$\boldsymbol{R}(\boldsymbol{\theta}) = I \cos\theta + \sin\theta \, [n]_\times + (1 - \cos\theta) n \otimes n \tag{7}$$

or in matrix form:

$$\boldsymbol{R}(\boldsymbol{\theta}) =$$
$$\begin{pmatrix} \cos\theta + n_x^2(1 - \cos\theta) & n_x n_y(1 - \cos\theta) - n_z \sin\theta & n_x n_z(1 - \cos\theta) + n_y \sin\theta \\ n_x n_y(1 - \cos\theta) + n_z \sin\theta & \cos\theta + n_y^2(1 - \cos\theta) & n_y n_z(1 - \cos\theta) - n_x \sin\theta \\ n_x n_z(1 - \cos\theta) - n_y \sin\theta & n_z n_y(1 - \cos\theta) + n_x \sin\theta & \cos\theta + n_z^2(1 - \cos\theta) \end{pmatrix} \tag{8}$$

where I is the identity matrix, $[n]_\times$ is the skew symmetric matrix from the components of the axis $\boldsymbol{n}$ and $\otimes$ is the tensor product [29]. If the axis and the angle of the rotation are known, it is then possible to establish a relation between the current and time forwarded orientation of the node:

$$\boldsymbol{R}^{t+\Delta t} = \boldsymbol{R}(\boldsymbol{\theta}) \cdot \boldsymbol{R}^t \tag{9a}$$

$$\theta = \sqrt{\left(\Delta t \boldsymbol{\omega}^{t+\Delta t}\right)^T \left(\Delta t \boldsymbol{\omega}^{t+\Delta t}\right)} \tag{9b}$$

where $\boldsymbol{R}$ is used as both operator and to represent the node orientations [19]. As far as the update of the node orientation, it is intuitive to think the angle of rotations is given by Eq. (9b) where $\boldsymbol{\omega}^{t+\Delta t}$ is the angular velocity computed from the updated angular momentum. With the current parameterization for finite rotations it is not possible to handle the change of angle of rotations bigger than 2π. In fact, the relation between the variation of the node orientation and the rotational vector is not a bijection at $\theta = 2k\pi$, $k = 1, 2, \dots$ [30]. However, for small increments of rotational vector, the node orientation update is possible and thus limited only within a time step (the incremental rotational vector) [31,23].

Newton's 2nd law applied to torque and angular acceleration is:

$$\boldsymbol{\tau} = \boldsymbol{I}\dot{\boldsymbol{\omega}}^t + \boldsymbol{\omega}^t \times \boldsymbol{I}\boldsymbol{\omega}^t \tag{10}$$

where $\boldsymbol{\tau}$ is the total torque at node; $\boldsymbol{I}$ the inertia tensor of the element lumped at node; $\boldsymbol{\omega}^t$ and $\dot{\boldsymbol{\omega}}^t$ the angular velocity and acceleration at time t respectively. The second term represents the precession of a rotating body that is the change in the orientation of the rotational axis. The torque $\boldsymbol{\tau}$ can be computed using the local element stiffness matrix and then transforming into global coordinates as already discussed for the out of balance forces in Section 2.2 (virtual work equivalence local–global reference system). In order to derive a suitable stepping procedure to update
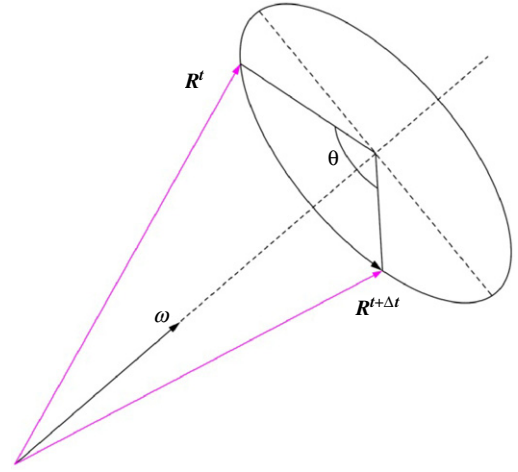

Fig. 2. The rotational vector.

the node orientation, the angular momentum is first computed. The angular momentum and its derivative are defined as:

$$\boldsymbol{L} = \boldsymbol{I}\omega; \qquad \dot{\boldsymbol{L}} = \boldsymbol{I}\dot{\omega} = \boldsymbol{\tau} \tag{11}$$

thus is possible to update the angular momentum and then compute the angular velocity multiplying it by the inverse of the inertia tensor:

$$\boldsymbol{L}^{t+\Delta t} = \boldsymbol{L}^t + \Delta t \, \boldsymbol{\tau}^t \tag{12}$$

$$\boldsymbol{\omega}^{t+\Delta t} = \boldsymbol{I}^{t^{-1}} \boldsymbol{L}^{t+\Delta t}. \tag{13}$$

The inertia tensor is computed in the local element frame but the other quantities in Eq. (13) are instead expressed in world coordinates. An efficient way to get around this problem is to rotate the angular momentum in the element's local frame, apply the inverse of the inertia tensor in local coordinates $\boldsymbol{I}_0^{-1}$ and then rotating back into world coordinates [22]:

$$\boldsymbol{\omega}^{t+\Delta t} = \boldsymbol{I}^{t^{-1}} \boldsymbol{L}^{t+\Delta t} = \boldsymbol{R}^t \boldsymbol{I}_0^{-1} \boldsymbol{R}^{-1^t} \boldsymbol{L}^{t+\Delta t}. \tag{14}$$

The complete sequence to update the node orientation can be summarized as:

$$\begin{cases} \boldsymbol{L}^{t+\Delta t} = \boldsymbol{L}^t + \Delta t \, \boldsymbol{\tau}^t \\ \boldsymbol{I}^{t^{-1}} = \boldsymbol{R}^t \boldsymbol{I}_0^{-1} \boldsymbol{R}^{-1^t} \\ \boldsymbol{\omega}^{t+\Delta t} = \boldsymbol{I}^{t^{-1}} \boldsymbol{L}^{t+\Delta t} \\ \boldsymbol{R}^{t+\Delta t} = \boldsymbol{R}\left(\Delta t \boldsymbol{\omega}^{t+\Delta t}\right) \cdot \boldsymbol{R}^t \end{cases} \tag{15}$$

where $\Delta t \boldsymbol{\omega}^{t+\Delta t}$ represents the incremental rotational vector given by the axis of the angular velocity and whose magnitude represents the rotation of node frame. It is worth noting that the scheme in Eq. (15) is the symplectic Euler as the angular velocity is obtained through a forward step while the update of the node orientation is implicit.

2.4. Convergence and stability

Convergence is usually achieved by damping the nodal movements using kinetic damping or artificial viscous damping. The first can be used when there is no need to track the dynamic behaviour of the system and only stable equilibrium is of interest. The latter applies to both load analysis and form-finding. Kinetic damping involves tracking the kinetic energy of the system and setting the velocities of the nodes to zero whenever a maximum is detected [11]. Kinetic damping is based on the fact that for a system made of masses in harmonic motion, minimum potential energy corresponds to the maximum kinetic energy. The iterative
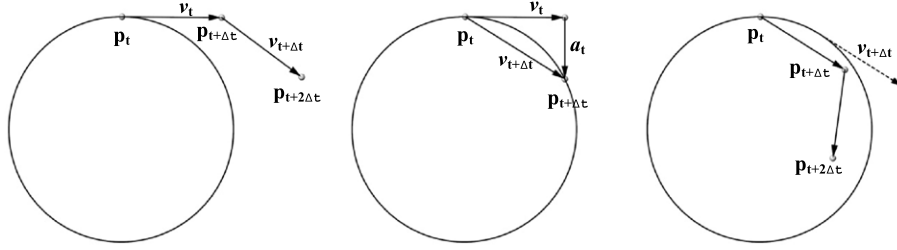
Fig. 3. (a) Forward; (b) symplectic; (c) backward.

application of kinetic damping gradually brings the system to the configuration of stable equilibrium.

Artificial viscous damping can be considered as a way of extracting energy from the system in order to bring it to a rest. The viscous damping factor only affects the way in which convergence is achieved. However, it must be chosen in order to avoid slow convergence due to either over or under damping.

If the damping factor is made proportional to the fundamental modes (critical damping) it is possible to trace the dynamic behaviour of the system. This involves solving the Eigenvalue problem for a system with multiple degrees of freedom. However, due to the fact that DR and co-rotational formulation can be well integrated with standard FEM libraries, the local element matrices are already conveniently implemented. For this reason, it is possible to use a standard assembly routine to pre-compute a global lumped mass and stiffness matrix and solve the eigenvalue problem:

$$\lambda = eig\left(\boldsymbol{M}^{-1}\boldsymbol{K}\right) \tag{16}$$

where λ is the vector of the eigenvalues thus providing the natural frequencies. The value of the damping factor can then be assigned as proportional to any of the natural frequencies or a fraction of the 1st natural frequency if experimental data are available:

$$C = 4\pi m\sqrt{\lambda_i}; \quad i = 1, 2 \ldots \tag{17}$$

where m is the total mass of the system. The computation of the eigenvalues can also be used to set a bound on the integration time step in order to guarantee stability. In fact, it can be proven that a first order approximation bound of Δt for stability is [10]:

$$\Delta t < \frac{2}{\sqrt{\lambda_1}} \tag{18}$$

where λ_1 is the largest eigenvalue. Alternatively another approximation for a bound on the time integration step is given in [32] as:

$$\Delta t < \min_{i<N} \sqrt{\frac{2M_{ij}}{K_{ij}}}; \quad j = x, y, z \tag{19}$$

where, as described previously, the subscripts ij refer for the ith node and the jth direction, N is the total number of nodes. Eq. (19) does not require any pre-computation but only the stiffness at nodes. For the automatic evaluation of DR parameters see also [33].

The integration used for both translational and rotational motion is the semi-implicit Euler (symplectic) for its energy preservation features. As already described the symplectic Euler is a simple modification of the forward Euler which inverts the computation of the velocity with respect to the position. In this way the velocity is computed using the forces at t but the position is updated using the velocity and $t + \Delta t$. Fig. 3 shows the difference between error accumulation between forward, implicit and semi-implicit Euler.

It can be seen that while the forward method (or even higher order method such as the RK4) accumulates the local error spiralling outwardly the ideal orbit whilst an implicit method accumulates error spiralling inwardly. On the contrary, the symplectic method remains stable thus preserving energy. The Stoermer–Verlet (velocity-Verlet) is a symplectic method 2nd order accurate $O(\Delta t^2)$ on position thus, in principle, should be preferable to the symplectic Euler $O(\Delta t)$. However, the former requires two updates of forces (i.e. compute twice forces and moments) each time step. Considering that most of the computational effort is spent computing the forces (especially when the rotational degree of freedom is involved), the gain that would be achieved by a stepping method $O(Dt^2)$ accurate is practically outweighed by the time it takes to perform double computations of forces and moments.

2.5. Beam 6DOfs

In this section, we present a synthetic description of a beam element model with 6 degrees of freedom (three translation and three rotation each node) which is integrated into the physics engine described in previous sections. The beam element can be taken directly from a standard FEM beam element matrix which can be derived either from Euler–Bernoulli or Timoshenko beam theory. The geometric non-linearity (large displacements and large rotations) are handled using the co-rotated element frame and nodes attached frames in order to separate the rigid from the deformational part of the translation and rotational total displacements. The element and the node orientations are updated using the parameterization for finite rotations explained in Section 2.3. Within the assumption of small-strains, the use of linear stress–strain relationships for the formulation of the elements stiffness matrix is allowed. The implementation of the beam model used in the interactive structural models (Section 3) is borrowed from [34] and its derivation can be found in [35].

Fig. 4 shows a deformed beam with co-rotated frame which has the z axis aligned with the vector $\mathbf{p}$ between the two ends' attached frames. The relationship between the frame of the node and beam's end frames is:

$$\boldsymbol{R}_j^t = \boldsymbol{R}_{node}^t \cdot \boldsymbol{R}_j^0; \quad j = 1, 2 \tag{20}$$

where $\boldsymbol{R}_{node}^t$ is the current orientation of the node obtained with Eq. (8) and $\boldsymbol{R}_j^0$ is the initial orientation of the beam end. The element end orientations rotate rigidly with the attached nodes and can be used to define how elements meet at the common node (rest orientation) or as target orientation for optimization tasks. Once the beam end orientations are updated, it is then possible to compute the rotational displacements that they make with the co-rotated element frame (the auxiliary $\mathbf{p}$ vector):

$$\boldsymbol{p} = \boldsymbol{p}_2 - \boldsymbol{p}_1 \tag{21}$$

$$\theta_{x1} = \frac{\boldsymbol{y}_1 \cdot \boldsymbol{p}}{L}; \qquad \theta_{x2} = \frac{\boldsymbol{y}_2 \cdot \boldsymbol{p}}{L} \tag{22}$$

$$\theta_{y1} = \frac{\boldsymbol{x}_1 \cdot \boldsymbol{p}}{L}; \qquad \theta_{x2} = \frac{\boldsymbol{x}_2 \cdot \boldsymbol{p}}{L} \tag{23}$$

Fig. 4. 6DOFs beam element.

$$\varphi = \theta_{z2} - \theta_{z1} = \frac{\boldsymbol{x}_1 \cdot \boldsymbol{y}_2 - \boldsymbol{x}_2 \cdot \boldsymbol{y}_1}{2} \tag{24}$$

where φ is the twist angle and θ_{ji} angles to compute the bending moments and L is the deformed length of the beam. The total elongation e due to the ends relative displacements e_a and the contribution due to bowing e_b can be computed as:

$$e = LArc - L_0 = e_a + e_b \tag{25}$$

$$e_a = \frac{L^2 - L_0^2}{2L_0} \tag{26}$$

$$e_b = \frac{L_0}{60} \begin{bmatrix} \theta_{x1} & \theta_{x2} \end{bmatrix} \begin{bmatrix} 4 & -1 \\ -1 & 4 \end{bmatrix} \begin{bmatrix} \theta_{x1} \\ \theta_{x2} \end{bmatrix} \\ + \frac{L_0}{60} \begin{bmatrix} \theta_{y1} & \theta_{y2} \end{bmatrix} \begin{bmatrix} 4 & -1 \\ -1 & 4 \end{bmatrix} \begin{bmatrix} \theta_{y1} \\ \theta_{y2} \end{bmatrix} \tag{27}$$

where L_0 is the undeformed length and $LArc$ is the arc-length obtained by integrating a Hermite cubic shape function [36]. Using virtual work, the forces and moments with respect to the beam ends' frames can be computed as the derivative of the total strain energy U:

$$T = \frac{\partial U}{\partial e_a} = \frac{EA}{L_0}(e_a + e_b) \tag{28}$$

$$\begin{bmatrix} M_{x1} \\ M_{x2} \end{bmatrix} = \begin{bmatrix} \frac{\partial U}{\partial \theta_{x1}} \\ \frac{\partial U}{\partial \theta_{x2}} \end{bmatrix} = \frac{TL_0}{30} \begin{bmatrix} 4 & -1 \\ -1 & 4 \end{bmatrix} \begin{bmatrix} \theta_{x1} \\ \theta_{x2} \end{bmatrix} \\ + \frac{EI_{xx}}{L_0} \begin{bmatrix} 4 & -1 \\ -1 & 4 \end{bmatrix} \begin{bmatrix} \theta_{x1} \\ \theta_{x2} \end{bmatrix} \tag{29}$$

$$\begin{bmatrix} M_{y1} \\ M_{y2} \end{bmatrix} = \begin{bmatrix} \frac{\partial U}{\partial \theta_{y1}} \\ \frac{\partial U}{\partial \theta_{y2}} \end{bmatrix} = \frac{TL_0}{30} \begin{bmatrix} 4 & -1 \\ -1 & 4 \end{bmatrix} \begin{bmatrix} \theta_{y1} \\ \theta_{y2} \end{bmatrix} \\ + \frac{EI_{yy}}{L_0} \begin{bmatrix} 4 & -1 \\ -1 & 4 \end{bmatrix} \begin{bmatrix} \theta_{y1} \\ \theta_{y2} \end{bmatrix} \tag{30}$$

$$M_\varphi = \frac{\partial U}{\partial \varphi} = \frac{GJ}{L_0} \varphi \tag{31}$$

where E is Young's modulus; G the shear modulus; A the cross-sectional area; I_{xx}, I_{yy} and J the 2nd moment of area and the polar moment. The shear forces do not appear because the moments are computed with respect to the end frame and not to the element frame. In order to transform the forces and moments from local to global coordinates, it is possible to evaluate the infinitesimal increment in the total strain energy (i.e. equivalence of the internal virtual work in both the global and local systems):

$$\delta U = \begin{bmatrix} \delta e_a & \partial \theta_{x1} & \partial \theta_{y1} & \partial \theta_{x2} & \partial \theta_{y2} & \partial \varphi \end{bmatrix} \begin{bmatrix} T \\ M_{x1} \\ M_{y1} \\ M_{x2} \\ M_{y2} \\ M_\varphi \end{bmatrix}. \tag{32}$$

After expressing the derivative terms in Eq. (32) as a function of the beam end frames and the auxiliary vector **p**, thus computing the derivatives of the rotation matrix **R** Eq. (8) [35], the resultant forces and moments in global coordinates can be written as:

$$\boldsymbol{F}_{1i} = \frac{1}{L_0} \left(Tp_i + M_{x1}y_{1i} - M_{y1}x_{1i} + M_{x2}y_{2i} - M_{y2}x_{2i} \right) \tag{33}$$

$$\boldsymbol{F}_{2i} = -\frac{1}{L_0} \left(Tp_i + M_{x1}y_{1i} - M_{y1}x_{1i} + M_{x2}y_{2i} - M_{y2}x_{2i} \right) \tag{34}$$

$$\boldsymbol{\tau}_{1i} = -\varepsilon_{ijk} \left(M_{x1} \frac{p_k y_{1i}}{L_0} - M_{y1} \frac{p_k x_{1i}}{L_0} + M_\varphi \frac{(x_{1j}y_{2k} - y_{1j}x_{2k})}{2} \right) \tag{35}$$

$$\boldsymbol{\tau}_{2i} = -\varepsilon_{ijk} \left(M_{x2} \frac{p_k y_{2j}}{L_0} - M_{y2} \frac{p_k x_{2j}}{L_0} - M_\varphi \frac{(x_{1j}y_{2k} - y_{1j}x_{2k})}{2} \right) \tag{36}$$

$$\varepsilon_{ijk} = \begin{cases} 1 & \text{if } (i, j, k) \text{ is } (1, 2, 3), (2, 1, 3), (3, 1, 2) \\ 0 & \text{if } i = j, j = k, k = i \end{cases}$$

where ε_{ijk} is the Levi-Civita symbol taken only in the anticlockwise direction. Although not explicitly computed in the local frames, the total shear force can be recognized in the last 4 terms of Eq. (33). Forces and moments from each element that meet at a node are added to the residuals to obtain a resultant force and moment for each node. Once the residuals Eq. (5) are computed it is possible to use Eqs. (3) and (4) for the translational displacements and the system of equations in Eq. (15) for the rotational displacements to complete a time-step in Eq. (1).

2.6. Bars and cables

Eqs. (28), (33) and (34) can also be utilized for bar and cable elements. In fact, the terms related to the deformation due to bowing and the moments are zero and the resultant force at node is only given by the axial term.

$$T = \frac{\partial U}{\partial e_a} = \frac{EA}{L_0}(e_a) \tag{37}$$

$$\boldsymbol{F}_{1i} = \frac{1}{L_0} \left(Tp_i \right) \tag{38}$$

$$\boldsymbol{F}_{2i} = -\frac{1}{L_0} \left(Tp_i \right). \tag{39}$$
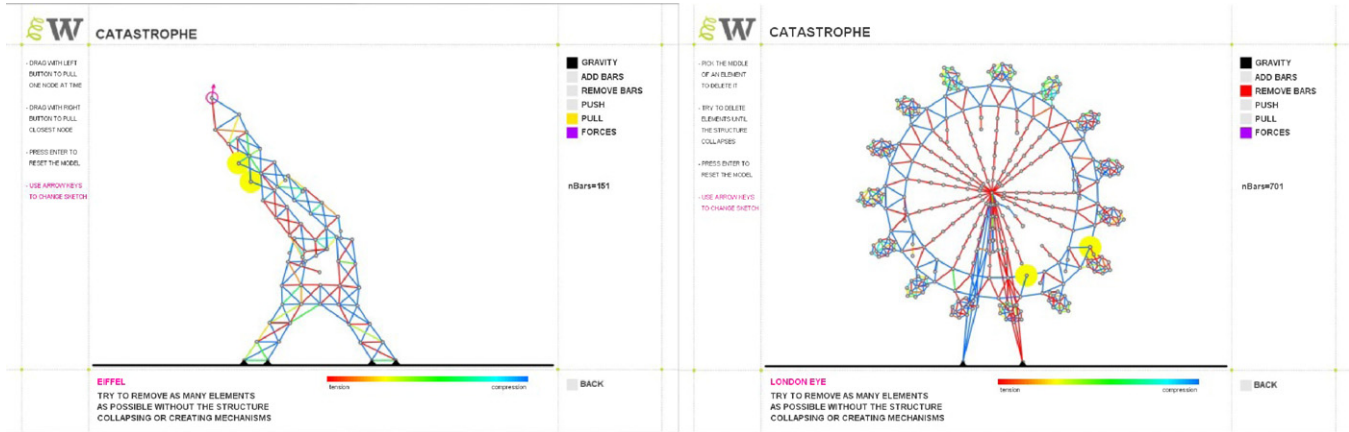
Fig. 5. Catastrophe game.

Fig. 6. (a) Eiffel tower, (b) London's Eye.

For cable elements the force is zeroed in the case of compression. The zero-length cable (usually called spring), only available for optimization tasks, is handled with a case structure in which the force becomes proportional to the length of the element which is also its extension. A recent publication [37] shows how to combine zero-length springs and variable mass proportional to the length of each spring to obtain zero bending moment at nodes for both shells and tension net forms under self-weight.

2.7. Validation

Since the implementation has been targeting primarily to applications where qualitative structural behaviour was of interest, rigorous and exhaustive validation of both stress and displacements obtained with the presented formulation is yet to be done. However, results obtained using configurations such as a cantilever beam subjected to end moment, end force or distributed load compare well against commercial FE software (Oasys GSA). The difference between tip displacements using the built-in non-linear analysis in GSA is less than 2%.

3. Applications

3.1. Qualitative interactive structural analysis

Based on discussions with engineering students and academics we believe that there is a significant opportunity to improve the qualitative understanding of structures through online gaming. In effect this would be a modern analogue for the way that people learnt engineering principles through games like Meccano [38]. Properly done this may also provide a useful way to engage with children at school. Precedents can be found in the work of Martini, Arcade [39].

This section describes the work carried out in developing the software "*PushMePullMe*". PushMePullMe is part of an educational project, the *Expedition Workshed*, which aims at developing a set of tools and digital documents to ease and enrich the process of teaching structural mechanics, engineering and architecture. PushMePullMe offers non-experts the possibility of gaining a qualitative understanding of the behaviour of simple and complex structural topologies, for both static and dynamic systems. The application implements the physics engine described in Section 2 and particularly focuses on the interaction user-model.

The first software development of the physics engine is a game called "*Catastrophe*" implemented as a Java applet.[1] The elements are limited to bars and cables and the displacements in 2 dimensions. (See Fig. 5).

The user is presented with a series of challenges designed around model representations of famous buildings. Initially designed as a student oriented game, Catastrophe challenges users to remove as many elements as possible without the structure collapsing. Through play, users develop an understanding of which elements are critical to system stability (Fig. 6) thanks to the real time feedback. Models can be built and tested under different loading scenarios while receiving immediate feedback on internal forces and constraints reactions. Fracture of highly stressed elements can be enabled (the yellow circle highlight elements fracture), Fig. 7.

PushMePullMe implements a 3D environment with navigation where models can be drawn directly using the built-in CAD tools or alternatively geometry can be imported from other CAD packages. The physics engine uses the parameterization for finite rotations,
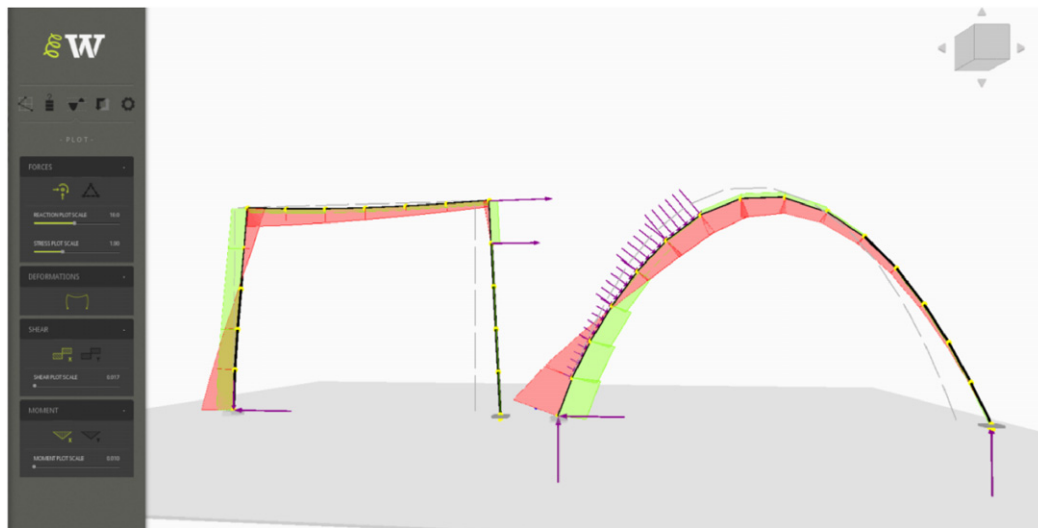

Fig. 7. Interactive loading; plotting shear and bending diagrams.
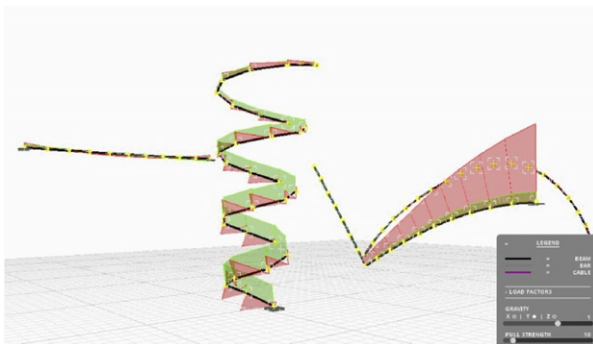

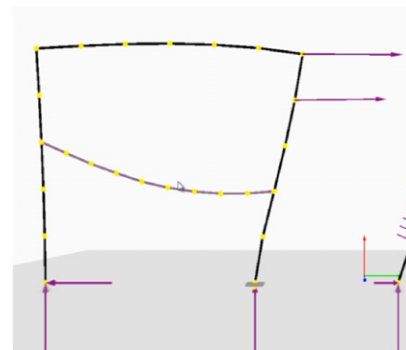Fig. 8. 6Dofs beam; torsion.

Fig. 9. Beams+cables interaction.

the symplectic integration scheme and the 6DOFs beam element described in Section 2.

The software, implemented as Java applet,[2] allows the creation and interactive manipulation of the geometry of structural models. Interaction is enhanced by the usage of interactive frames [40] which are associated to each node and element of the structures. The interactive frames are objects that can be directly picked in 3D (Figs. 7–9). Loading can be done by directly pulling or by applying point loads on the nodes and distributed loads on the elements. Pulling is implemented using zero rest-length cable-elements (springs) which are instantiated on the fly as soon as the mouse is within a certain distance from a node or an element. Point and distributed loads are instead vector fields.

To further enhance the interaction, an icon-based interface with sliders is implemented. The icons can be dragged and dropped from the interface tab onto the 3D canvas to apply/remove loads/constraints, create/change type of elements, release internal element degrees of freedom, plot internal force and moments and stress diagrams. The sliders can be used to modify section/material properties E, G, A, Ixx, Iyy, J (see Section 2.5), plot scales and other parameters for the integration such as the viscous damping factor C and the time step Δt.

Pre-stress is modelled by setting the rest-length of elements shorter than their initial length. This makes it possible to

interrogate structures stable only in states of self-stress. Tension structures can be modelled by simply reducing the rest-length of the cable elements that can be changed interactively (Fig. 10).

Using the same logic, the rest-length can be set to simulate element expansion thus exerting both compression and tension forces. This can be used to simulate both determinate and inde-terminate truss structures with embedded actuators which can re-spond actively to loads by redirecting internal forces [41] or morph into different shapes (Fig. 11).

A basic optimization routine is built-in to allow minimization of bending stresses while at the same time equalizing the length of the elements (Fig. 12). This is simply achieved by releasing the in-ternal degrees of freedom of the beams and setting each element's rest length the average length of the neighbouring elements it-eratively. It is worth noting that due to the explicit nature of the physics engine, optimization tasks of this kind are implemented as straightforward extension of the main routine.

PushMePullMe is currently used in several universities in UK amongst which the University College London, Brunel, Strathclyde and East Anglia. Recent studies, which were carried out at the de-partment of Architecture and Civil engineering at Bath Univer-sity, provide some qualitative evidence of the benefit achieved after the introduction of PushMePullMe as a learning experience [42,43]. According to these studies PushMePullMe is perceived by the students as the most useful resource for learning and under-standing the structural behaviour compared to other software and basic physical testing. However, quantitative data (i.e. increase in the pass rate on the exam) were difficult to attribute solely to the use of the software.

___

[2] The application can be accessed at http://expeditionworkshed.org/workshed/pmpm-3d/; demo movies can be viewed at https://vimeo.com/groups/pushmepullme.
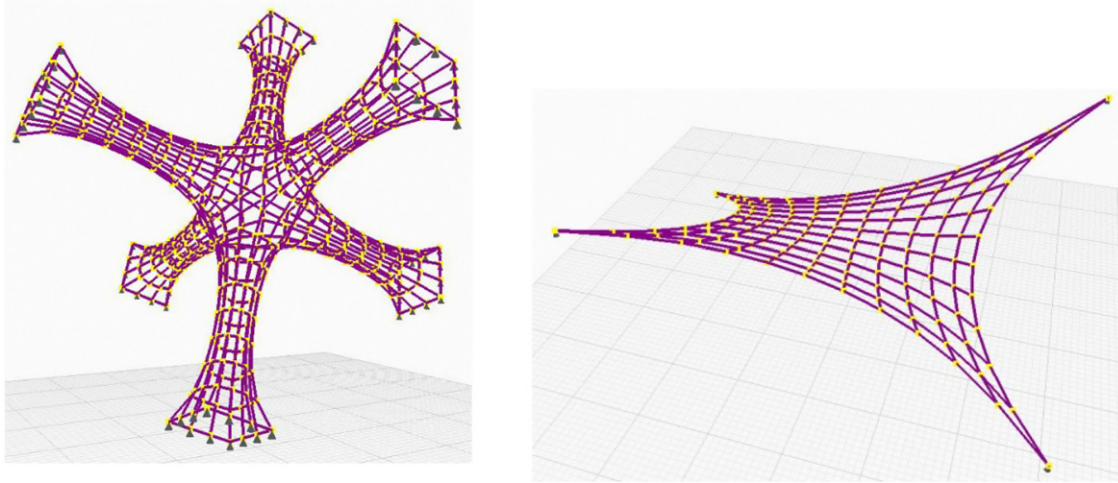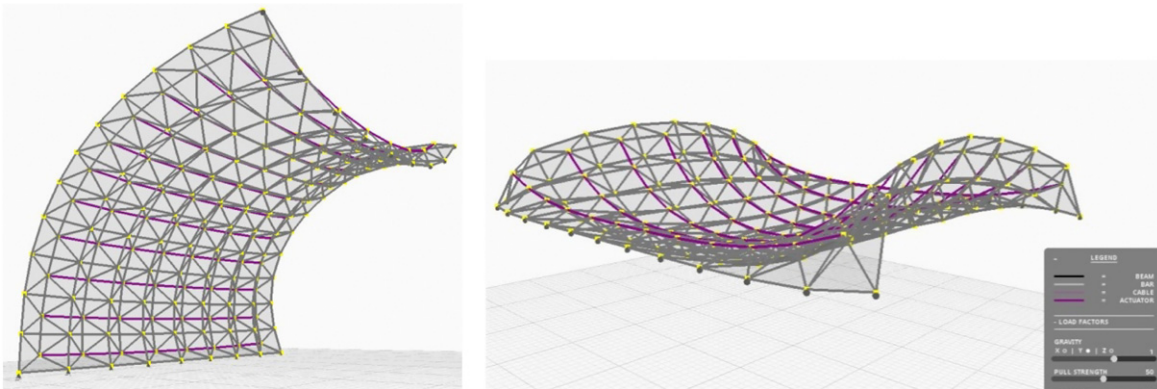
Fig. 10. Cable nets form-finding.

Fig. 11. Truss structures with embedded actuators (magenta lines). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
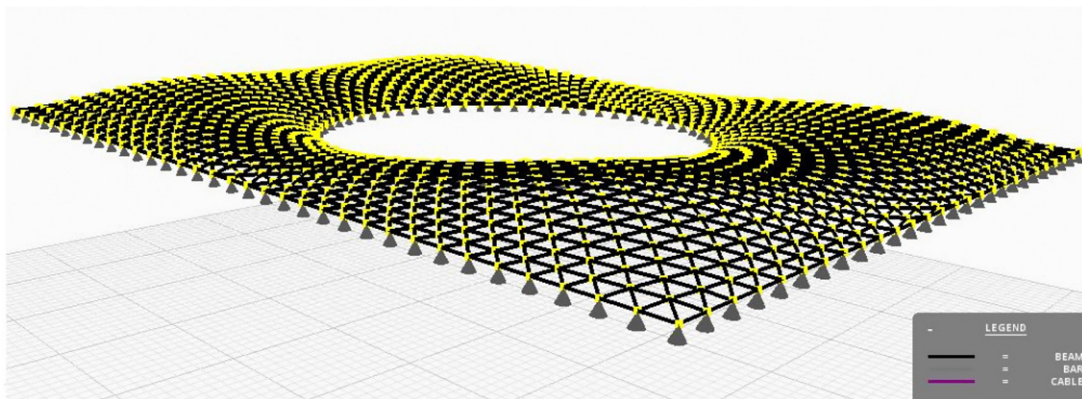

Fig. 12. British Museum Great Court Roof case study.

4. Conclusions

Conventional engineering education draws a neat separation between linear and non-linear methods for structural analysis. In addition, design through optimization whether for size, shape or geometry is regarded as an advanced topic and most often is not part of the students' curriculum. Combining computational techniques developed in computer graphics and established numerical methods for structural analysis creates a platform that can be used by students and designers to gain a qualitative and quantitative understanding of the structural behaviour of simple and complex systems. Real-time feedback and the possibility to interact with the model offer tremendous potential in teaching.

Next steps will focus on expanding the range of elements available to plates/shells and 3d elements such as tetrahedrons and 8-node bricks. Further investigation to understand the benefits that such a platform can bring in teaching and design by means of analysis of performances in terms of structural efficiency, fabrication constraints and aesthetics will continue in order to gather stronger quantitative evidence.

Acknowledgements

References

[1] Day A. An introduction to dynamic relaxation. Engineer 1965;220–1.
[2] Crisfield M. A consistent co-rotational formulation for non-linear, three-dimensional, beam-elements. Comput Methods Appl Mech Engrg 1990;81(2): 131–50.
[3] Williams C. British Museum Great Court Roof, 2000. [Online] Available at: http://people.bath.ac.uk/abscjkw/BritishMuseum/ [accessed 14.04.13].
[4] Li J, Knippers J. Rotation formulations for dynamic relaxation—with application in 3D framed structures with large displacements and rotations. Seoul, s.n. 2012.
[5] Nealen A, et al. Physically based deformable models in computer graphics. Dublin: The Eurographics Association; 2005.
[6] Witkin A. Physically based modeling: principles and practice, 1997. [Online] Available at: http://www.cs.cmu.edu/~baraff/sigcourse/ [accessed 08.04.11].
[7] Verlet L. "Computer experiments" on classical fluids, I. Thermodynamical properties of Lennard-Jones molecules. Phys Rev 1967;159(98).
[8] Newmark N. A method of computation for structural dynamics. ASCE J Engrg Mech 1959;83:67–94.
[9] Otter J. Computations for prestressed concrete reactor pressure vessels using dynamic relaxation. Nucl Struct Eng 1965.
[10] Brew J, Brotton D. Non-linear structural analysis by dynamic relaxation. Internat J Numer Methods Engrg 1971;3(4):463–83.
[11] Barnes M. Finding and analysis of tension space structures by dynamic relaxation. London: City University; 1977.
[12] Wakefield D. Dynamic relaxation analysis of pre-tensioned networks supported by compression arches. London: City University; 1980.
[13] Ramesh G, Krishnamoorthy C. Post-buckling analysis of structures by dynamic relaxation. Internat J Numer Methods Engrg 1993;36:1339–64.
[14] Sauve R. Advances in dynamic relaxation techniques for nonlinear finite element. J Press Vessel Technol 1995;117:170–6.
[15] Lewis W. Tension structures. London: Thomas Teldford Publishing; 2003.
[16] Kilian A, Ochsendorf J. Particle-spring systems for structural form finding. J Int Assoc Shell Spat Struct 2005.
[17] Block P. Thrust network analysis. Cambridge (MA, USA): Massachusetts Institute of Technology; 2009.
[18] Piker D. Kangaroo: form finding with computational physics. Archit Des 2013; 83(2):136–7.
[19] Crisfield N. Non-linear finite element analysis of solids and structures: advanced topics. 1st ed. New York: John Wiley and Sons; 1997.
[20] Belytschko T, Hsieh B. Application of higher order corotational stretch theories to nonlinear finite element analysis. Comput Struct 1979;11:175–82.
[21] Felippa C, Haugen B. Unified formulation of small-strain corotational finite elements. Boulder (Colorado): University of Colorado; 2005.
[22] Van Den Bergen G, Gregorius D. Game physics pearls. Natick: A K Peters; 2010.
[23] Battini J. Corotational beam elements in instability problems. Stockholm: Royal Institute of Technology; 2002.
[24] Nam Le T. Corotational formulation for nonlinear dynamic analysis of flexible beam structures. Stockholm: KTH School of Architecture and the Built Environment; 2012.
[25] Argyris J. An excursion into large rotations. Comput Methods Appl Mech Engrg 1982;32:85–155.
[26] Barnes M. Form finding and analysis of tension structures by dynamic relaxation. Int J Space Struct 1999;14:89–104.
[27] Ambrosio J, Neto M. Dynamics of structural components subjected to large rotations using a flexible multibody approach (Invited Paper). CapeTown: Taylor and Francis; 2013.
[28] Tong P, Pian THH, Bucciarelli LL. Mode shapes and frequencies by finite element method using consistent and lumped masses. Comput Struct 1971; 1:623–38.
[29] Taylor CJ, Kriegman J. Minimization on the lie group SO(3) and related manifolds. s.l.: Yale University; 1994.
[30] Ibrahimbegovic A. On the choice of finite rotation parameters. Comput Methods Appl Mech Engrg 1997;149:49–71.
[31] Cardona A, Geradin M. A beam finite element non-linear theory with finite rotations. Internat J Numer Methods Engrg 1988;26:2403–38.
[32] Barnes M. Dynamic relaxation analysis of tenion networks. Struct Eng Rev 1994;6:175–202.
[33] Papadrakakis M. A method for the automatic evaluation of the dynamic relaxation parameters. Comput Methods Appl Mech Engrg 1981;25: 35–48.
[34] Williams C. Private communications. Bath, s.n. 2011.
[35] Adriaenssens SML. Stressed spline structures, Ph.D. Thesis, Bath: Bath University, 2000.
[36] Kythe K, Wei D. One-dimensional shape functions. In: An introduction to linear and nonlinear finite element analysis. Boston: Birkhäuser Boston; 2004. p. 25–36.
[37] Harding J, Shepherd P. Structural form finding using zero-length springs with dynamic mass. In: IABSE-IASS Symposium London 2011. 2011.
[38] McCann E, Senatore G, Crowther A. ExpeditionWorkshed: engineering education for the 21st century. London, s.n. 2012.
[39] Martini K. Non-linear structural analysis as real-time animation. In: Computer aided architectural design futures. s.l.: Springer Netherlands; 2001. p. 643–56.
[40] Charalambos J. Proscene, 2013. [Online] Available at: https://code.google.com/p/proscene/ [accessed 15.05.13].
[41] Senatore G, et al. Large scale adaptive structures for whole life energy savings. Int Assoc Shell Spat Struct (IASS) 2011;52(4): December n. 170.
[42] Palmer J. Structural engineering education: what are the gaps between industry and universities? Bath: Department of Architecture and Civil Engineering The University of Bath; 2011.
[43] MacDonald M. The benefits of online educational resources toward understanding structural analysis. Bath: Department of Architecture and Civil Engineering The University of Bath; 2012.