COMPUTATIONAL ISSUES REGARDING LATTICE MODELS FOR WOOD

Thomas Reichert[1]**, Daniel Ridley-Ellis**[2]

ABSTRACT: This paper describes means to overcome some of the computational issues related to lattice models. The first is the use of a solution technique, different from a Newton-Raphson approach, called the Step-Size-Control (SSC) algorithm to handle strain-softening behaviour of individual elements and to account for the possible snap-back of the load-displacement path. The second is a method to circumvent the need to recalculate the global stiffness matrix in each load step, called the Method of Inelastic Forces (MIF). Thirdly, a significant reduction in the model's degrees of freedom is achieved by using a hybrid system of lattice and solid elements, for which the lattice is only used in areas of high stress gradients. Details of these optimisations are given and their implementation is shown for a 3D example model. While the hybrid model and the MIF can be generally used and significantly reduce computational costs, the SSC routine is better suited for lattice models in which only a small number of links change to a plastic or strain-softening state.

KEYWORDS: Lattice model, 3D FEM, Fracture

1 INTRODUCTION

A relatively new approach to model timber's material behaviour is the use of lattice models to predict nonlinear material behaviour. Recent publications in that field demonstrate the possibilities of this method [*e.g.*1-4]. Elastic-plastic bar elements arranged in a 3D lattice that are able to provide softening are commonly used.

One problem with these models is the large amount of degrees of freedom (DoF) due to the sheer size of a lattice structure; the density of the lattice depends on the incorporated material heterogeneity. The lattice model takes into account material variability of wood by adjusting properties such as the elastic modulus and yield and tensile strengths of the individual lattice elements at the scale of growth rings. Therefore, a large number of such elements and thus DoF are required. This paper presents several approaches that, combined, try to overcome the computational problems that arise by modelling large-scale timber specimens in three dimensions.

2 LATTICE MODEL STRUCTURE

The basic structural element in a lattice that is used in these models, is a 3D bar element.

In order to create a lattice with as few nodes as possible, the unit cells of a lattice have been arranged in a diagonal checked pattern. Thus, instead of constructing nodes at each potential junction of links only every second node is used (Figure 1). With this diagonal arrangement (in contrast to a rectangular one) it is possible to therefore reduce the degrees of freedom of the structure while still retaining the detailed variation in strength and stiffness parameters of individual links.

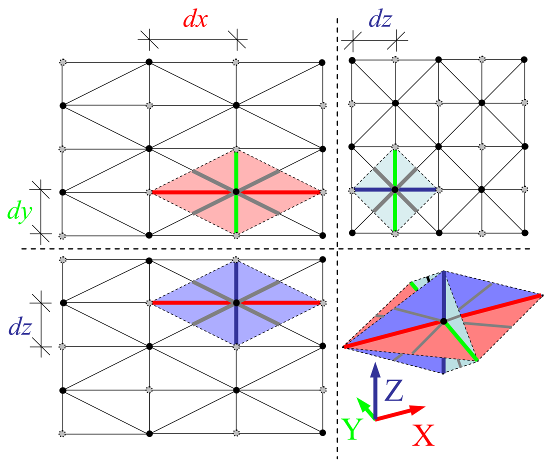

***Figure 1:** Lattice cell structure, one unit cell consists of 'half'-links extending from the node towards all 9 possible directions*

Each bar element follows a bi-linear load-strain relation, as can be seen in Figure 2. For the case that plastic compressive behaviour is accounted for (which, for reasons of clarity, shall not be included in this paper) this

[1] Thomas Reichert, School of Engineering & the Built Environment, Centre for Timber Engineering, Edinburgh Napier University, Edinburgh, United Kingdom. Email: thomas.reichert@gmx.net
[2] Daniel Ridley-Ellis, Centre for Timber Engineering, Edinburgh Napier University, Edinburgh, United Kingdom. Email: d.ridleyellis@napier.ac.uk

would be extended to a tri-linear curve. Parameters that describe the material behaviour of one link l are the elastic stiffness K_l, tensile strength $S_{T,l}$ and post yield softening parameter $\gamma_{T,j}$, from which the remaining parameters can be calculated (strain at maximum load $\varepsilon_{T,p,l}$ and strain at failure $\varepsilon_{T,f,l}$).
During the nonlinear solution algorithm each link can take on one of 4 different states:

① initial state, stiffness K_l
② softening, stiffness $K_{S,l}$
③ reloading, stiffness $K_l^{③}$
④ broken, stiffness $K = 0$
⑤ temporary state after reloading

This is depicted graphically in Figure 2. The solution algorithm is explained further in Section 3.1.

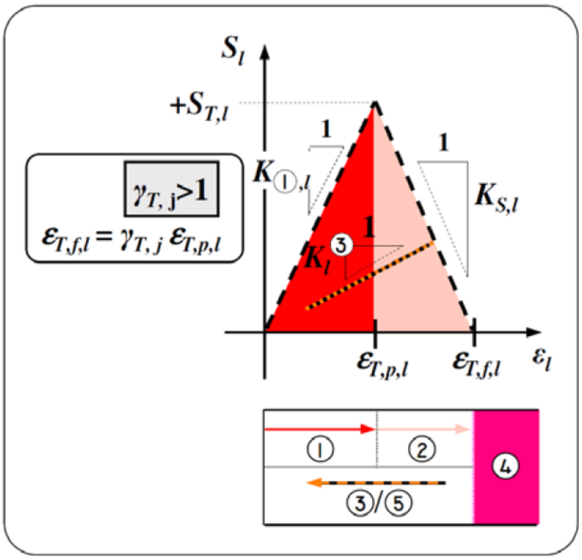

Figure 2: Load-strain curve and parameters of single lattice element l

After a link has reached its maximum tensile strain ε_f (status ④), the element is removed from the global stiffness matrix.
The bar elements in one unit cell are arranged in a 3D lattice as shown in Figure 1. Each unit cell consists of 6 different types of elements that represent material behaviour in the longitudinal, lateral and diagonal (shear) direction.
For the purpose of the study the anisotropic character of wood is accounted for by adjusting strength and stiffness parameters of individual link elements. In addition to generally applied random values with a defined mean and standard deviation as parameters, these are further adjusted according to the link's position in a randomly generated growth ring structure with a density profile in between the rings. This virtual growth ring structure with its density profile is based on several input parameters which were measured on the timber samples used. The density is then translated into stiffness and strength variation of individual link elements. A more detailed explanation on how the heterogeneous character of timber is mapped onto the lattice structure can be found in [5].

3 COMPUTATIONAL OPTIMISATION

Several approaches can be adopted to minimise the computational effort which arises from solving lattice models. The two first presented here are based on a paper by Jirasek and Bazant [6].

3.1 STEP SIZE CONTROL ALGORITHM (SSC)

Since traditional solution algorithms, such as the Newton-Raphson method, cannot handle the jagged character of the load displacement curve (Figure 3) very well, a new approach was chosen in this study.

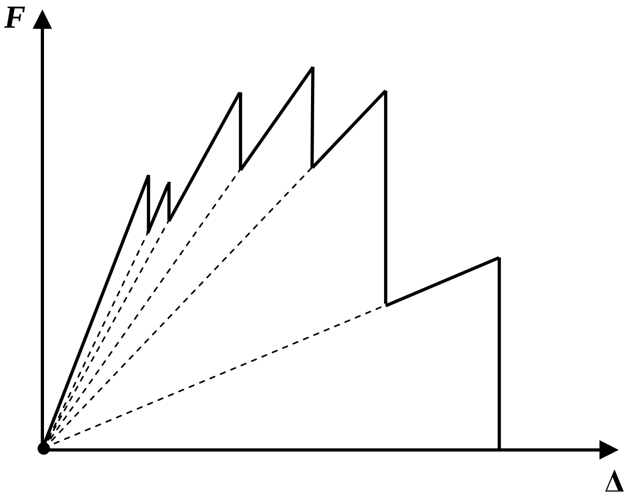

Figure 3: Load-displacement plot of a small perfectly-brittle lattice model (i.e. no softening behaviour of individual links is incorporated)

Rather than iterating towards an admissible solution, the Step-Size-Control (SSC) algorithm treats the overall load-displacement curve as the sum of single linear steps. In each of these steps only one link changes its stiffness. Thus, it is possible to overcome the problems that result from the 'snap-back' phenomena where the load suddenly drops and the next admissible solution has to be found on the next lower stiffness gradient (dashed line in Figure 3).
A drawback of this method is that, for large lattice systems, the number of load steps amounts to at least the number of broken/changed links. However, no additional iterations are needed. In contrast, iterative methods such as Newton-Raphson can require an indefinite number of iterations and might not, in certain circumstances, come to an admissible solution at all.
The method shall be derived here in brief. In a general incremental solution method for a structural system with global stiffness matrix $[K_i]$ and a vector of applied forces $\lambda_i \cdot \{F_{ref}\}$ at load step i, the unknown global displacements $\{\Delta_i\}$ can be calculated as follows (Equation 1).

$$[\boldsymbol{K}_i]\{\Delta_i\} = \lambda_i \{\boldsymbol{F}_{ref}\}. \tag{1}$$

For the SSC algorithm, the necessary λ_i has to be chosen as the smallest value for which one link changes its status. That is from the initial stiffness ① to softening ② and from softening to broken ④. Thus a list has to be created with all load factors $\lambda_{i,l}$ for the respective link l to bring the link into the next state. These factors can be calculated by first solving the above equation with a temporary load factor $\lambda_i = 1$ and thus determining a reference displacement vector $\Delta_{i,ref}$. With this the strain increment $d\varepsilon_{i,l}$ for each link l can be extracted. The individual load factor $\lambda_{i,l}$ can then be calculated from the previous strain and the critical strain for which a link would change its state (Equation 2).

$$\lambda_{i,l} = \frac{\varepsilon_{cr,l} - \varepsilon_{i-1,l}}{d\varepsilon_{i,l}}. \quad (2)$$

For the simplest case when only positive $\lambda_{i,l}$ are present the lowest load factor of all links can be chosen.
However, since the case also arises where links that are in a softening state ② can start to reload when a further positive load F_{ref} is applied (*i.e.* have a negative strain increment and thus also a negative load factor is calculated), these links must have the ability to experience a negative strain increment and change to status ③.
Still, to which link this applies is not known in advance since in that case a new stiffness $K_l^{③}$ needs to be calculated and Equation (1) be solved again. Thus, an 'iteration of status' has to be performed where links that have a negative strain increment temporarily obtain state ⑤ with the newly calculated stiffness. In this iteration the links could change their strain increment again and thus their state needs to be changed. It would be necessary to try every possible combination of softening ② and reloading ⑤ links. This would clearly be not practical, therefore, the number of links (in states ② and ⑤) for which their status is consistent with their strain increment are compared to those which are inconsistent. If there are more inconsistent ones the smallest negative load step is chosen and the status of links that are consistent is swapped (⑤ to ② and ② to ⑤). If there are more consistent ones, a positive load step is chosen and the status of the inconsistent ones is swapped.
After that, all links with status ⑤ are transferred to ③, the link that corresponds to the smallest positive or respectively negative $\lambda_{i,l}$ changes its state and the new displacement Δ_{i+1} and load F_{i+1} is calculated with the chosen λ_i. This procedure is repeated until the applied load F reaches zero.
The computation time is not really reduced with this procedure if, for each load step, the entire global stiffness matrix needs to be decomposed again. The following method uses a trick to get around this problem.

3.2 METHOD OF INELASTIC FORCES (MIF)

Instead of assembling and decomposing the entire global stiffness matrix for each load step i, the Method of Inelastic Forces can be used.
In each step of the SSC algorithm only one link will change its stiffness. The idea is to calculate the force that is required to be applied at the link's nodes in order to deform the link in the same way as it would deform when the stiffness of this link changes. Thus, only these forces need to be applied instead of an actual change in the stiffness matrix.
The initial global stiffness matrix with geometric linear link elements can be expressed as:

$$[\boldsymbol{K}_1] = [\boldsymbol{B}]^T [\boldsymbol{D}_1][\boldsymbol{B}] \quad (3)$$

where $[B]$ is the geometric matrix with each link element in one row and the global DoF in each column. This geometric vector for one link element can be derived from $\{e\}=[B]\cdot\{\Delta\}$, where $\{e\}$ is a vector with all the extensions of the respective links and $\{\Delta\}$ the global displacement vector. The square matrix $[D_1]$ consists of the initial stiffness values of the respective links on the diagonal. Similarly for the current step i it can be written that:

$$[\boldsymbol{K}_i] = [\boldsymbol{B}]^T [\boldsymbol{D}_i][\boldsymbol{B}]. \quad (4)$$

Note that since only geometric linear elements are used only the [D] matrix will change while [B] remains the same throughout the solution procedure.
The general aim of the method is to solve a system of equations as stated in Equation (1) without assembling and decomposing the global tangent stiffness matrix $[K_i]$. Instead, the once-only decomposed initial stiffness matrix $[K_1]$ shall be used. The derivation of this method is be presented below.
By adding and subtracting the initial stiffness matrix to Equation (1) it can be rewritten as:

$$([\boldsymbol{K}_1] + [\boldsymbol{K}_i] - [\boldsymbol{K}_1])\{\Delta_i\} = \lambda_i\{\boldsymbol{F}_{ref}\} \quad (5)$$

which rearranged gives

$$[\boldsymbol{K}_1]\{\Delta_i\} = \lambda_i\{\boldsymbol{F}_{ref}\} - ([\boldsymbol{K}_i] - [\boldsymbol{K}_1])\{\Delta_i\}. \quad (6)$$

Substituting $[K_i]$ and $[K_1]$ on the right hand side with the definition in Equation (4) and Equation (3) the former Equation (6) can be further expressed as:

$$[\boldsymbol{K}_1]\{\Delta_i\} = \lambda_i\{\boldsymbol{F}_{ref}\} - [\boldsymbol{B}]^T([\boldsymbol{D}_i] - [\boldsymbol{D}_1])[\boldsymbol{B}]\{\Delta_i\}. \quad (7)$$

It is important to note that $([D_i]-[D_1])$ is a diagonal matrix with non-zeros (on the diagonal) only for links where stiffness changes occur. Thus, it is possible to write much smaller matrices $[\hat{\boldsymbol{B}}]$ und $[\hat{\boldsymbol{D}}]$ by leaving out all the rows in $[B]$ that correspond to zero columns in $([D_i]-[D_1])$, thus:

$$[\boldsymbol{B}]^T[\boldsymbol{D}_i] - [\boldsymbol{D}_1][\boldsymbol{B}] = [\hat{\boldsymbol{B}}]^T[\hat{\boldsymbol{D}}][\hat{\boldsymbol{B}}]. \quad (8)$$

Equation (7) can be rewritten as:

$$[\boldsymbol{K}_1]\{\Delta_i\} = \lambda_i\{\boldsymbol{F}_{ref}\} - ([\hat{\boldsymbol{B}}]^T[\hat{\boldsymbol{D}}][\hat{\boldsymbol{B}}])\{\Delta_i\}. \quad (9)$$

Since $\{e\}=[B]\{\Delta_i\}$ is the vector of all axial link extensions in step i, $\{\hat{e}\}=\{\hat{\boldsymbol{B}}\}\{\Delta_i\}$ is the axial extension of only the changed links. Thus, multiplied with the stiffness $[\hat{\boldsymbol{D}}]$, the term:

$$\{\hat{s}\} = [\hat{D}][\hat{B}]\{\Delta_i\} \tag{10}$$

can be interpreted as the actual *inelastic* force vector of the changed links.
With Equation (10), the previous Equation (9) can be rewritten as:

$$[K_1]\{\Delta_i\} = \lambda_i\{F_{ref}\} - [\hat{B}]^T\{\hat{s}\}. \tag{11}$$

Since the reference displacement $\{\Delta_{1,ref}\}$(for the initial load step $i = 1$ and $\lambda_i = 1$) is calculated by:

$$[K_i]\{\Delta_{1,ref}\} = \{F_{ref}\} \tag{12}$$

and it can be stated that:

$$[K_1][\hat{R}] = -[\hat{B}]^T. \tag{13}$$

It follows by inserting Equation (12) and Equation (13) into Equation (11) that:

$$[K_1]\{\Delta_i\} = [K_1]\lambda_i\{\Delta_{i,ref}\} - [K_1][\hat{R}]\{\hat{s}\}. \tag{14}$$

Dividing by $[K_1]$ leads to:

$$\{\Delta_i\} = \lambda_i\{\Delta_{i,ref}\} - [K_1][\hat{R}]\{\hat{s}\}. \tag{15}$$

Now, the fundamental equation of this method can be obtained by substituting Equation (15) into Equation (10) and after rearranging this becomes:

$$([I] - [\hat{D}][\hat{B}][\hat{R}])\{\hat{s}\} = [\hat{D}][\hat{B}]\{\Delta_{i,ref}\}\lambda_i. \tag{16}$$

With this Equation (16) the inelastic forces $\{\hat{s}\}$ can be solved for. By substitution into Equation (15) the displacements $\{\Delta_i\}$ are obtained. Since for this step the actual load factor λ_i is not known in advance it is preliminarily set to 1.
The factor will be determined only after the SSC algorithm delivers the required factor to change a link from one status to another.

Thus, not the actual displacement vector is calculated, but the reference displacement $\Delta_{i,ref} = \Delta_i$ with $\lambda_i = 1$ for load step *i*.
To summarise the MIF method: a link that changes its stiffness in one load step is listed in the rows of the $[\hat{B}]$ matrix. Furthermore a new column calculated with Equation (13) is added to matrix $[\hat{R}]$ which has as many rows as there are DoF. Then Equation (16) is solved for inelastic forces $\{\hat{s}\}$ and the global reference displacement vector can be obtained. Thus, in each step only a brief forward calculation in Equation (13) needs to be performed for a single column. If the initial global stiffness matrix $[K_1]$ is already transformed with a Cholesky decomposition this reduces to a simple forward and backward substitution. After that, Equation (16) requires little computation time.
The MIF method is generally applicable to all nonlinear solution algorithms where nonlinearity arises from change of stiffness values. For structural systems with a large number of DoF this process is less computational expensive than decomposing the entire current stiffness matrix. In the case that the computation of the inelastic forces due to a large number of changed link elements becomes more time consuming than decomposing the entire global stiffness matrix, a new tangent stiffness matrix based on the current changed links can be calculated and replaces the old one in the MIF algorithm. Thus, this routine will always be computational more effective than the decomposition of the stiffness matrix in every single load step.

3.3 HYBRID MODEL OF LATTICE AND SOLID ELEMENTS

A further significant saving of computation time can be accomplished by using the denser lattice structure only in areas of the model where large strains are expected, and using ordinary linear solid elements elsewhere. This requires some knowledge of the fracture path before setting up the arrangement of elements in the model.

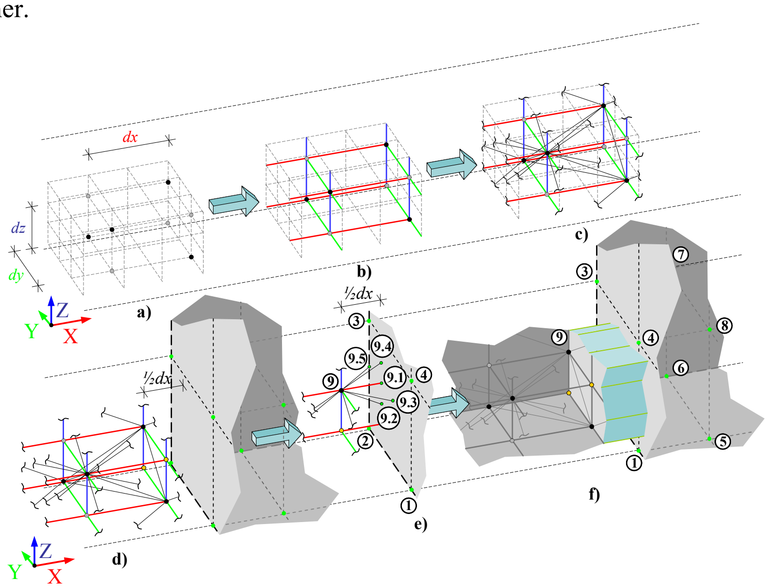

Figure 4: *Hybrid model of lattice and solid elements. Connection of node 9 to the solid elements requires virtual slave nodes to be created (9.1 to 9.5) that are linked to the master nodes (1 to 4).*

The basic principle to connect the two types of elements is a master-slave relation of the participating nodes. Note that the slave nodes on the surface of the lattice structure represent a denser mesh than the connecting master surface.
Since one node with its connecting links represents the material surrounding it in a volume with the dimensions of $2dx \times 2dy \times 2dz$ the whole lattice has to be shifted half of these lengths away from the connecting surface.
The nodes at the connecting side of the lattice structure are linked to the nodes of the solid elements with the penalty element method. A constraint equation can be formulated that connects these nodes. As an intermediate step virtual slave nodes are created which lie on the surface of the solid element (nodes 9.1 to 9.5 in Figure 4).
After formulating the constraint equations between nodes 9.1 to 9.5 and 1 to 4, these can be described as well directly connecting slave node 9 with master nodes 1 to 4 including the required stiffness of a half link in between the two connecting blocks (green area). This spares the effort to create additional nodes on the master surface (9.1 to 9.5).

3.4 EXAMPLE MODEL

A cleavage specimen shall be used here to demonstrate the different elements of a lattice model.
These fracture tests were conducted with small clear Sitka spruce specimens that contained a notch with a 4 mm hole drilled at the root of the notch. The test set-up can be seen in Figure 5. Two brackets fix the specimen in place. The load (under displacement control) is applied to the upper bracket while the lower one is kept fixed to the testing machine. One displacement transducer, which is mounted on the notch side, measures the crack mouth opening displacement.

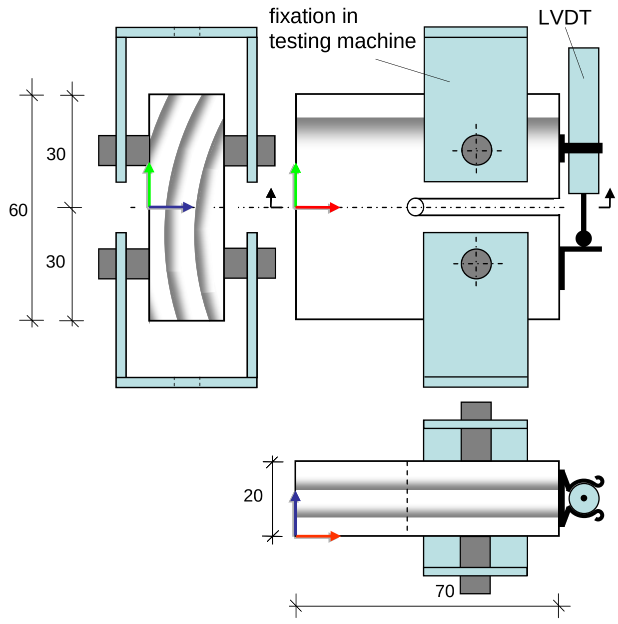

***Figure 5:** Test setup for cleavage test*

Figure 6 shows the arrangement of areas with lattice and solid elements. The red arrows represent the applied load, while green triangles depict the boundary conditions.
As an input to the model, the elastic parameters for the link elements where calibrated to values from literature. For this, stiffness parameters K_l were directly inferred from the elastic moduli and Poisson ratios.

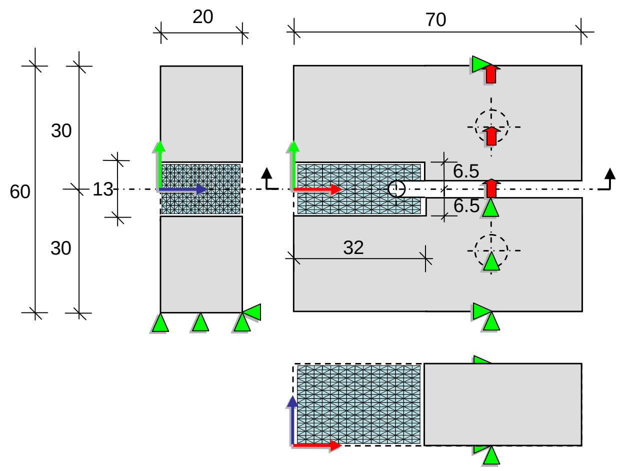

***Figure 6:** Schematic of the cleavage model*

Strength S and post yield parameters γ were determined by an iterative calibration routine. Load displacement curves of tested timber specimens under different loading conditions (parallel and perpendicular to the grain compression and tension and parallel to the grain shear) were compared to FE model results. The parameters were then iteratively adjusted by several FE model runs. For a more detailed explanation of the calibration routine the reader is referred to [5, 7]

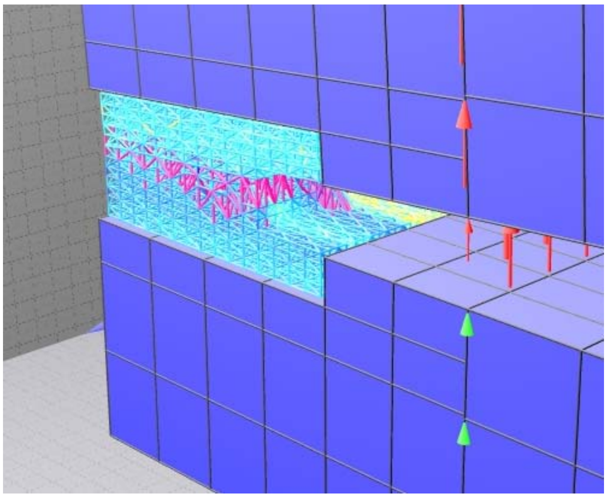

***Figure 7:** Rendered output of the cleavage model, broken links and link's strength ratio is depicted as purple and light-to-dark blue coloured lines respectively*

Figure 7 shows the model output with broken links as purple lines. The fracture path can be seen running from the root of the notch towards the other side. It can be seen that material heterogeneity influences the fracture path. This is depicted in Figure 8, where the broken lattice elements (purple) follow the growth ring structure along the black dotted line.

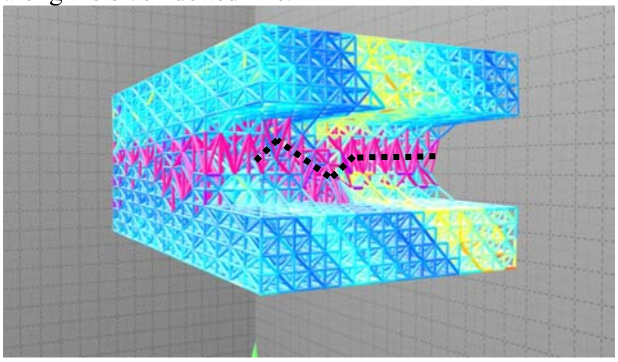

Figure 8: *Detail of fractured lattice area*

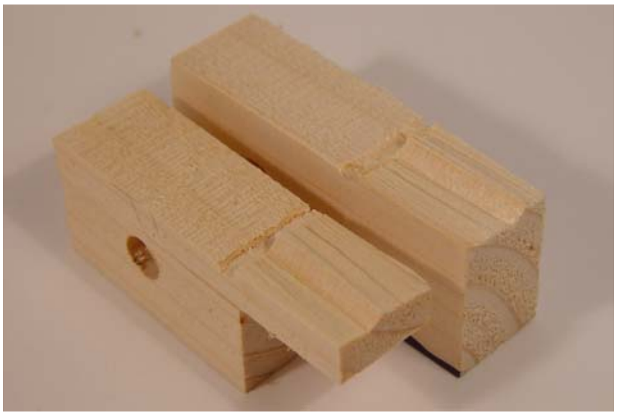

Figure 9: *Tested cleavage specimen*

A broken specimen is pictured in Figure 9. The influence of the growth ring structure on the fracture path can be seen.

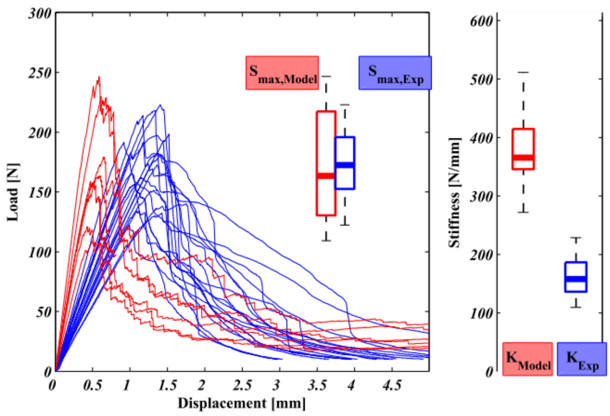

Figure 10: *Comparison of load-displacement plots between tested fracture specimen and FE model*

The load-displacement plots from the tested specimens can be compared with the plots from several FE model runs. The respective median and standard deviations of a) maximum load S_{max} and b) specimen stiffness K are plotted on the right hand side of Figure 10. While good agreement can be found for the comparison of maximum loads, this is less the case for the specimen's stiffness, which is due to the limitations imposed by the lattice structure (for a more in-depth explanation of numerical results the reader is referred to [5, 7]).

4 CONCLUSIONS

Three different methods to reduce computational cost related to the modelling of lattice structures have been presented. The MIF method can be applied to practically any solution algorithm that solves lattice structures and reduces significantly the number of computations due to the fact that it usually requires to decompose the global stiffness matrix only once at the beginning of the FE analysis.

The SSC algorithm can be used easily for lattice models with links that are capable of strain softening. However, when plastic compressibility is included, which is not described in this paper, the solution algorithm becomes more complex and less effective.

The linking of lattice blocks with linear elastic elements is another simple way of reducing the overall number of DoF. The fracture path has to be known in advance. One drawback is that heterogeneity can be incorporated only with difficulty into the areas with elastic solid elements and has not been done in this study.

In conclusion, while lattice models represent a challenge regarding the high demand in computational power, the issues can be overcome by a) relative simple means such as rearranging lattice nodes and using hybrid models and b) developing more sophisticated solution algorithms.

REFERENCES

[1] Landis E. N., Vasic S., and Davids W. G.: Coupled experiments and simulations of microstructural damage in wood. Experimental Mechanics, 42(4): 389–394, 2002.

[2] Fournier C. R., Davids W. G., Nagy E., and Landis E. N.: Morphological lattice models for the simulation of softwood failure and fracture. Holzforschung, 61, 360–366, 2007.

[3] Smith I., Snow M., Asiz A. and Vasic S.: Failure mechanics in wood-based materials: A review of discrete, continuum, and hybrid finite-element representations. Holzforschung, 61, 352–359, 2007.

[4] Snow M. A.: Fracture Development in Engineered Wood Product Bolted Connections. PhD thesis, University of New Brunswick, 2006.

[5] Reichert T. and Ridley-Ellis D.: 3D lattice model for post-yield and fracture behaviour of timber. In: 10th World Conference on Timber Engineering WCTE, Miyazaki, Japan, 2008.

[6] Jirasek M. and Bazant Z. P.: Macroscopic fracture characteristics of random particle systems. International Journal of Fracture, 69(3), 201–228, 1995.

[7] Reichert T.: Development of 3D lattice models for predicting nonlinear timber joint behaviour. PhD-thesis, Edinburgh Napier University, UK, 2009.