BESLE: Boundary Element Software for 3D Linear Elasticity

Andres F. Galvis[1,2*], Daniel M. Prada[1], Lucas S. Moura[1],
Cecilia Zavaglia[1], Jamie M. Foster[2], Paulo Sollero[1], Luiz C. Wrobel[3,4]

[1]*School of Mechanical Engineering, University of Campinas, Campinas* 13083-860, SP, *Brazil.*
[2]*School of Mathematics and Physics, University of Portsmouth, Portsmouth,* PO1 2UP, UK.
[3]*College of Engineering, Design and Physical Sciences, Brunel University London, Uxbridge* UB8 3PH, UK.
[4]*Department of Civil and Environmental Engineering, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro* 22451-900*, Brazil*

Abstract

BESLE is the first available parallel open-source code to analyse the mechanical behaviour of heterogeneous materials using the boundary element method (BEM) in 3D and in both an elastostatic and elastodynamic setting. Unlike all the other codes that are presently available, the software presented here is capable of simulating both isotropic *and* anisotropic materials comprised of single or multiple domains. Furthermore, the user-interface has been designed to provide a convenient way for configuring simulations involving many complex material constituents. Results are described by the displacement and traction fields, also, the stress and strain tensors are available for post-processing. BESLE is largely implemented in Fortran-MPI, but some of the sub-packages are based on other languages such as C and C++, and as such a the software is best used on a multi-core architecture where the parallelisation can be fully exploited. The main features and functionality of BESLE are presented here, and the User's Guide, available from the repository listed below, gives further details and outlines how users can carry out bespoke simulations.

Keywords: Boundary element method, 3D Linear elasticity, Heterogeneous

*Corresponding author
Email address: andres.galvis@port.ac.uk (Andres F. Galvis[1,2])

April 22, 2021

materials, Isotropy, Anisotropy.

Program summary

Program Title: BESLE
CPC Library link to program files: (to be added by Technical Editor)
Developer's repository link: https://github.com/Afgr1087/BESLE_v1.0.git
Licensing provisions: GPL-v3.0
Programming language: Fortran 90, C++/C
External libraries: LAPACK [1], BLAS [2], SCOTCH [3], ScaLAPACK [4], MUMPS [5], Voro++ [6], Triangle [7]
Nature of problem: The solution of 3D elasticity models of heterogeneous materials is often arduous owing to the complexity of the underlying system of partial differential equations. Some analyses that require intensive computation are solids under quasi-static, inertial, and high-rate loading (all of which are treated by BESLE).
Solution method: BESLE provides a strategy to configure and solve complex problems of 3D heterogeneous solids using the elastostatic and elastodynamic formulations of the boundary element method (BEM). Moreover, it provides flexible means create surface meshes, to impose both Neumann and Dirichlet boundary conditions, and comes with a material database for fast parameterisation. It allows large scale problems to be treated in a straightforward framework. The mechanical behaviour of isotropic and anisotropic bodies which can include several domains, each comprised of heterogeneous materials with a diverse range of constitutive properties and complex morphologies, are feasibly analysed.
Restrictions: Fortran is limited in the size of arrays that can be allocated. Thus, for very large problems the number of degrees of freedom in the discretised system can exceed that allowed in Fortran, even if the machine has a large amount of RAM. To avoid over-allocation we have added functionality to check whether BESLE, and it's underlying Fortran code, will be able to handle the requisite number of degrees of freedom in advance of a calculation being executed. In the event that this limitation will be approached, BESLE will return an error message advising that the number of entries required will exceed 70% of the limit imposed by Fortran.

[1] LAPACK: http://www.netlib.org/lapack/

[2] BLAS: http://www.netlib.org/blas/

[3] SCOTCH: https://gforge.inria.fr/projects/scotch/

[4] ScaLAPACK: http://www.netlib.org/scalapack/

[5] MUMPS: http://mumps.enseeiht.fr/

[6] Voro++: http://math.lbl.gov/voro++/

[7] Triangle: https://www.cs.cmu.edu/~quake/triangle.html

1. Introduction

The boundary element method is a powerful numerical tool for solving systems of partial differential equations in applications such as Laplace problems, linear elasticity, fluid dynamics, and acoustics [1–3]. It facilitates the evaluation of the domain behaviour by defining a boundary value problem which involves only the surface information, even in 3D spaces. In the modelling of linear elasticity applications, BEM allows the modelling of complex heterogeneous structures composed of isotropic or anisotropic domains. Some additional desirable features include the possibility to assess high-gradient mechanical fields and accurate solutions at material points within the domain. The BEM has been successfully implemented in homogenization, failure and multiscale analyses in polycrystal aggregates [4–9], composites [10–12] and bone tissue [13, 14] and has been shown to be a versatile and adaptable method for realistic continuum modelling.

As far as the authors are aware, all existing BEM implementations are in-house and not open-source codes. Furthermore, it is well known that BEM formulations require a robust implementation due to possible singularities. This, and the mandatory use of a fundamental solution are likely the main reasons that little effort has been dedicated to developing a commercial *computer aided design* (CAD) software based on boundary element analyses. The motivation for developing the **B**oundary **E**lement **S**oftware for 3D **L**inear **E**lasticity (BESLE) is to provide a computational package, available for simulating a variety of 3D customised elasticity models.

BESLE is the first open-source software based on boundary elements for applications in 3D linear elasticity. Classical limitations of the BEM stem from the high numbers of degrees of freedom (DOF) and that the incidence matrices are fully populated non-symmetrical. To deal with this issue an MPI parallel algorithm described in [5, 15] was incorporated allowing for an efficient solution. For this purpose, it is also possible to use adaptive cross approximation, also known as hierarchical matrices [16–18] and other high-performance BEM implementations, such as those in [19, 20]. However, in BESLE we adopt the strategy of finding an adequate balance between the number of DOFs and

subdomains under consideration and this mean that very large scale problems are now feasible to be treated without any additional arduous method.

A variety of simulation types can be configured in BESLE, namely; i) static, quasi-static, dynamic, and transient regimes, using ii) morphologies with a single or multiple domains with iii) isotropic and anisotropic materials. Moreover, it supports datasets of Dirichlet and Neumann boundary conditions and material elastic properties. Building of 3D surface meshes, composed of single or multiple domains is a complex task and not many surface generators are available. For this purpose, BESLE includes a module to generate the mesh and boundary conditions based on 3ds Max[1]. Additionally, materials containing cracks can also be modelled. However, in this version, failure criteria and propagation algorithms were not included in the implementation. Therefore, stress concentration and stress intensity factors (SIF) can be analysed. Results of SIF for mode-I were reported in [12, 21] using BESLE and Ncorr[2], proposed as an adequate approximation without additional implementations.

BESLE has been used in several applications, and is a product that leverages the work in [15], which focused on treating failure in 3D polycrystal aggregates. It was the first attempt to combine BEM and molecular dynamics within a multiscale approach [9]. Composites and adhesive bonded joints [12, 21] were analysed using BESLE coupled to a generalized energy failure criterion [22] to capture different failure modes. A multiscale model to analyse healthy and osteoporotic bone tissue was developed in [13, 14], this novel approach considered the anisotropy, and BESLE was used to evaluate the effective elastic properties using elastostatic homogenization spanning the relevant length scales. Currently, applications such as 3D printing of short fibre composites, design of bone prostheses, bone failure modelling, among others, are being explored using BEM and its surface capabilities.

The formulation used in the implementation of BESLE was described in detail in [15]. It uses three-node discontinuous triangular boundary elements, with linear interpolation. The inertial effects are evaluated using the dual reciprocity BEM with the particular solutions provided in [23]. The fundamental solutions based on the double Fourier series [24] give the opportunity to generate a separated database of material coefficients, ready to be used by the main BEM package. The solver of the resulting system of equations is a critical module, and

[1] 3D modelling and rendering software for design visualisation, games, and animation.
[2] 2D digital image correlation MATLAB program.

here we use the 64-bit parallel version of the Multifrontal Massively Parallel Sparse (MUMPS) [25, 26] solver. With this parallel version of MUMPS, matrix assembly is optimised and this facilitates assembly of the overall system of equations directly distributed through the threads in a compact sparse format. Additionally, other parallel sections such as the calculation of the incidence matrices, and the interfaces between domains contribute significantly to the performance of the overall software.

The parallel software is available to be executed in computer and server Ubuntu versions. In general, the main BEM package is supported by five sub-codes: i) material database, ii) solver, iii) general mesh generator based on 3ds Max, iv) polycrystalline structure generator, and v) box mesh generator based on MATLAB. The user should build BESLE and the external utilitarian sub-packages following the instructions in the `README` and `INSTALL` files. The main software and each external sub-package contain the `Set_parameters.*` script that serves as a user interface to configure the simulations with reserved commands. The user is encouraged to read the User's Guide and to explore the examples therein. Finally, for post-processing, `*.vtk` files are exported ready to visualise in ParaView.

The complexity of the BEM formulation and its implementation has been an obstacle in the development of a commercial BEM software, and this is reflected in the widespread adoption of the other numerical methods such as the finite element method. Nowadays, the increased computational power available on moderns clusters or workstations allows the solution of systems with large number of DOFs, and some BEM packages are beginning to appear. The FastBEM [27] is a package based on the fast multipole method (FMM) for thermal analysis, 2D and 3D elasticity, stokes flow, and acoustics. It solves general 3D linear elasticity problems with homogeneous and isotropic materials, which is accelerated using the FMM, ACA, and fast direct solvers. The BELT [28] provides generic building blocks for the Galerkin discretization of a large class of boundary integral operators in 3D. It can solve differential equations such as Laplace's, Helmholtz', and Maxwell's equations as well as linear elastostatics for isotropic materials. BEM++ [29] is an open-source Python library for the solution of 3D boundary integral equations of Laplace, Helmholtz, and Maxwell operators. Developed by the same research group as BEM++, the Bempp [29] is an open-source computational boundary element platform to solve electrostatic, acoustic, and electromagnetic problems. Finally, the recent Bembel [30] code is a C++ library featuring higher-order isogeometric Galerkin boundary element methods for Laplace, Helmholtz, and Maxwell problems.

Despite the range of features offered by these other pieces of software, BESLE is the first offering that is able to simulate 3D linear elasticity problems for isotropic and anisotropic materials configured under a variety of regime options.

2. Boundary element formulation

BESLE is based on the boundary element formulation described in [15] that uses the well-known boundary integral equation (BIE), to treat 3D elasticity problems physically described by continuum mechanics and infinitesimal strain theory using the following equations

$$\frac{\partial \sigma_{ij}}{\partial x_j} + \rho b_i = \rho \ddot{u}_i, \quad \varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \tag{1}$$

as well as the well-known linear elastic constitutive equation

$$\sigma_{ij} = C_{ijkl}\varepsilon_{kl} \quad , \tag{2}$$

where σ_{ij} is the stress tensor within the solid. The strain tensor ε_{ij} is a function of the displacements u_i in Eq. (1) which is valid for small strain. Finally, C_{ijkl} is the stiffness tensor of material properties. In Eq. (1), ρ is the mass density, ρb_i is body force vector and $\rho \ddot{u}_i$ represents inertia.

The central approach in the BEM is to construct the solution to the problem of interest by first finding the solution to the governing equations in an infinite medium, and in a situation where the inertial and body force terms are taken to be a point force located at an arbitrary "source" point (x'_1, x'_2, x'_3), which, without loss of generality can be selected to be the origin [1, 3, 31, 32], i.e., we solve Kelvin's problem. We eliminate σ_{ij} and ε_{ij} between equations (1) and (2) and then set $\rho(\ddot{u}_i - b_i) = -\delta(x'_1)\delta(x'_2)\delta(x'_3)f_i$ such the problem that we seek to solve is given by, $\bar{u}_i$, which satisfies

$$C_{ijkl}\bar{u}_{k,lj} = -\delta(x'_1)\delta(x'_2)\delta(x'_3)f_i \quad . \tag{3}$$

where $\delta(x_i)$ denotes the Dirac delta function. The solution to this problem can be obtained via the Green's function, or fundamental solution, $\mathbf{U}(\mathbf{x}, \mathbf{x}')$, presented by Ting and Lee [33]. We have

$$\bar{\mathbf{u}} = \mathbf{U}(\mathbf{x}, \mathbf{x}')\mathbf{f} \quad . \tag{4}$$

The indicies i and j on U_{ij} can be interpreted as the excitation to the x_i-component of the displacement at $\mathbf{x}$ when a perturbation in the x_j-direction is

applied at $\mathbf{x}'$. Hence, following the procedure described in [1, 3], after some manipulation of Eqs. (1) and (2) using the principle of virtual work and the Somigliana's identity, the BIE is given by

$$c_{ij}u_i + \int_\Gamma T_{ij}u_i \,\mathrm{d}\Gamma = \int_\Gamma U_{ij}t_i \,\mathrm{d}\Gamma - \int_\Omega \rho(b_i - \ddot{u}_i)U_{ij} \,\mathrm{d}\Omega \quad , \tag{5}$$

Note that inertial effects are accounted for in the BIE in the final term on the right-hand side. It is a relationship between the as yet unknown displacements, u_i, and tractions, t_i, on the surfaces between the different materials, and it involves the fundamental solutions U_{ij} and T_{ij}, where T_{ij} is the fundamental solution for the traction that can be computed from that for the displacement (U_{ij}) using Cauchy's theorem. We note that T_{ij} inherits knowledge about the elastic properties of the medium C_{ijkl}. In the BIE, Eq. (5), $c_{ij} = \delta_{ij}/2$ for a source point located on a smooth part of the surface boundary, where δ_{ij} the Kronecker delta. In BESLE, the fundamental solutions based on double Fourier series [24] are implemented which offers the advantage to create a separate material database composed of the Fourier coefficients and elastic properties for different isotropic and anisotropic materials.

In situations where inertial forces, body forces, or external loads mean that the domain integral on the right-hand side of (5) cannot be justifiably neglected, it is required to transform this term into a boundary or surface integral through the use of the dual reciprocity boundary element method (DRBEM) described in [34, 35]. Following the procedure described in [23] the general BIE, Eq. (6), in terms of only surface integrals is

$$\begin{aligned} c_{ik}u_i + \int_\Gamma T_{ik}u_i \mathrm{d}\Gamma = \int_\Gamma U_{ik}t_i \mathrm{d}\Gamma + \\ \sum_{j=1}^{M} \alpha_n^j \left\{ c_{ik}\hat{u}_{kn}^j - \int_\Gamma U_{ik}\hat{t}_{kn}^j \mathrm{d}\Gamma + \int_\Gamma T_{ik}\hat{u}_{kn}^j \mathrm{d}\Gamma \right\} \quad . \end{aligned} \tag{6}$$

In Eq. (6), α_n^j represents unknown coefficients that are functions of radial expressions related to the body force term $\rho\ddot{u}_i$, see details in [6, 36]. The $\hat{u}_{in}^m$ term is called a particular solution, and in BESLE, and many other studies in the literature, it is a radial function [35] written as

$$\hat{u}_{kn} = \delta_{kn}\left(r^2 + r^3\right) \quad , \tag{7}$$

from which, and via the use of Cauchy's theorem, the particular solution $\hat{t}_{kn}$ can be evaluated.

2.1. Spatial discretisation

BESLE applies a discretisation of the surface into linear three-node discontinuous triangular boundary elements [15]. They are linear to gain a drastic reduction in the number of DOF, and discontinuous due to the advantages in the implementation of the multidomain algorithm. Applying the discretisation procedure showed in [15] and widely described in the User's Guide to Eqs. (5) and (6), the BIE expressed in matrix form for elastostatic problems including body and inertial forces is

$$\mathbf{Hu} = \mathbf{Gt} + \mathbf{Mf} \quad , \tag{8}$$

and in situations where body forces and interia are negligible we can neglect the final term, **Mf**, so that this reduces to

$$\mathbf{Hu} = \mathbf{Gt} \quad . \tag{9}$$

For transient analysis when high-rate boundary conditions are applied, the equation becomes

$$\mathbf{M\ddot{u}} + \mathbf{Hu} = \mathbf{Gt} \quad . \tag{10}$$

2.2. Numerical integration

The surface is discretised into linear three-node discontinuous elements, shown in Fig. 1. In this element, the three nodes are functions of the two intrinsic parametric coordinates, namely (η, ξ), and thus the position of all nodes in the element can be characterised by a parametric distance from the origin, λ.

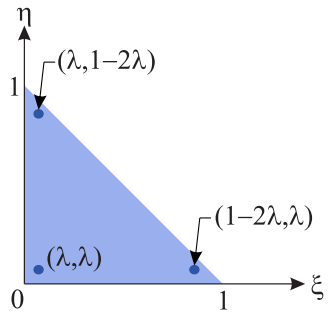


Figure 1: Linear three-node discontinuous element.

We tested different values of λ and found that for a representative trial simulation the value of $\lambda = 0.155$ (i.e. `lambda = 0.155d0`) gave the most

accurate results. As such this is the default value in BESLE but users are free to modify this should they wish. The surface response is linearly interpolated within an element from the nodal values, following the procedure explained in the User's Guide and illustrated in [1, 15].

In BESLE, the integration of the regular and singular kernels in Eq. (6) is based on the Gaussian quadrature. The displacement fundamental solution U_{ij} contains an integrable, weak singularity $(1/r)$ at the origin which must be integrated to obtain the diagonal entries in the matrix $\mathbf{G}$. At such points, where the source and field points coincide, we require a special quadrature method designed to accurately resolve this singularity. In BESLE we follow the procedure recommended by Kane [1] in which the Gauss points must be concentrated close to the singular node as shown in Figs. 2 for the nodes at $(1-2\lambda,\lambda)$ and $(\lambda,1-2\lambda)$.

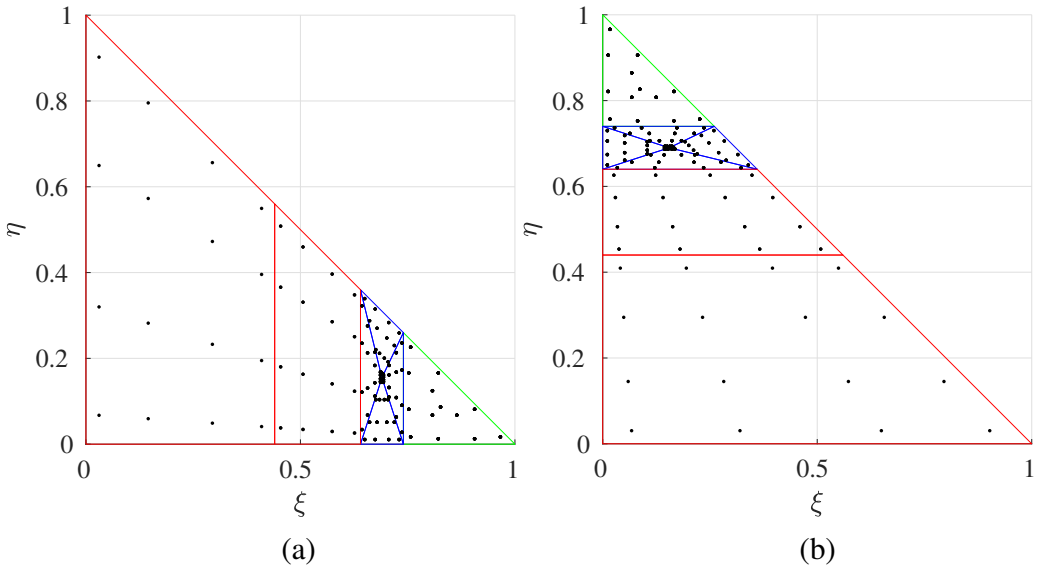


Figure 2: Distribution of Gauss points for the integration of singular kernels in the isoparametric element: (a) nodes at $(1-2\lambda,\lambda)$, and (b) nodes at $(\lambda,1-2\lambda)$.

Figure 3 illustrates the distribution of the integration potins for the node at (λ,λ). Full details of this integration approach are given in the User's Guide (chapters 2 and 5).

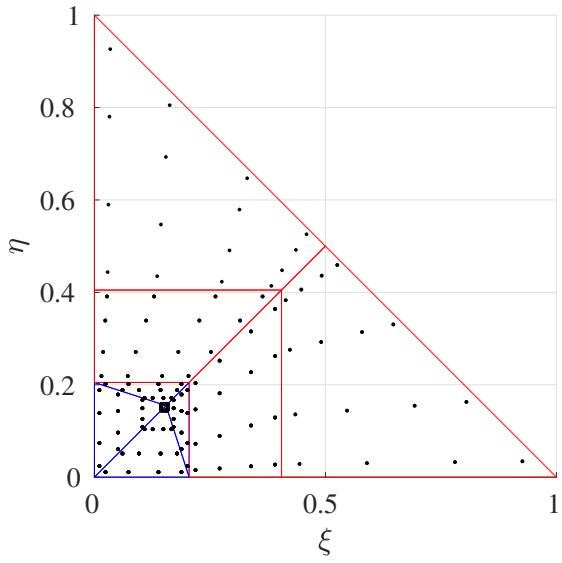


Figure 3: Distribution of Gauss points for the integration of singular kernels in the isoparametric element when the source and field points coincide at node (λ,λ).

The fundamental solution for tractions T_{ij} contain a strong singularity $(1/r^2)$ at the origin and this also needs to be integrated in order to obtain the diagonal elements of the matrix $\mathbf{H}$. Once again a bespoke approach to integration is required and here we take advantage of the well-known rigid body motion concept [1, 31, 37] to alleviate the problem.

2.3. Temporal integration

We used Houbolt's algorithm [38, 39] in order to relate the solution at the present time step to those in the past. The acceleration term in Eq. (10) is approximated using the following finite difference formula

$$\ddot{\mathbf{u}}_{\tau+\Delta\tau} = \frac{1}{\Delta\tau^2}\left(2\mathbf{u}_{\tau+\Delta\tau} - 5\mathbf{u}_{\tau} + 4\mathbf{u}_{\tau-\Delta\tau} - \mathbf{u}_{\tau-2\Delta\tau}\right) \quad , \tag{11}$$

the new equation for the transient problem at the time step $\tau+\Delta\tau$ is given by

$$\left[\frac{2}{\Delta\tau^2}\mathbf{M} + \mathbf{H}\right]\mathbf{u}_{\tau+\Delta\tau} = \mathbf{G}\mathbf{t}_{\tau+\Delta\tau} + \frac{1}{\Delta\tau^2}\mathbf{M}\mathbf{u}_{\alpha} \quad , \tag{12}$$

where $\mathbf{u}_{\alpha}$ is a linear combination of the deformation vector at the three previous time steps defined as follows

$$\mathbf{u}_{\alpha} = 5\mathbf{u}_{\tau} - 4\mathbf{u}_{\tau-\Delta\tau} + \mathbf{u}_{\tau-2\Delta\tau} \quad . \tag{13}$$

The incidence matrices, $\mathbf{G}$ and $\mathbf{H}$, are obtained using the collocation method [1, 31]. They result from the integration of the traction and displacement fundamental solutions, respectively. The mass matrix, $\mathbf{M}$, contains the mass density ρ [35]. The vectors $\mathbf{u}$ and $\mathbf{t}$ are the displacement and traction fields, respectively. At each node we have 3 DOF from the displacements and a further 3 from the tractions. In a well-posed BEM problem, half of these DOFs must be known and are typically specified by the boundary conditions, and the body forces vector $\mathbf{f}$ should be known at every nodal point. In situations where the tractions are specified and the displacements are to be solved for, the corresponding rows in $\mathbf{H}$ in (12) need not be manipulated, however, if the displacements are specified it is necessary to exchange the columns in matrix $\mathbf{H}$ with the corresponding columns in $\mathbf{G}$ for the unknown tractions. On doing so Eq. (8) becomes

$$\mathbf{A}\mathbf{x} = \mathbf{B}\mathbf{k}^{bc} + \mathbf{M}\mathbf{f} \quad . \tag{14}$$

Carrying out an analogous manipulation in the cases of elastostatics without inertial and body forces, (9) and in the transient setting, (12), results in

$$\mathbf{A}\mathbf{x} = \mathbf{B}\mathbf{k}^{bc} \quad , \tag{15}$$

$$\left[\frac{2}{\Delta\tau^2}\mathbf{M} + \mathbf{A}\right]\mathbf{x}_{\tau+\Delta\tau} = \mathbf{B}\mathbf{k}^{bc}_{\tau+\Delta\tau} + \frac{1}{\Delta\tau^2}\mathbf{M}\mathbf{u}_{\alpha} \quad , \tag{16}$$

respectively. Here, $\mathbf{k}^{bc}$ is a vector containing all the known boundary conditions, $\mathbf{x}$ corresponds to the unknown fields, $\mathbf{A}$ and $\mathbf{B}$ are matricies resulting from the exchange columns between $\mathbf{H}$ and $\mathbf{G}$. Equations (14)-(16), each comprise a linear system of equations. In transient problems information about the solution at the last three-time steps are encoded as source terms as shown in Eq. (12). For all analyses, the final system of equations is solved using MUMPS and the stress and strain fields are computed a posteriori at each time or load step following the procedure described in [1].

For heterogeneous solids with more than one constituent where the boundary conditions are imposed in the external domains, the equations are the same. However, the displacement compatibility and traction equilibrium must be applied at the interfaces between domains as follows

$$\begin{aligned} \mathbf{u}_i^j &= \mathbf{u}_j^i \quad , \\ \mathbf{t}_i^j &= -\mathbf{t}_j^i \quad . \end{aligned} \tag{17}$$

where the i and j indices represent the Ω_i and Ω_j domains. Figure 4 shows an interface, being Γ_i and Γ_j related to the surfaces, and $\mathbf{n}$ to the outward normal vector.

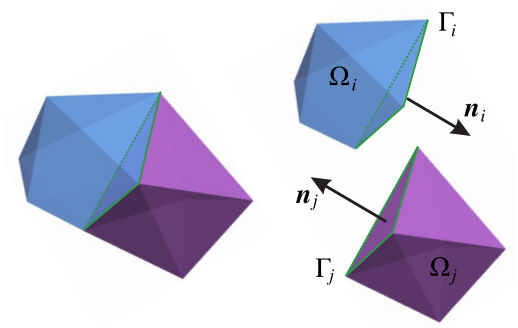


Figure 4: Interfaces.

Before we go on to discuss the structure of the software and demonstrate its functionality we note that there are alternative methods to construct BEM systems of equations, and some of these alternatives are presented in [1, 40]. This introduction is intended to serve as an overview of the approach taken in BESLE and readers/users are referred to the User's Guide for a complete description.

3. Software structure

This section will outline the structure of the BESLE software including how to generate meshes, how to define material properties, and how to assign boundary conditions. BESLE is intended for use on an Ubuntu operating system and on hardware that has more than one processor. The software dependencies are minimal and full installation instructions are given in the Users Guide.

3.1. Surface mesh generation

The surface mesh needs to be given to BESLE in the form of an input `*.dat` file as shown in Script. 1. Users are free to generate this input file in their preferred way.

```
nreg
nPoints ! matrix of POINTS
x_1  y_1  z_1
x_2  y_2  z_2
.
nElements ! matrix of ELEMENTS
node_1_el_1   node_2_el_1   node_3_el_1
node_1_el_2   node_2_el_2   node_3_el_2
.
nElements  ! matrix of NORMAL_VECTORS
n_x_el_1   n_y_el_1   n_y_el_1
n_x_el_2   n_y_el_2   n_y_el_2
.
nreg   ! El_reg matrix
nElements_reg_1
nElements_reg_2
.
```

Script 1: Mesh file structure. Here, `nreg` is the number of regions, the matrix of `POINTS` contains the spatial coordinates of the mesh nodes in double precision whilst the matrix of `ELEMENTS` contains the connectivity of those nodes. Furthermore, the matrix of elements per region `EL_reg` and the matrix of `NORMAL_VECTORS` of each boundary element. `nreg`, `ELEMENTS`, and `EL_reg` are interger variables. The other are variables defined in double precision.

For simple meshes constructing the mesh files by hand might be an option, however, we provide a description of some tools which users might like to explore to mitigate the time required to generate complex meshes. In subsections 3.1.1 and 3.1.2 we outline how users can import 3ds Max meshes and meshes from the program Polycrystal respectively for use in BESLE. It is also possible to generate simple meshes using MATLAB and some details on doing this are given in the User's Guide.

BESLE includes three methods for mesh generation. First, a `Fortran` sub-package based on 3ds Max that can define general meshes and boundary conditions for several cases. Second, a `C++/C` sub-package is available to artificially reproduce polycrystal aggregates. Finally, a basic `MATLAB` box generator is also provided. After the construction of the mesh, `*.obj` or `*.vtk` files are available to be checked using `ParaView`.

3.1.1. Generating meshes using 3ds Max

Surface meshes can be generated using multimedia software like 3ds Max (Autodesk) or Blender [13, 14]. Usually these packages can export the mesh of

a solid in a `*.obj` file as a standard. In BESLE, it is provided a `General` mesh generator implemented using `Fortran`. The main function of this sub-package is the transformation of the `*.obj` into the `*.dat` structure mesh file shown in the Script 1.

Additionally, the user can export `*.obj` files corresponding to each portion of the surface where any boundary condition is applied. Then, in the `General` sub-package, the user can also configure the desired boundary conditions from a large variety of possibilities which will be exported into the `BCs_i.dat` files. In summary, all these `*.dat` files for the mesh and boundary conditions are the inputs that BESLE reads at the beginning of a simulation. As an example, a vertebra L2 mesh is shown in Fig. 5.

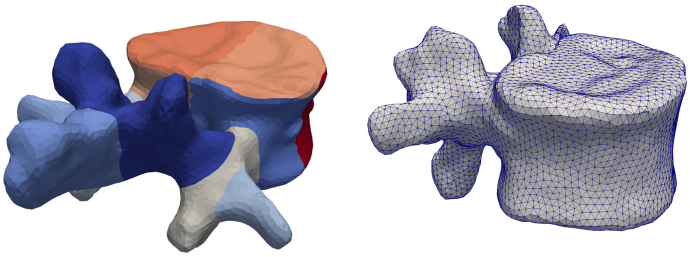


Figure 5: Vertebra L2: mesh of 21612 boundary elements.

Firstly, the L2 vertebra was segmented from a fully anonymized computerised tomography (CT) of the 3DSlicer database[3] [41]. Thereafter, it was exported as `*.stl` and finally ajusted into 3ds Max. The model of the L2 vertebra was divided into trabecular and cortical bone by segmenting the trabecular tissue from the CT and then, superimposing it with the vertebra model. The anisotropic properties of cortical bone could be oriented with respect to the normal direction of the bone surface. To include this in the model, the cortical domain was split into 15 new domains and a representative material orientation of each region was evaluated using the mean of the outward normal direction of the elements in each region.

3.1.2. Generating meshes using Polycrystal

Polycrystalline materials are represented by the aggregate of several crystals or grains each with their own orientations and crystal planes. At the microscale,

[3]https://www.slicer.org/wiki/CitingSlicer

each grain is usually modeled as a linear elastic anisotropic material as presented in the majority of metallic and ceramic crystalline materials. This `Polycrystal` mesh generator is a `C++/C|` multi-compiler code based on the procedure presented in [5, 15, 42] for generating artificial polycrystalline structures. This sub-package requires as input parameters the number of grains in each direction, it means the number of grains is given by `ngrains_x*ngrains_y*ngrains_z`. Furthermore, all polycrystals are generated in a box shape, then, the size is defined by three double variables `x_max`, `y_max`, and `z_max`. Figure 6 presents a polycrystalline material constituted by 96 grains in a box with the dimensions $L_x = 15\,\mu$m, $L_y = 15\,\mu$m, and $L_x = 45\,\mu$m.

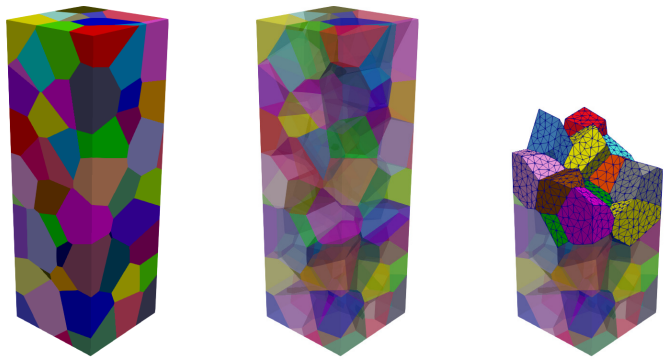
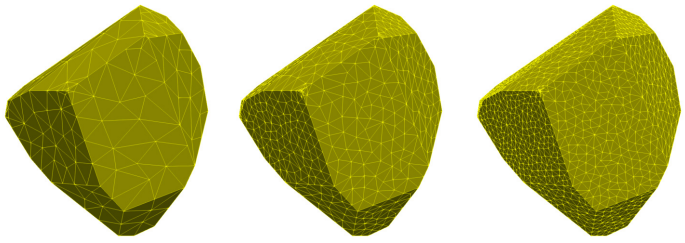


Figure 6: Polycrystal aggregate, 96 grains and 22767 boundary elements.

Additionally, `Polycrystal` can vary the mesh density throughout the `dm` integer parameter `dm` that takes a minimum value of `1`. In order to illustrate the effect of this parameter, in Fig. 7, three cases are shown for different values of `dm`. It can be appreciated how the refinement increases with the increment of `dm`.



(a) (b) (c)

Figure 7: Mesh density parameter, (a) d_m = 1, 399 elements; (b) d_m = 2, 1478 elements; and (c) d_m = 3, 3250 elements.

Finally, the mesh can be checked in `ParaView` with the `Mesh.vtk` file. The boundary conditions to analyse the mechanical behavior of these materials are defined directly in BESLE in the user's script.

3.2. Elastic properties using Material

Several options for modelling elastic materials are implemented in BESLE, it is possible to consider isotropic or anisotropic materials in specimens with only one or multiple domains. Furthermore, for heterogeneous materials, useful miscellaneous lists were included to facilitate the modelling of specimens with a large number of domains. First, a background of the fundamental solutions is presented, and then, the computational aspects and guidance for running the `Material` sub-package are also illustrated.

BESLE requires, as material input parameters, the Fourier coefficients, and the stiffness tensor that are generated in a separated database. In order to compute the fundamental solutions for displacements U_{ij} and tractions T_{ij} the parameters that need to be exported from the `Material` sub-package are shown in Eq. (18). The formulation to get these quantities is presented in subsection 4.1 of the User's Guide.

$$\begin{gathered} \tilde{R}_{uv}^{(m,n)}, \quad \hat{R}_{uv}^{(m,n)}, \quad \tilde{I}_{uv}^{(m,n)}, \quad \hat{I}_{uv}^{(m,n)}, \quad R_{uv}^{(0,m)}, \quad I_{uv}^{(0,m)}, \\ R_{uv}^{(m,0)}, \quad I_{uv}^{(m,0)}, \quad R_{uv}^{(0,0)}, \quad C_{ijkl} \quad . \end{gathered} \tag{18}$$

The algorithm of this `Fortran` sub-package is explained in more detail in [5, 15]. It uses a listing scheme for some special cases, where the user can define several materials from external files that contain lists of material properties. Figure. 8 is provided for a better illustration. The package is desined to satisfy several material configurations required for engineering applications. As observed, there are two groups for generating Fourier coefficients, for single $N_{Mat} = 1$ or multiple $N_{Mat} > 1$ domains.

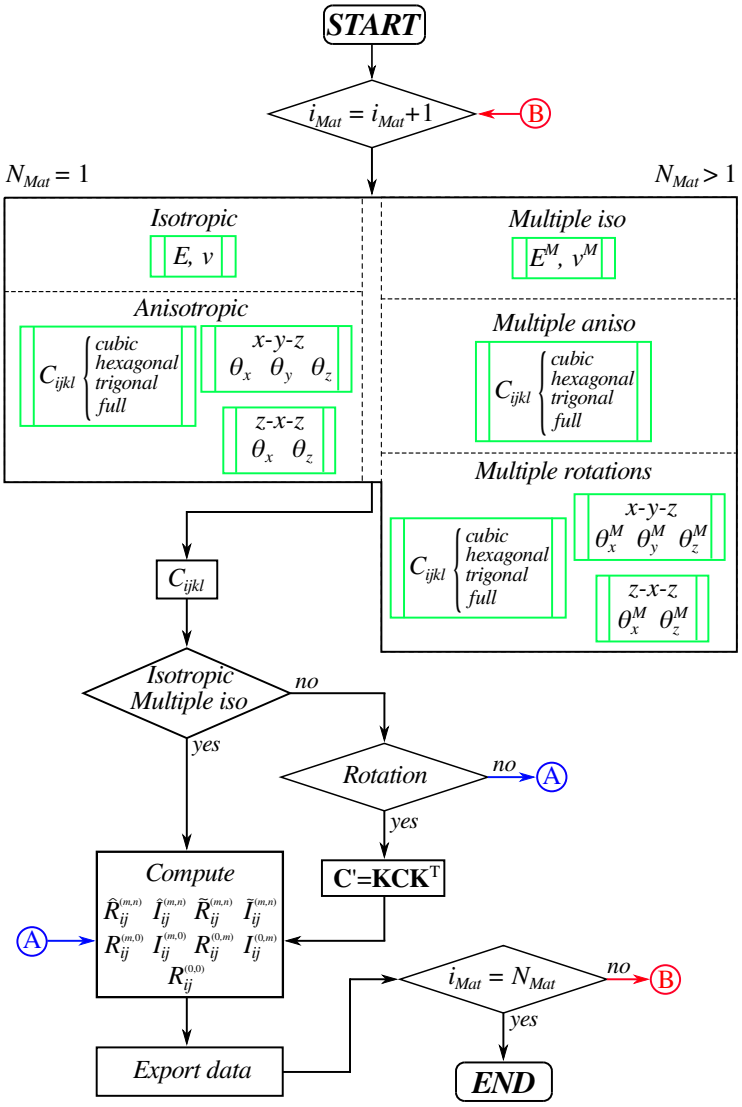


Figure 8: Material database algorithm.

For a single domain $N_{Mat} = 1$, the first configuration is the isotropic case, where the idea is to obtain the data for materials defined by only two elastic properties, the Young's modulus, and Poisson ratio. For anisotropic materials, the code supports different Lattice structures such as cubic (BCC and FCC), hexagonal (HCP), trigonal, and the full anisotropic case. If the material is rotated to an specific orientation, by default the code proceeds with the $x-y-z$ convention. However, an additional $z-x-z$ rotation convention was also

included [5].

Now, consider an heterogeneous material with multiple domains or constituents, then $N_{Mat} > 1$. The first case as shown in Fig. 8, is for a material composed of multiple isotropic domains. For this a list containing the Young's moduli E^M and Poisson ratios ν^M is read, where M is the superscript that corresponds to each domain or material. The second case is the multiple anisotropic materials, for facility this list comprises the 21 components of the stiffness tensor for each C^M_{ijkl} tensor. Finally, the multiple rotations with a list of angles for the same anisotropic material is included.

This sub-package offers the advantage to create large material databases, that could be used at any time reducing the computational cost for simulations that need to be run more than once. This is accomplished through the functionalities to reproduce materials composed of different constituents using the several list schemes presented in the examples.

3.3. Boundary element algorithm

The main implementation of BESLE corresponds to the algorithm of the boundary element method that is coupled to the external sub-packages for solving the 3D elasticity problems. For illustrative purposes, Fig. 9 shows the general BEM algorithm used in BESLE. After the correct installation and link of the external libraries, the configuration starts with the definition of the solid `Surface Mesh`. As mentioned, for this, there are three available options such as `General`, `Polycrystal` and `Box`. However, the user is free to use any other surface mesh generator if the requirements of the input format shown in Script 1 are satisfied.

The `General` sub-package comes with an additional option for generating custom boundary conditions (`Custom BCs`), the reader is referred to the User's Guide for a better explanation of all the option of boundary conditions provided by this sub-package. Nonetheless, if the specimen has a box shape, BESLE includes a straightforward way to configure the boundary conditions manually (`Standard BCs`) within the main setup. The remaining `User Parameters` to be defied in the `Setup` are the link directions of the external sub-packages (mesh and `Material`), and regime analysis such as: i) quasi-static $\mathcal{Q}$-$\mathcal{S}$, ii) dynamic $\mathcal{D}$ or iii) transient $\mathcal{T}$.

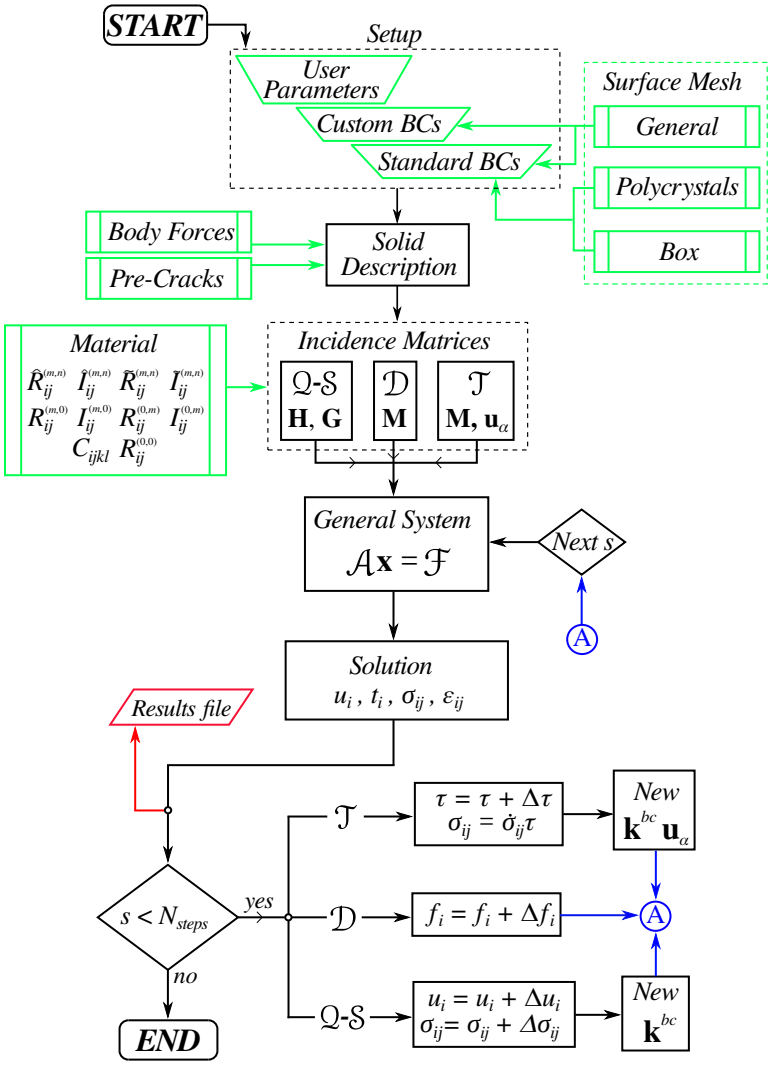


Figure 9: BESLE structure.

If a simulation of `Body Forces` varying throughout the time or load step is required, the user can define an external list of values in the three directions (x, y, z) for this force. Furthermore, `Pre-Cracks` could be imposed in a model with more than one domain. The user defines virtual boxes across the specimen in which all the interfaces contained within a specific box will be separated before the simulation. At every step, a results `*vtk` file is exported. In ParaView, the user can get the visualisation of the displacement u_i field, the total displacement u_t, stress σ_{ij} tensor, strain ε_{ij} tensor, and the von Mises σ_{VM} stress.

Two main error situations were included in the debug process. Due to the allocation limit of Fortran and depending on the available RAM, for really large problems the number of degrees of freedom could exceed its limit that was established approximately as $0.70 \cdot (nthreads) \cdot (2^{31} - 1)$ non-zero entries of the main general system of equations, the matrix $\mathcal{A}$, see Fig. 9. In this case, a message will appear indicating the number of additional DOF in each rank. To solve this, the user must run the simulation again increasing the number of threads if available, or reduce the mesh if possible. The number of non-zero entries of the matrix $\mathcal{A}$ ($nZ\mathcal{A}$) can be previously evaluated using the following equation

$$nZ\mathcal{A} = 81 \left\{ \sum_{j=1}^{N_{reg}} \left(nEl_j \right)^2 + \sum_{j=1}^{N_{reg}} \left(nEl_j \cdot nInt_j \right) \right\} \quad , \tag{19}$$

where nEl_j is the number of elements, and $nInt_j$ represents the number of interfaces in the Ω_j domain. A second error occurs when there are overlap elements or repeated nodes that make the minimum segment of all the boundary elements be zero. In that case, the mesh must be reviewed and edited. Finally, as a recommendation and good practice, no more than 20 boundary elements must converge to the same node.

4. Validations

The software has the capability for modelling several physical problems on solids. As mentioned before isotropic and anisotropic materials can be simulated, furthermore, single and multiple domains for the case of heterogeneous media. For validation purposes, three simulation tests are presented here one problem of a body force acting on an anisotropic solid, and two transient problems of isotropic materials under Heaviside and harmonic load conditions. This serves as a check of the correct installation since the software comes ready with the configuration for simulating the harmonic case.

4.1. Body forces acting on anisotropic materials

A box specimen with dimensions $L_x = 15$ mm, $L_y = 15$ mm and $L_z = 60$ mm, Fig. 10 is considered. The material is constituted by 8 domains of the same trigonal alpha-quartz anisotropic material with six independent elastic constants $C_{11} = 87.6$ GPa, $C_{12} = 6.07$ GPa, $C_{13} = 13.3$ GPa, $C_{14} = 17.3$ GPa, $C_{33} = 106.8$ GPa, and $C_{44} = 57.2$ GPa. The orientation of the stiffness tensor is $\theta_z = -30^\circ$,

$\theta_y = -30°$, and $\theta_z = -60°$ to get a new material orientation. The specimen is fully constrained in all axes, thus $u_i = 0$ at the bottom in the Γ_5 surface. When nothing is defined by the user, the default boundary condition corresponds to a free surface $t_i = 0$ imposed on the lateral and top surfaces.

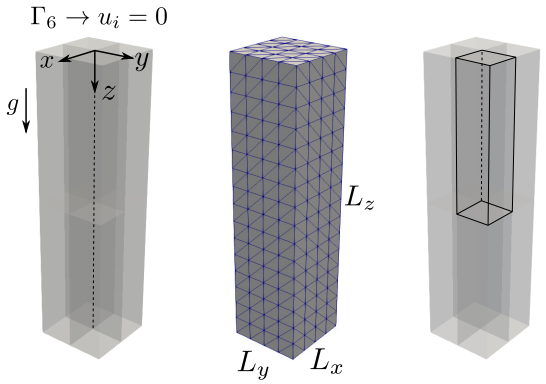


Figure 10: Box model of 8 domains discretised with 1152 boundary elements.

In this model, the effect of the body force ρg is observed, being the density $\rho = 2648\,\mathrm{ks/m^3}$ and the acceleration of gravity $g = 9.81\,\mathrm{m/s^2}$. Therefore, it is expected the results should reproduce the deformation of the solid under its own weight. The `box` generator is used to build the mesh using the `Mesh_Generator.m` script as shown in Script 2.

```
file = 'Body_force';

a = 15; % Dimension in x
b = 15; % Dimension in y
c = 60; % Dimension in z

nreg_a = 2; % Regions in x
nreg_b = 2; % Regions in y
nreg_c = 2; % Regions in z

nel_a = 2; % Elements in x / region
nel_b = 2; % Elements in y / region
nel_c = 8; % Elements in z / region
```

Script 2: Box model of 8 domains.

This is a basic `MATLAB` code for general box geometries that can be composed by one or multiple domains. If this is the first time using alpha-quartz, the

`Material` sub-package is used for generating the database file. For this purpose, the `Set_parameters.f90` module is configured, and the setup is shown in the Script 3.

```
SUBROUTINE Setup

    fileplace="Data/Anisotropic/"
    file_name="Material_4.dat"

    Material="Anisotropic"
    Lattice="trigonal"

    C11=87600.d0; C12=6070.d0;
    C13=13300.d0; C14=17300.d0;
    C33=106800.d0; C44=57200.d0;

    theta_x=-30.d0;
    theta_y=-45.d0;
    theta_z=-60.d0;

END SUBROUTINE Setup
```

Script 3: Trigonal alpha-quartz oriented with $x-y-z$ convention, units in [MPa].

Now the configuration of BESLE is run from its main user's script named as `Set_parameters.f90`. The setup for the body forces simulation is shown in the Script 4. In BESLE, the integration is carried out using Gauss-quadrature, then, this simulation two cases are validated for 28 and 52 Gauss points, the Script 4 shows the configuration of 4 triangles with 7 Gauss points each one. This model can be validated using the analytical solution obtained by Lekhnitskii [43] shown in Eq. (20).

$$\begin{aligned} u_x &= \rho g\left[-0.5a_{35}z^2+\left(a_{13}x+0.5a_{36}y\right)\left(L_z-z\right)\right] \quad , \\ u_y &= \rho g\left[-0.5a_{34}z^2+\left(a_{23}x+0.5a_{36}y\right)\left(L_z-z\right)\right] \quad , \\ u_z &= \rho g\left[-0.5\left(a_{13}x^2+a_{23}y^2+a_{36}xy\right)+\right. \\ &\qquad \left.\left(a_{34}x+a_{35}y\right)L_z+0.5a_{33}\left(2L_z-z\right)\right] \quad . \end{aligned} \tag{20}$$

The comparison is made along the the z-axis $(0,0,z)$ according to the reference coordinate system shown in Fig. 10. In Eq. (20), the a_{ij} terms correspond to the compliance tensor computed from $a_{ij}=C^{-1}_{ijkl}$.

```
1  MODULE Set_parameters
2  !-------------------------------------------------------------!
3  USE Global_variables
4  !-------------------------------------------------------------!
5  CONTAINS
6      SUBROUTINE Setup
7
8          mesh_file = 'Body_force'
9          fileplace_mesh = 'Mesh/Box/'
10         scale_size_1 = 1e-3
11
12         material_coefficients_file = 'Material_4'
13         fileplace_material = "Material/Data/Anisotropic/"
14         scale_prop_mat = 1e6
15
16         npoints_rule_NSI = 7
17         Int_type = 4
18
19         bodyforces = 1
20         bodyforce = (/0.d0,0.d0,-2.5976d0*(1e4)/)
21         density = 2648.d0
22
23         quasi_static = 1
24         static_steps = 1
25
26         box_face_6 = (/0.d0,0.d0,0.d0,0.d0,0.d0,0.d0/)
27
28         scale_results = 1.5e6
29         results_file = 'Results'
30         fileplace_results = 'Results/'
31
32     END SUBROUTINE Setup
33 END MODULE Set_parameters
```

Script 4: Body forces acting on anisotropic materials. Lines 8-10: Mesh file and scaling the solid dimensions; lines 12-14: Material database and scale the units of material properties; lines 16-17: split the boundary element into 4 triangles and used 7 Gauss-points in each one for the non-singular integration; lines 19-21: body force definitions and mass density; lines 23-24: quasi-static regime and static load steps; line 26: fully constraint condition imposed at the Γ_6 surface (see pages 46 and 47 of the User's Guide); and lines 28-30: scaling the displacement field and results definitions.

Setting $x = 0$ and $y = 0$ in Eq. (20), Fig. 11 illustrates how close the BESLE solution is to the analytical model for the horizontal displacements. The numerical solutions show a good agreement with the analytical model for both 28 and 52 Gauss points. Due to the level of anisotropy of the alpha-quartz material, the deformation caused by the acceleration of gravity is not symmetric on the $x-y$ plane of the specimen, it means the axis becomes curved.

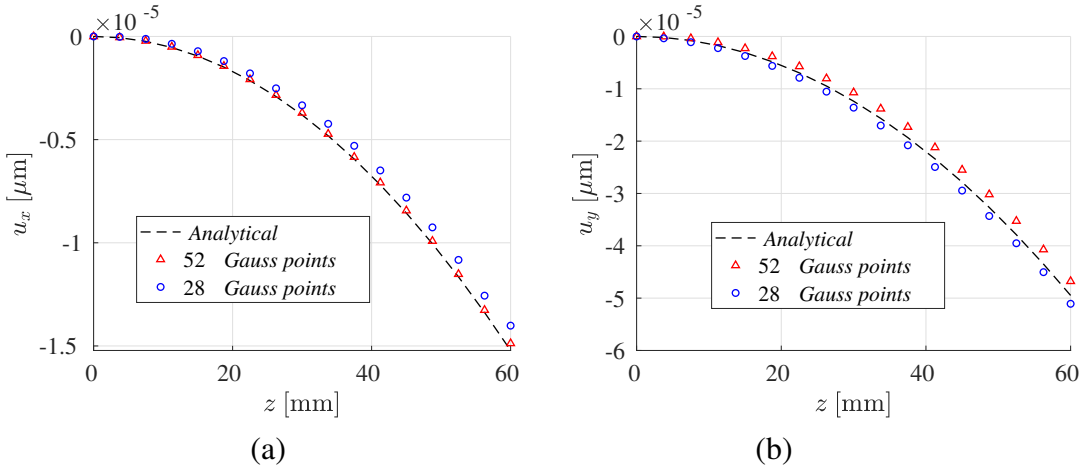


Figure 11: Displacements of z-axis, (a) u_x and (b) u_y.

From Eq. (20), it can be observed that cross-sections do not remain plane but take the shape of a second-degree polynomial surface. The vertical displacement is also compared,

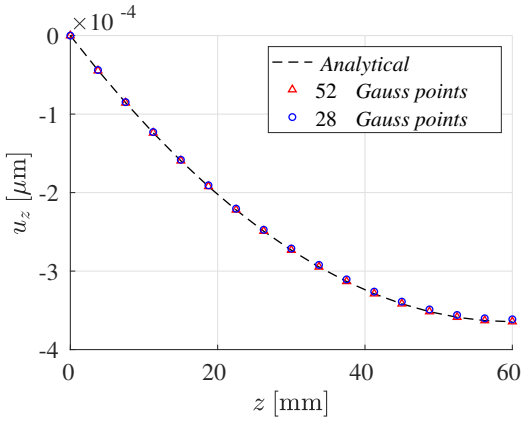


Figure 12: Vertical displacement of z-axis.

In fact, because these solutions are continuum functions, the boundary element method can evaluate accurately the material behaviour with a lower number of Gauss points. Finally, these results were extracted directly from ParaView.

4.2. Transient behaviour of isotropic materials

Consider a thin isotropic bar with dimensions $L_x = 100\,\text{mm}$, $L_y = 10\,\text{mm}$ and $L_z = 10\,\text{mm}$, Fig. 13. The elastic constants are the Young's modulus $E = 200$ MPa and Poisson ratio $\nu = 0$. The mass density is $\rho = 7850\,\text{kg/m}^3$. The bar is fully constrained in all axes $u_i = 0$ at the Γ_4 $(x = 0)$ surface. A Heaviside step load $t_1(\tau) = 100$ MPa is applied at the Γ_2 $(x = x_{max})$ surface, where τ indicates the time scale variable. In order to capture the transient regime and dynamic effects, high-rate boundary conditions are applied in a period of 0.1 ms. Hence, the simulation comprises 100 steps of $1\,\mu$s. The purpose of this simulation is to track the displacement $u_1 \equiv u_x(\tau)$ at the point $c(50,5,5)$ along the simulation time to compare it with the analytical solution.

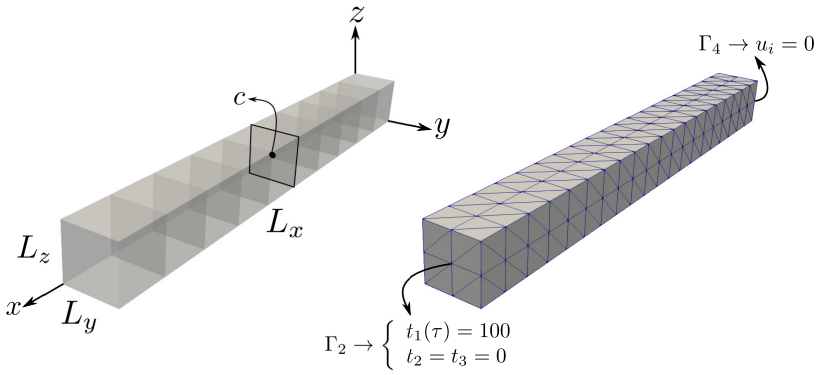


Figure 13: Box model of 10 domains discretised with 480 boundary elements.

The box generator is used to build the mesh using the `Mesh_Generator.m` script as shown in Script 5.

```
file = 'Transient';

a = 100; % Dimension in x
b = 10; % Dimension in y
c = 10; % Dimension in z

nreg_a = 10; % Regions in x
nreg_b = 1; % Regions in y
nreg_c = 1; % Regions in z

nel_a = 2; % Elements in x / region
nel_b = 2; % Elements in y / region
nel_c = 2; % Elements in z / region
```

Script 5: Box model of 10 domains.

If this is the first time using this isotropic material, the `Material` package is employed for generating the `"Material_9.dat"` database. For this purpose, the `Set_parameters.f90` script is shown in Script 6.

```
SUBROUTINE Setup
    fileplace="Data/Isotropic/"
    file_name="Material_9.dat"
    Material="Isotropic"
    E=200000.d0; nu=0.d0;
END SUBROUTINE Setup
```

Script 6: Isotropic material for transient tests.

Now the configuration of BESLE `Set_parameters.f90` script is illustrated in Script 7.

```
MODULE Set_parameters
!-------------------------------------------------------------!
USE Global_variables
!-------------------------------------------------------------!
CONTAINS
    SUBROUTINE Setup
        mesh_file = 'Transient'
        fileplace_mesh = 'Mesh/Box/'
        scale_size_1 = 1e-3

        material_coefficients_file = 'Material_9'
        fileplace_material = "Material/Data/Isotropic/"
        scale_prop_mat = 1e6

        transient = 1
        time_steps = 300
        dt = 1e-6
        density = 7850.d0

        box_face_4 = (/0.d0,0.d0,0.d0,0.d0,0.d0,0.d0/)
        box_face_2 = (/100.d0*(1e6),1.d0,0.d0,1.d0,0.d0,1.d0/)

        results_file = 'Results'
        fileplace_results = 'Results/'
    END SUBROUTINE Setup
END MODULE Set_parameters
```

Script 7: Transient behavior under a Heaviside boundary condition. Lines 7-9: Mesh file and scaling the solid dimensions; lines 11-13: Material database and scale the units of material properties; lines 15-18: transient regime definitions and mass density; line 20-21: fully constraint condition imposed at the Γ_4 surface and $t_1(\tau) = 100$ MPa, $t_2 = t_3 = 0$ at the Γ_2 surface (see pages 46 and 47 of the User's Guide); and lines 23-24: results definitions.

In this problem, the mass or inertial effects will be captured by the simulation as the time scale is very small. This model can be validated using the analytical solution shown in detail by Clough and Penzien [23]. The transient response of a prismatic bar under axial Heaviside load $\sigma(\tau) = \sigma_o$, is obtained using the mode-superposition analysis, then the displacement analytical response is

$$u(x,\tau) = \frac{8P_o}{\pi^2}\frac{L_x}{EA}\sum_{n=1}^{\infty}\left\{(-1)^{n+1}\left(\frac{1-\cos\omega_n\tau}{(2n-1)^2}\right)\sin\left[\frac{(2n-1)}{2}\frac{\pi x}{L_x}\right]\right\} \quad , \tag{21}$$

where P_o is the applied force, E is the Young's modulus, L and A are the length and cross-sectional area respectively, x is the position from the point to be analyzed, n is the number of vibration modes and ω_n is the natural frequency of the nth vibration mode expressed as

$$\omega_n = (2n-1)\frac{\pi}{2}\sqrt{\frac{EA}{mL_x^2}}, \quad \text{where } n = 1,2,\ldots \ , \tag{22}$$

being m the mass per unit length. Results of the displacement $u_1(\tau)$ at point c of the bar are presented and compared to the analytical curve in Fig. 14.

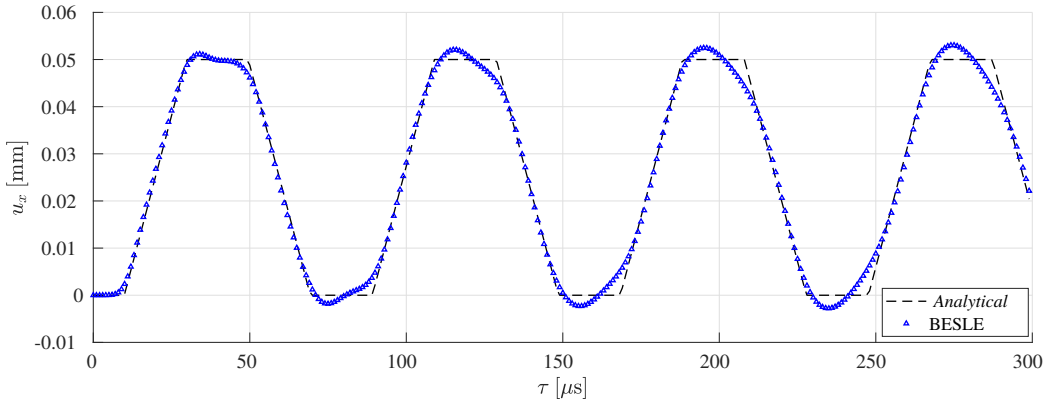


Figure 14: Displacement $u_x(\tau)$ at point $c(50,5,5)$ mm.

In this case, the instabilities occur at the points where the solution is non-differentiable, see Eq. (21). We observe that the discrepancy between the numerical and analytic solutions increases as τ increases, see Fig. 14 which was extracted directly from ParaView.

Now, a thin bar of the same material is analysed, Fig. (15), where an harmonic load $t_1 \equiv t_x(\tau) = 100\sin(\omega\tau)$ MPa is applied. In this special case, the resonance condition is simulated by setting $\omega = 0.99\omega_1$, where ω_1 is the first natural frequency of the system calculated with Eq. (22).

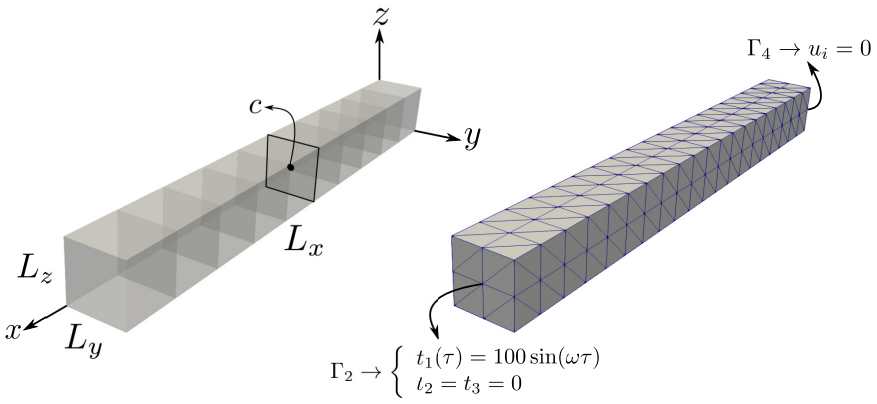


Figure 15: Box model of 10 domains discretised with 480 boundary elements (harmonic load).

In order to capture the transient regime and dynamic effects, high-rate boundary conditions are applied in a period of 0.2 ms. Hence, the simulation comprises 200 steps of 1 μs. The purpose of this simulation is to track the displacement $u_1 \equiv u_x(\tau)$ at the point $c(50,5,5)$ along the simulation time to compare it with the analytical model obtained using the superposition method as shown by Clough and Penzien [23] and given in [6] as follows

$$u(x,\tau) = \frac{2P_o}{mL_x}\sum_{n=1}^{\infty}\left\{\frac{(-1)^{n+1}}{\omega_n(\omega_n^2-\omega^2)}\left(\omega_n\sin\omega\tau - \omega\sin\omega_n\tau\right)\right. \\ \left.\sin\left[\frac{(2n-1)}{2}\frac{\pi x}{L_x}\right]\right\} \quad n = 1,2,\ldots \ . \tag{23}$$

The configuration of BESLE `Set_parameters.f90` script is illustrated in Script 8.

```
1  MODULE Set_parameters
2  !-------------------------------------------------------------!
3  USE Global_variables
4  !-------------------------------------------------------------!
5  CONTAINS
6
7      SUBROUTINE Setup
8
9          mesh_file = 'Transient'
10         fileplace_mesh = 'Mesh/Box/'
11         scale_size_1 = 1e-3
12
13         material_coefficients_file = 'Material_9'
14         fileplace_material = "Material/Data/Isotropic/"
15         scale_prop_mat = 1e6
16
17         transient = 1
18         time_steps = 200
19         dt = 1e-6
20         density = 7850.d0
21
22         load_profile='harmonic'
23         omega = 0.99*79286.645975178;
24         phase = 0
25
26         box_face_4 = (/0.d0,0.d0,0.d0,0.d0,0.d0,0.d0/)
27         box_face_2 = (/100.d0*(1e6),1.d0,0.d0,1.d0,0.d0,1.d0/)
28
29         results_file = 'Results'
30         fileplace_results = 'Results/'
31
32     END SUBROUTINE Setup
33
34 END MODULE Set_parameters
```

Script 8: Transient behavior under harmonic boundary conditions. Lines 8-10: Mesh file and scaling the solid dimensions; lines 12-14: Material database and scale the units of material properties; lines 16-19: transient regime definitions and mass density; line 21-23: harmonic load definitions ω and ϕ; lines 25-26: fully constraint condition imposed at the Γ_4 surface and $t_1(\tau) = 100$ MPa, $t_2 = t_3 = 0$ at the Γ_2 surface (see pages 46 and 47 of the User's Guide), in this case the load is $t_1 \equiv t_x(\tau) = 100\sin(\omega\tau)$ MPa; and lines 28-29: results definitions.

Finally, results of the displacement $u_x(\tau)$ at point c of the bar are presented and compared to the analytical curve in Fig. 16.

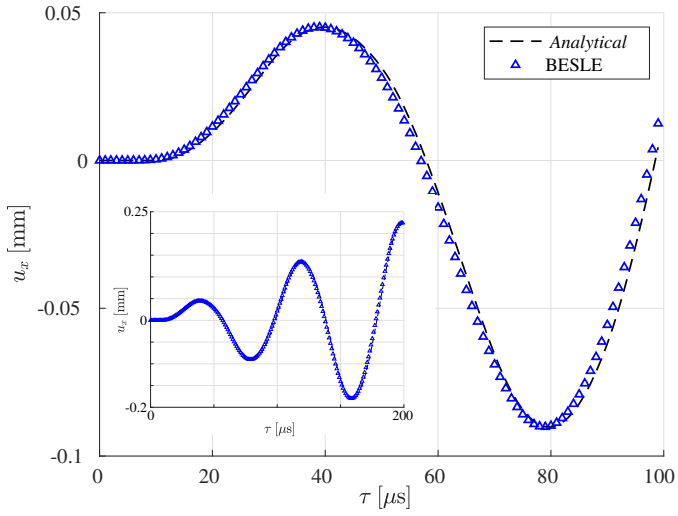


Figure 16: Displacement $u_x(\tau)$ at point $c(50,5,5)$ mm.

The numerical result is in excellent agreement with the analytical solution. It can be appreciated that this time, the analytical solution does not have sudden changes or non-differentiable points. Hence, the solution is stable for more time steps of simulation when compared to the solution for a Heaviside load. This result was extracted directly from ParaView.

It is worth noting that the Houbolt's algorithm is an implicit method where the solution for the next time step $\tau + \Delta\tau$ is calculated from the last three-time steps τ, $\tau - \Delta\tau$ and $\tau - 2\Delta\tau$. Then, as expected, reliable and accurate results for the transient regime are obtained when the time step, $\Delta\tau$, becomes small. However, there is a minimum size for $\Delta\tau$ under which the method may fail. The choice of time step for isotropic materials has been addressed by Gaul and Kogh [3] who provide a way to estimate an appropriate choice for the size of the time step using considering the velocity of the wave that propagates across the body. Unfortunately, their method is not readily generalisable to anisotropic materials. Hence, in BESLE the choice of time step must be determined via intuition, trial and error, or by taking advantage of analytical solutions or previous models in the literature [6, 15, 23].

5. Illustrative examples

Some physical models are presented here to illustrate the BESLE functionalities. Only, the description of each problem and their results are shown, since a detailed explanation of all variables and commands are given in the User's Guide.

5.1. Transient behavior of polycrystal aggregates

A 96-grain iron polycrystal aggregate with dimensions $L_x = 15\,\mu$m, $L_y = 15\,\mu$m and $L_z = 45\,\mu$m, Fig. 17 is considered. Each grain has its proper crystalline orientation from the reference of the three independent elastic constants $C_{11} = 230$ GPa, $C_{12} = 135$ GPa, and $C_{44} = 117$ GPa. Therefore, the stiffness tensor is rotated following the $x-y-z$ convention using the multiple rotations option of the `Material` sub-package. The specimen is fully constrained in all axes $u_i = 0$ at the bottom in the Γ_5 surface. Free surface $t_i = 0$ boundary conditions were imposed in the lateral surfaces. At the top in the Γ_6 surface, a Heaviside $t_3(\tau) = 200$ MPa traction is imposed.

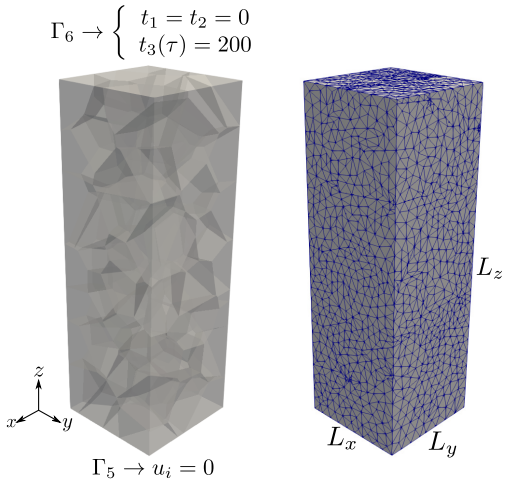


Figure 17: Model of a 96-grains polycrystalline material discretised with 22767 boundary elements.

In order to capture the transient regime and dynamic effects, high-rate boundary conditions are applied in a period of 0.25 μs. Hence, the simulation comprises 50 steps of 5 ns. Figures 18 and 19 present the displacement field at different instants of time.

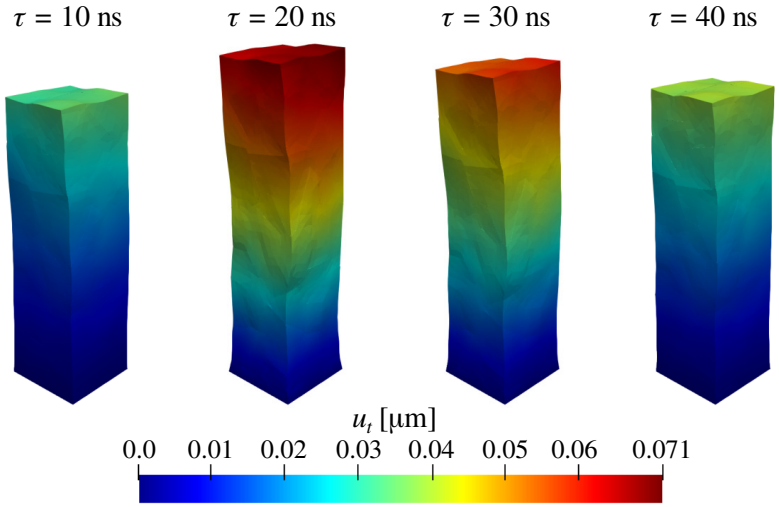


Figure 18: Total displacement u_t from 10 ns to 40 ns.

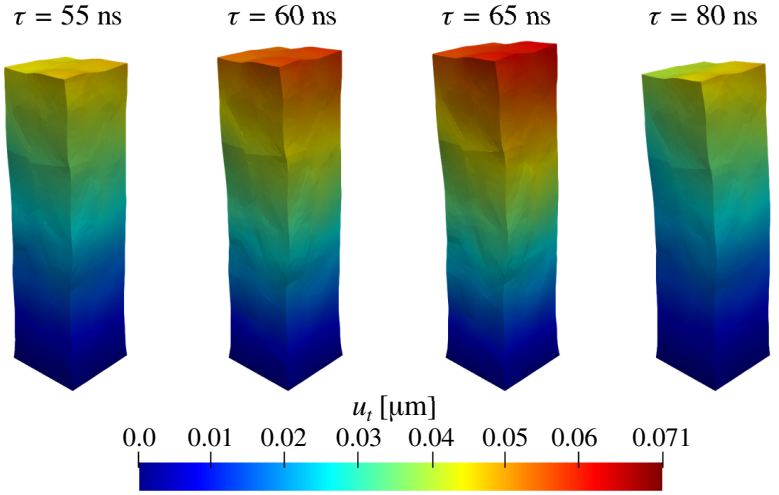


Figure 19: Total displacement u_t from 55 ns to 80 ns.

Results show how the stochastic anisotropy affects the deformation throughout time. Due to the high-rate boundary condition, the stress and strain waves propagate along with the specimen. As the load is a Heaviside function, these waves reflect in the body without damping. Furthermore, at some instants of time, only a small portion of the solid experiences deformation e.g. 10 ns and

40 ns. The consequences of this transient behaviour on the failure of polycrystals were explored in more detail in [9].

5.2. *Vertebra* L2 *modelling*

This example attempts to illustrate a realistic model of a vertebra L2 shown in Fig. 20. The mesh was modeled as previously described in subsection 3.1.1.

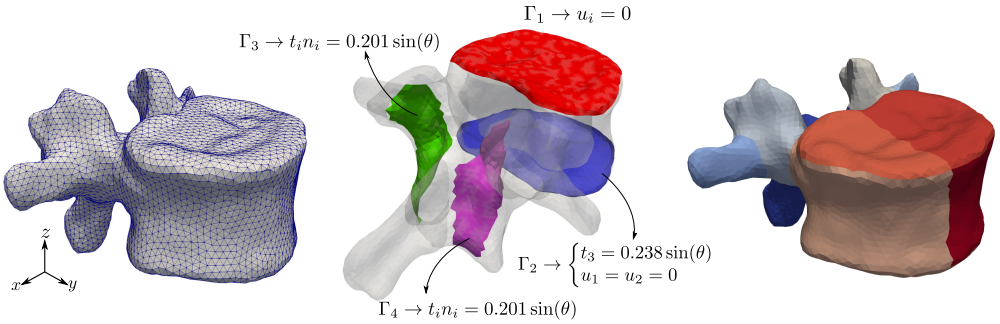


Figure 20: Vertebra L2 model of 15 domains discretised with 21612 boundary elements.

The boundary conditions to be applied to the vertebra L2 are set assuming that it supports upper body weight of 54 kg, and the selection of the boundary elements is based on the loading described by Jong et al. [44].

Figure 20 shows the full constraint at Γ_1, the vertebral body at Γ_2 and the articular processes at Γ_3 and Γ_4. The BCs were prepared for a quasi-static simulation of 4 steps where the variable θ varies $\pi/2$ at each step load. In this case, the boundary conditions will be imposed throughout the `General` mesh generator. There are two anisotropic materials involved here, the trabecular and cortical bones. Special care must be taken in the indication of each material domain. Since the trabecular bone accounts with 12 orientations located at specific locations of the vertebra L2. Finally, results are presented in Fig. 21.

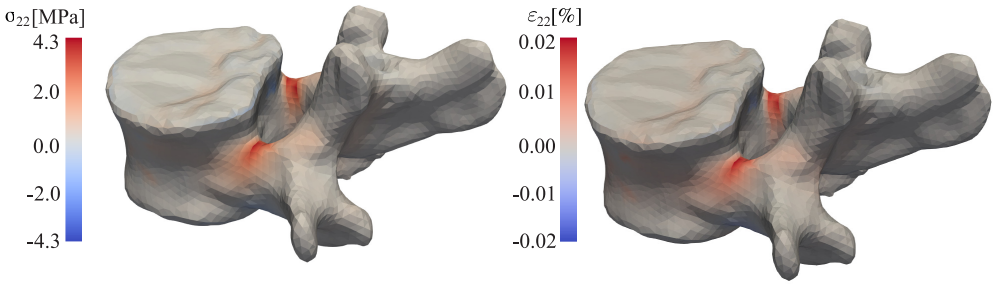


Figure 21: Vertebra L2 results at the four load step.

High levels of stress and strain were evidenced in the vertebral arch, specifically in the pedicle, and in the vertebral body. This is consistent with the results reported by Jong et al. [44]. Nevertheless, in the subdomain interfaces, a stress concentration due to stress shielding was observed. Then, it is recommended, when anisotropic tissues are being modelled, to subdivide them into small pieces to provide a smooth transition of their elastic properties.

6. Summary

A new software based on boundary element analysis was developed, which has a variety of functionalities that allow simulating practical problems of 3D linear elasticity of solids including: i) static, quasi-static, dynamic, and transient regimes, using ii) morphologies with a single or multiple domains with iii) isotropic and anisotropic materials and iv) Dirichlet or Neumann boundary conditions. Moreover, it comes complete with datasets for the elastic properties of a range of materials, allowing realistic simulations to be easily setup and executed. Implementation of BEM formulations in 3D are challenging due having to deal with singular integration and fully populated incidence matrices generated by the collocation point scheme. However, taking advantage on the multidomain algorithm, the general system of equations of homogeneous materials is enforced to be sparse, up to adequate levels, to avoid the degeneration of the mechanical behaviour. Since the software can be executed on distributed architecture memory, it is ideal to be used in workstations and clusters. This aids in the reduction of the processing time of the critical steps, such as the computation of the incidence matrices, and the solution of the system of equations using MUMPS.

We have demonstrated that simulations of highly stochastic materials, composed of random domain morphologies, can be carried out with BESLE and some of its functionalities. Furthermore, simulations involving many complex material constituents are feasible to be configured without requiring any special programming feature, which is an advantage over commercial or ready published packages for analysing linear elasticity. The correct configuration of boundary conditions is also a difficult task, especially in 3 dimensions. A diverse set of options for imposing loads is provided, however, the user has also the option to employ external software if the input style is compatible.

7. Declaration of competing interest

The authors wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgements

The authors would like to thank to the University of Campinas (Brazil), Brunel University London (UK), and the University of Portsmouth (UK) for the facilities and structure provided to develop this version of BESLE. The project was funded by the National Council for Scientific and Technological Development-CNPq (Grant Numbers: 312493/2013-4, 154283/2014-2 and 312536/2017-8), the Brazilian Coordination for the Improvement of Higher Education Personnel-CAPES (Grant Number: 435214/2019-01).

This material is based upon work supported by the Air Force Office of Scientific Research-AFOSR under Award Numbers FA9550-18-1-0113 and FA9550-20-1-0133.

Andres F. Galvis was supported by the EPSRC New Investigator Award "Multiscale modelling of mechanical deterioration in lithium-ion batteries" Grant number EP/T000775/1.

Luiz C. Wrobel also thanks the CNPq (Grant Number: 303770/2019-8) for his personal financial support.

Computational resources were provided by the Center for Computational Engineering and Science-CCES at the University of Campinas funded by the São Paulo Research Foundation-FAPESP (Grant Number: 2013/08293-7).

References

[1] J. H. Kane, Boundary Element Analysis In Engineering Continuum Mechanics, Prentice-Hall, New Yersey, 1994.

[2] M. H. Aliabadi, The Boundary Element Method Applications in Solids and Structures, Vol. 2, Wiley, London, 2002.

[3] L. Gaul, M. Kögl, M. Wagner, Boundary Element Method for Engineers and Scientists, Springer, Berlin, 2003.

[4] V. Gulizzi, A. Milazzo, I. Benedetti, Comput. Mech. 56 (2015) 631–651. doi:10.1007/s00466-015-1192-8.

[5] A. F. Galvis, R. Q. Rodriguez, P. Sollero, Mech. Mater. 117 (2018) 58–72. doi:10.1016/j.mechmat.2017.10.009.

[6] A. F. Galvis, R. Q. Rodriguez, P. Sollero, Comput. Struct. 200 (2018) 11–20. doi:10.1016/j.compstruc.2018.02.009.

[7] V. Gulizzi, C. H. Rycroft, I. Benedetti, Comput. Methods Appl. Mech. Eng. 329 (2018) 168–194. doi:10.1016/j.cma.2017.10.005.

[8] I. Benedetti, V. Gulizzi, A. Milazzo, Mech. Mater. 117 (2018) 137–151. doi:10.1016/j.mechmat.2017.11.001.

[9] A. F. Galvis, P. A. Santos-Flórez, P. Sollero, M. de Koning, L. C. Wrobel, Comput. Methods Appl. Mech. Eng. 362 (2020) 112868. doi:10.1016/j.cma.2020.112868.

[10] R. Q. Rodríguez, A. F. Galvis, P. Sollero, C. L. Tan, E. L. Albuquerque, Eur. J. Comput. Mech. 26 (5-6) (2017) 525–540. doi:10.1080/17797179.2017.1379863.

[11] R. Q. Rodriguez, L. S. Moura, A. F. Galvis, E. L. Albuquerque, C. L. Tan, P. Sollero, Eng. Anal. Boundary Elem. 104 (2019) 94–106. doi:10.1016/j.enganabound.2019.03.017.

[12] L. S. Moura, Numerical and experimental analysis of bonded composites, Master's thesis, University of Campinas (2019).

[13] D. M. Prada, A. F. Galvis, A. C. Alcântara, P. Sollero, Eur. J. Comput. Mech. 27 (5-6) (2017) 425–442. doi:10.1080/17797179.2018.1524054.

[14] D. M. Prada, Multiscale anisotropic elastostatic homogenization of healthy and osteoporotic bone tissue using 3D BEM, Master's thesis, University of Campinas (2019).

[15] A. F. Galvis, Multiscale Modeling of Dynamic Failure in 3D Polycrystalline Materials using BEM and MD, Ph.D. thesis, University of Campinas (2019).

[16] I. Benedetti, M. H. Aliabadi, G. Davì, Int. J. Solids Struct. 45 (2008) 2355–2376. doi:10.1016/j.ijsolstr.2007.11.018.

[17] R. Q. Rodríguez, C. L. Tan, P. Sollero, E. L. Albuquerque, CMES-Comp. Model. Eng. Sci. CMES 102 (5) (2014) 359–372. doi:10.3970/cmes.2014.102.359.

[18] A. M. Haider, M. Schanz, J. Theor. Comput. Acoust. 2019 (1) (2019) 1850060. doi:10.1142/S2591728518500603.

[19] Z. Yao, X. Zheng, H. Yuan, J. Feng, Eng. Comput. 36 (8) (2019) 2530–2556. doi:10.1108/EC-10-2018-0477.

[20] A. Ayala, X. Claeys, L. Grigori, J. Comput. Appl. Math. 368 (2020) 112528. doi:10.1016/j.cam.2019.112528.

[21] L. S. Moura, A. F. Galvis, E. L. Albuquerque, P. Sollero, Failure analysis of adhesively bonded composite joints using 3D BEM formulation, in: Advances in Boundary Element & Meshless Techniques, Vol. 20, pp. 128–134.

[22] R. T. Qu, Z. J. Zhang, P. Zhang, Z. Q. Liu, Z. F. Zhang, Sci. Rep. - Nature 6 (2016) 23359–8. doi:10.1038/srep23359.

[23] R. Clough, J. Penzien, Dynamic of Structures, Third Edition, Computers & Structures, Inc, Berkeley, 2003.

[24] C. L. Tan, Y. C. Shiah, C. Y. Wang, Int. J. Solids Struct. 50 (2013) 2701–2711. doi:10.1016/j.ijsolstr.2013.04.026.

[25] P. R. Amestoy, I. S. Duff, J. Koster, J.-Y. L'Excellent, SIAM J. Matrix Anal. Appl. 23 (1) (2001) 15–41. doi:10.1137/S0895479899358194.

[26] P. R. Amestoy, A. Guermouche, J.-Y. L'Excellent, S. Pralet, Parallel Comput. 32 (2) (2006) 136–156. doi:10.1016/j.parco.2005.07.004.

[27] Y. Liu, Fast Multipole Boundary Element Method Theory and Applications in Engineering, Cambridge University Press, 2009.

[28] R. Hiptmair, L. Kielhorn, BETL — A generic boundary element template library, techreport, Seminar for Applied Mathematics, ETH Zürich (2012).

[29] W. Śmigaj, T. Betcke, S. Arridge, J. Phillips, M. Schweiger, ACM Trans. Math. Software 41 (2) (2015) 6:1–40. doi:10.1145/2590830.

[30] J. Dölz, H. Harbrecht, S. Kurz, M. Multerer, S. Schöps, F. Wolf, SoftwareX 11 (2020) 100476. doi:10.1016/j.softx.2020.100476.

[31] C. A. Brebbia, J. C. F. Telles, L. C. Wrobel., Boundary Element Techniques. Theory and Applications in Engineering, Springer Verlag, 1984.

[32] P. W. Partridge, C. A. Brebbia, L. C. Wrobel, The Dual Reciprocity Boundary Element Method, Springer Verlag, 1991.

[33] T. C. T. Ting, V. G. Lee, Q. J. Mech. Appl. Math. 50 (1997) 407–426. doi:10.1093/qjmam/50.3.407.

[34] P. W. Partridge, C. A. Brebbia, L. C. Wrobel, The Dual Reciprocity Boundary Element Method, Southampton: Computational Mechanics Publications, 1991.

[35] M. Kögl, L. Gaul, CMES-Comp. Model. Eng. Sci. CMES 1 (4) (2000) 27–43. doi:10.3970/cmes.2000.001.479.

[36] R. Rodríguez, A. F. Galvis, P. Sollero, C. L. Tan, E. L. Albuquerque, Acta Mech. 229 (2018) 1893–1910. doi:10.1007/s00707-018-2108-4.

[37] C. A. Brebbia, J. Dominguez, Boundary Elements An Introductory Course, McGraw-Hill, 1989.

[38] J. Dominguez, Boundary Elements in Dynamics, Computational Mechanics Publications, Southampton, 1993.

[39] E. L. Albuquerque, P. Sollero, M. H. Aliabadi, Int. J. Solids Struct. 39 (2002) 1405–1422. doi:10.1016/S0020-7683(01)00173-1.

[40] J. T. Katsikadelis, Boundary Elements: Theory and Applications, Elsevier, Oxford, 2002.

[41] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J.-C. Fillion-Robin, S. Pujol, C. Bauer, D. Jennings, F. Fennessy, M. Sonka, J. Buatti, S. Aylward, J. V. Miller, S. Pieper, R. Kikinis, Magn. Reson. Imaging 30 (9) (2012) 1323–41. doi:10.1016/j.mri.2012.05.001.

[42] I. Benedetti, M. H. Aliabadi, Comput. Mater. Sci 67 (2013) 249–260. doi:10.1016/j.commatsci.2012.08.006.

[43] S. G. Lekhnitskii, Theory of Elasticity of an Anisotropic Body, MIR Publishers, 1981.

[44] J. K. Shin, T. S. Goh, M.-S. Kim, K. Kim, M. J. Shin, S. M. Son, H. J. Lee, J. S. Lee, C.-S. Lee, Biomed. Res. 29 (2018) 464–471. doi:10.4066/biomedicalresearch.29-17-763.