GABES

Graphical & Accelerated Boundary Element Solver

User's Guide

SI HADJ MOHAND Rahim
BELKACEMI Yacine
RECHAK Said

NATIONAL POLYTECHNIC SCHOOL | ALGIERS, ALGERIA | JANUARY 2026

1 INTRODUCTION

GABES (**G**raphical & **A**ccelerated **B**oundary **E**lement **S**olver) is an open-source application based on the Boundary Element Method (BEM), accelerated using an $\mathscr{H}$-Matrix library, for solving three-dimensional linear elastostatic problems in isotropic, homogeneous media. The application runs exclusively on 64bit Windows platforms and provides a user-friendly graphical interface with an interactive 3D environment. Users can import meshed 3D models, define numerical and physical parameters, apply boundary conditions interactively, and launch simulations. GABES also includes post-processing tools for graphical visualization of results and the generation of tabulated reports.

2 Download and setup

GABES is available as an open-source package on the GitHub platform, with the repository accessible at https://github.com/rahimsihadjmohand-png/GABES.git. The repository contains the latest committed source code provided as a Visual Studio solution/project, as well as several binary releases. Detailed instructions for setting up the source code and installing the binary releases are presented in the following sections.

2.1 Download the latest GABES release

The GABES project is currently in active development. To facilitate immediate testing and use, each incremental update is published as a direct release. This allows users to download and execute the application without needing to compile the source code themselves.

At present, two beta pre-releases are available:

- **GABES v0.9.0-beta**
- **GABES v0.9.1-beta**

These versions are functional and ready for testing. However, some planned features are not yet fully implemented, as development is still ongoing.

How to Download a Release

1. Navigate to the **Releases** section on the GABES main repository page (see Figure 1).
2. All available releases will be displayed (see Figure 2).
3. Click on the latest release (**Version 0.9.1-beta**) to open its dedicated page.
4. Download the release package by selecting the first **.zip** file, **GABES v0.9.1-beta.zip** (see Figure 3). This package contains the **GABES.exe** executable along with all required dependency (**.dll**) files necessary for proper execution.

The release page also includes the complete source code corresponding to that specific release version. However, keep in mind that source code changes are sometimes committed by the author without publishing an entirely new release. To ensure you obtain the latest source code, it is recommended to download it directly from the repository's main page, as will be demonstrated in the following sections.

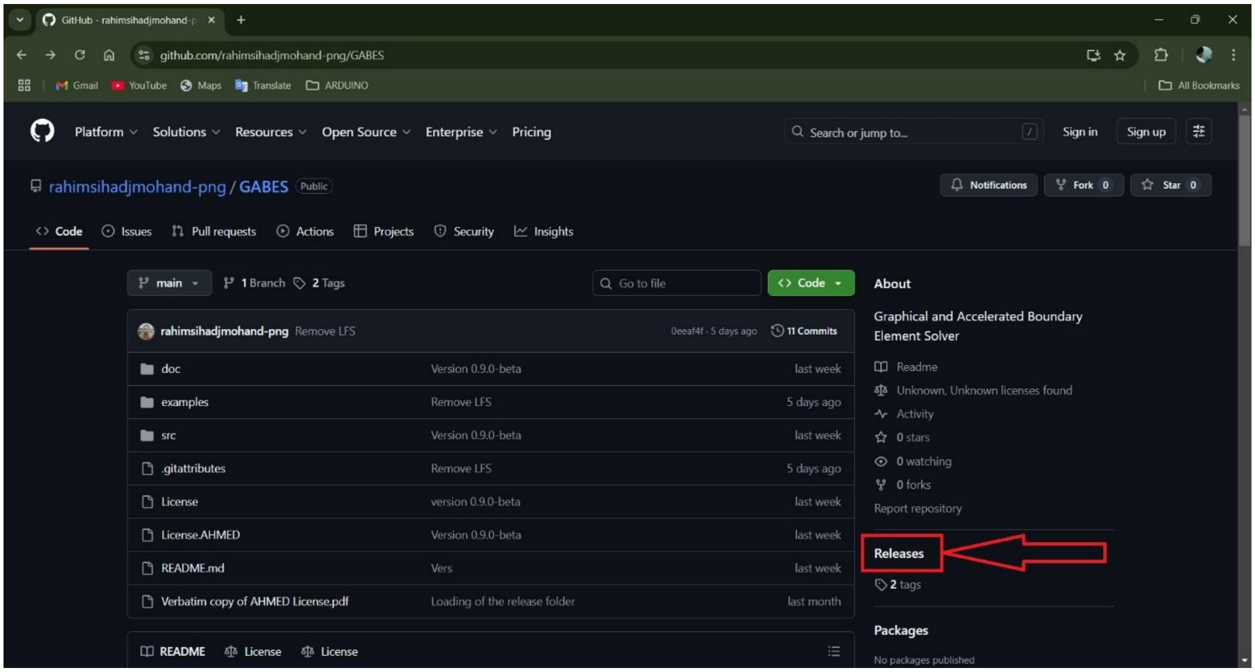


Figure 1: Accessing the releases

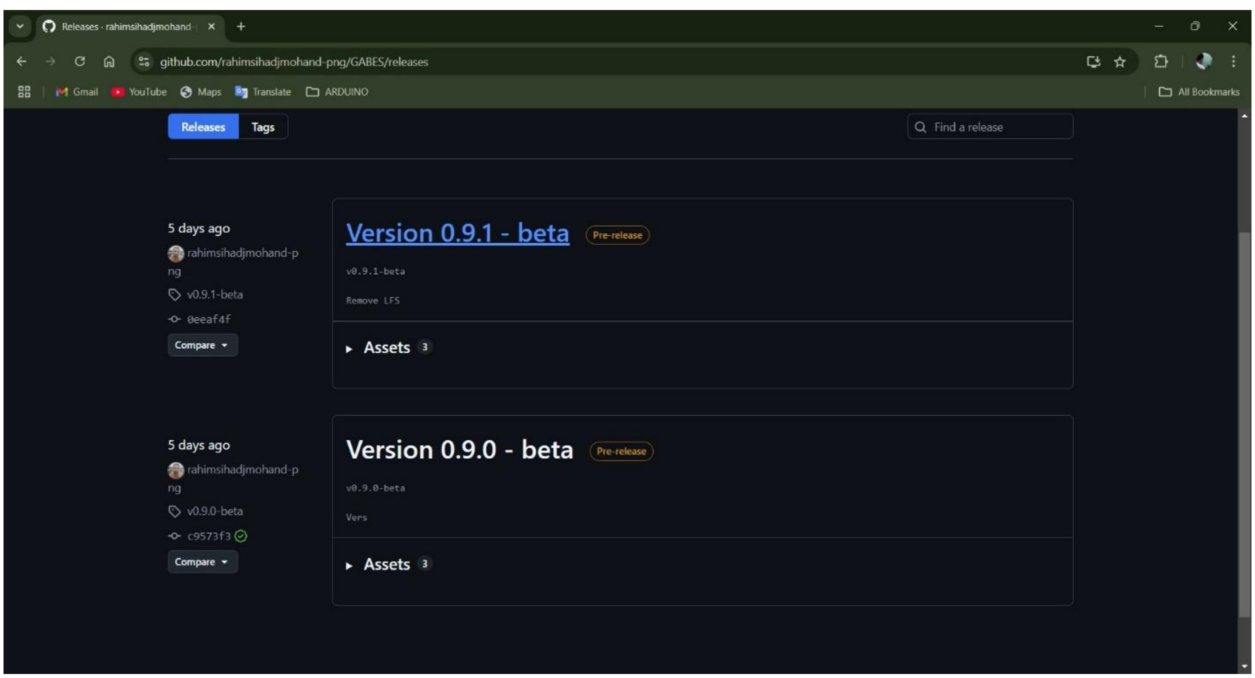


Figure 2: The available releases

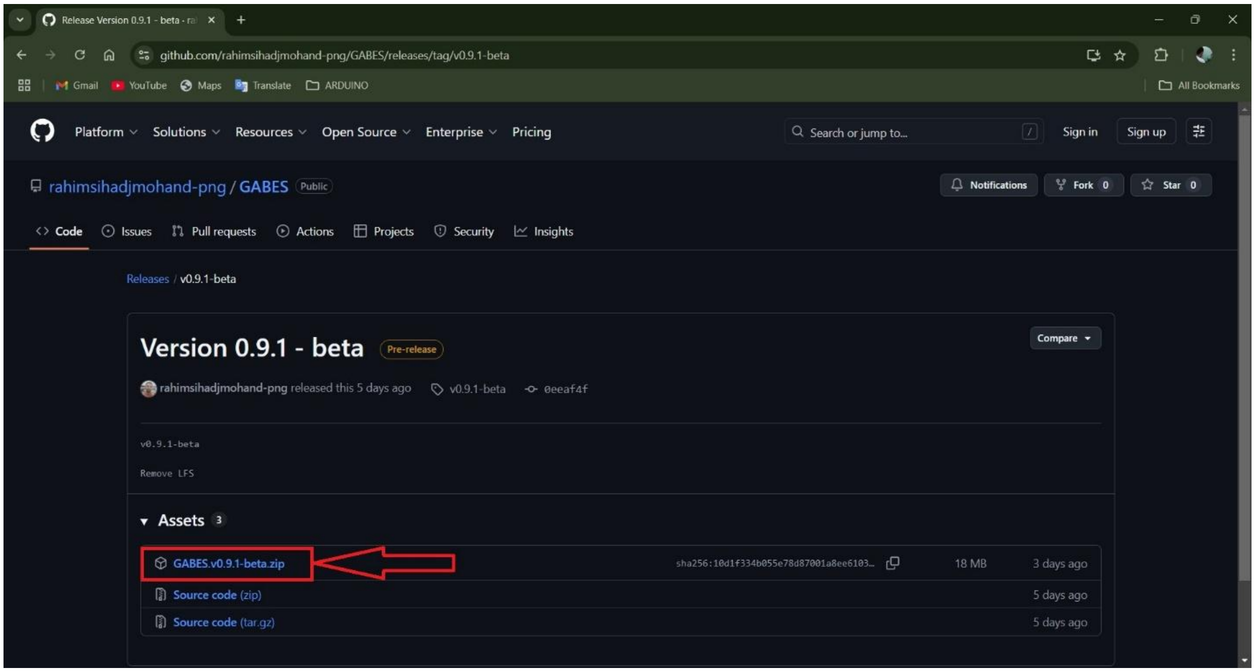


Figure 3: Latest release download

2.2 Download the source code

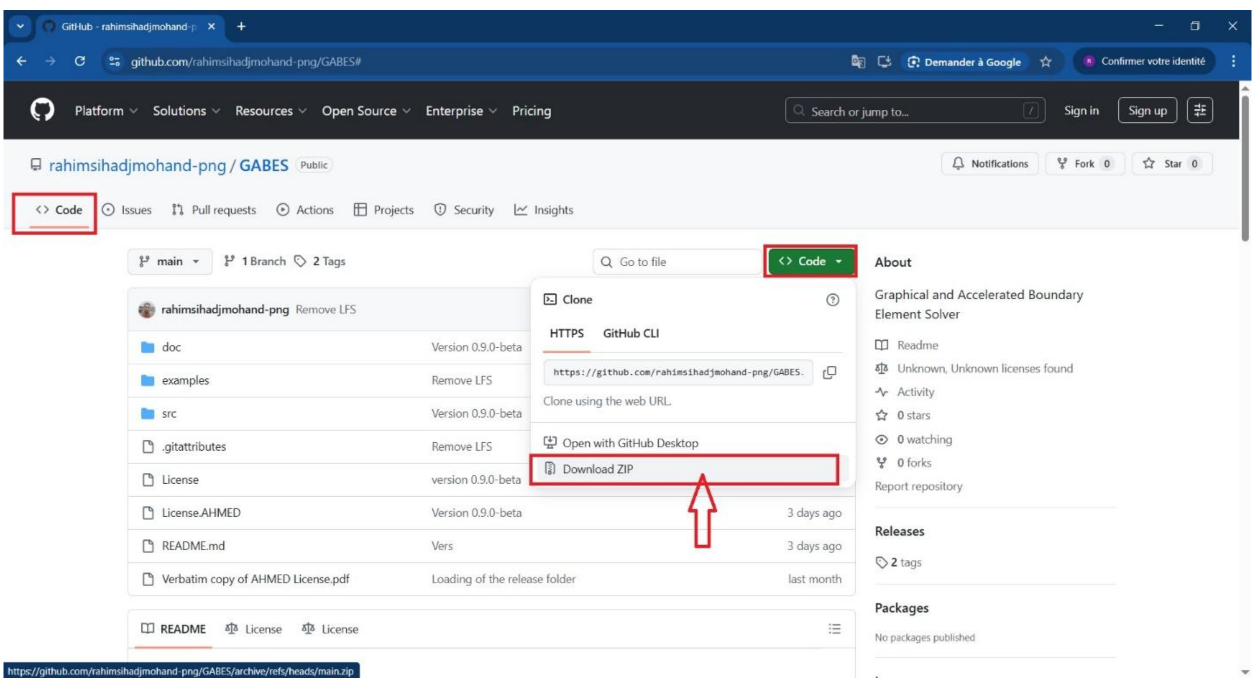


Figure 4: Download the source code

To download the full source code package, navigate to the Code tab. Click the green Code button; a context menu will appear. Select Download ZIP to start the download automatically. The ZIP archive will be saved to the browser's default download directory, as shown in Figure 5. Alternatively, users with Git installed on Windows can download the repository via the command line, by opening a command prompt or terminal in the desired directory and execute the command:

```
git clone https://github.com/rahimsihadjmohand-png/GABES.git
```

However, for most Windows users, the first method is sufficient and more practical.

Upon completion of the download and extraction of the archive, the directory structure is organized as illustrated in Figure 6 and consists of the following contents:

1. The **doc** directory contains the present user's manual.
2. The **examples** directory contains two example meshes along with the original 3D models used to generate them. The first example is a beam model created in **Autodesk 3ds Max** and exported as an **.OBJ** file, together with its original **.3DS** file. The second example is a connecting lug modeled in **SolidWorks**, exported as a **.STEP** file, and subsequently meshed and refined into an **.STL** file using **Gmsh.**
3. The **src** directory contains the complete source code of GABES provided as a Microsoft Visual Studio solution.
4. The **.gitattributes** file contains GitHub configuration instructions.
5. The **License** file specifies the main license terms governing GABES.
6. The **License.AHMED** file contains the license terms for the AHMED $\mathcal{H}$-Matrix library.
7. The **README.md** file provides brief setup and usage instructions, summarized with less detail than in the present user's manual.
8. The **Verbatim copy of AHMED License.pdf** file is a verbatim copy of the AHMED $\mathcal{H}$-Matrix library license and contains the same information as the **License.AHMED** file.

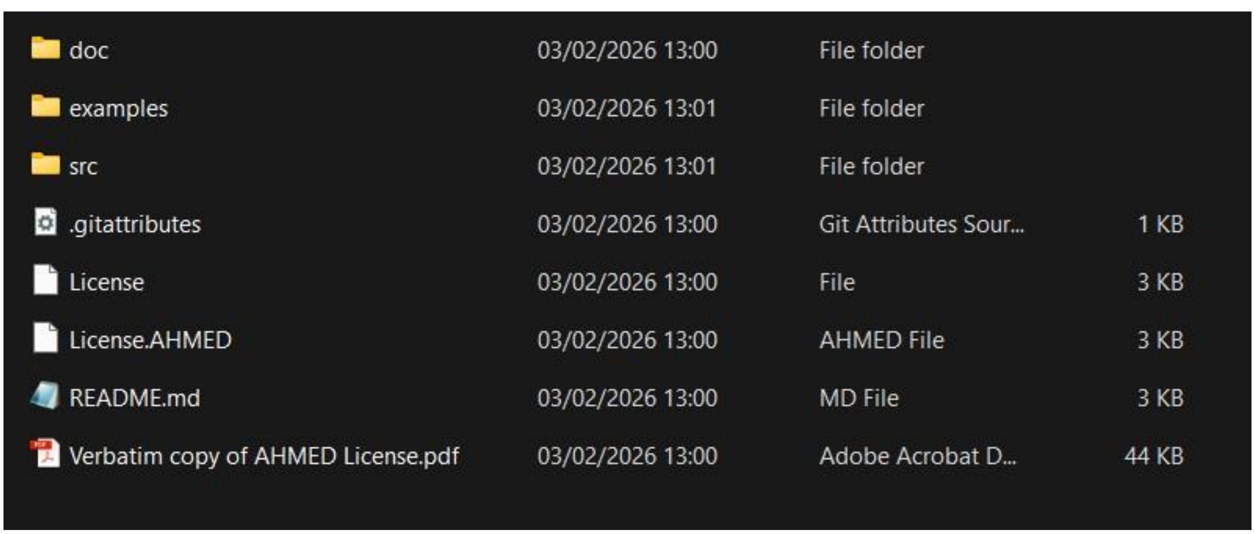


Figure 5: Downloaded source package

2.3 Source code modification and compilation

To successfully modify and compile the source code, the installation of the **Microsoft Visual Studio IDE** with C++ and MFC support is required. The source code is provided as a Visual Studio solution/project package that relies on several external dependencies, which must be pre-installed and properly configured within the project properties.

To open the (solution/project) package in visual studio, click on the **GABES.sln** file inside the **src** folder which opens the entire code in Visual studio as shown in Figure 6.

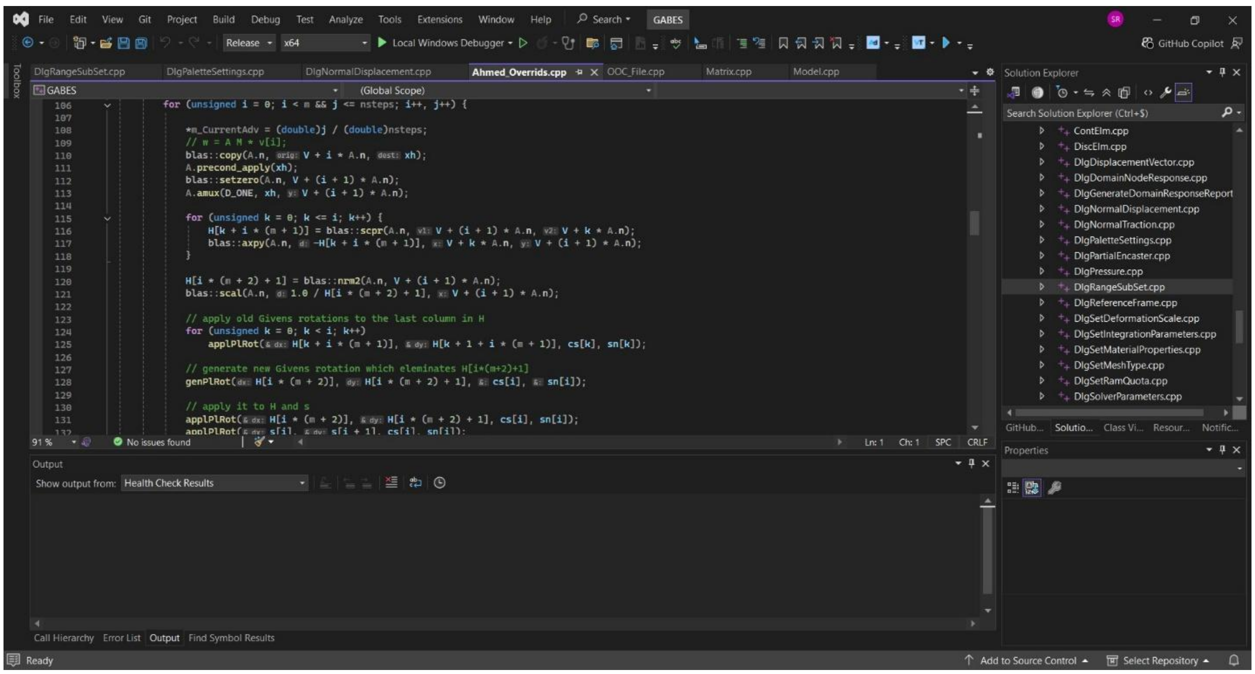


Figure 6: The GABES Visual Studio solution/project

The configuration process is end-user dependent due to possible variations in third-party library installation paths. This configuration enables Visual Studio to correctly locate the binary dependencies as well as the directories containing the required library header files.

2.3.1 Third-party dependencies

GABES integrates several third-party libraries that support different components of the application, including the graphical user interface, the 3D visualization environment, and the $\mathcal{H}$-matrix framework. Some of these libraries are natively available in standard Windows installations, while others are automatically provided or can be integrated directly through Visual Studio. In contrast, certain libraries must be installed separately, requiring manual configuration of the **include** and **library** paths by the user. All libraries used in GABES are listed in Table 1.

Table 1: GABES third-party dependencies

Library	Description	Configuration
MFC	**M**icrosoft **F**oundation **C**lasses (MFC): a C++ wrapper library for Windows graphical user interface (GUI) and desktop application development.	The library is an integral part of the Microsoft Visual Studio toolchain and is automatically configured. In older Visual Studio distributions, it is installed by default. However, in recent Visual Studio versions, MFC installation must be explicitly selected during the Visual Studio setup process; otherwise, it can be added later using the Visual Studio Installer.
DirectX 9.0	a Microsoft-specific 3D multimedia library used for 3D programming in multimedia applications, particularly games.	On older Windows distributions, DirectX must be manually installed by the user. On recent Windows versions, the core DirectX runtime components are included with the operating system; however, certain legacy DirectX components (such as D3DX libraries) may still require separate installation.
AHMED	The AHMED library: provides the $\mathscr{H}$-Matrix framework, including routines for block partitioning, construction of $\mathscr{H}$-Matrix approximations, and various $\mathscr{H}$-Matrix linear algebra and solver operations.	The library must be installed, and both its header files and compiled binaries must be properly configured for the source code.
OpenMP	OpenMP is a parallel programming framework used for thread-level parallelism on multi-core, shared-memory machines. It allows programs to execute multiple threads concurrently within a single process. OpenMP does **not** provide parallelism across a network of separate machines (cluster-level parallelism).	OpenMP is available in the Visual Studio toolchain. However, its support must be enabled in the Visual Studio project settings.
BLAS	Basic Linear Algebra Subprograms (BLAS) is a standard library providing highly optimized routines for vector and matrix operations, including dot products, vector scaling, and matrix multiplication. It is widely used as a foundational library in scientific computing and high-performance numerical applications.	The BLAS library must be installed separately, and the project must be configured to include its header files and link to the compiled binaries

2.3.1.1 DirectX 9.0

GABES relies on the DirectX 9.0 Software Development Kit (June 2010), which provides all the development tools required for 3D multimedia applications. This includes the core runtime, helper libraries such as D3DX, complete header files for function prototypes, and tools for tasks such as texture management. The SDK also contains numerous tutorial examples to facilitate learning and development. The DirectX SDK can be obtained from Microsoft's official website: https://www.microsoft.com/en-us/download/details.aspx?id=6812. Installation is straightforward and guided through a graphical Windows installer, which is familiar to most users.

2.3.1.2 AHMED

GABES uses the AHMED library primarily to transform dense linear system matrices into $\mathscr{H}$-Matrices and to provide specialized $\mathscr{H}$-Matrix solver methods. These solvers have reduced algorithmic complexity, making relatively large-scale simulations feasible on a personal computer. Unlike classical direct solvers, which have cubic asymptotic complexity, the $\mathscr{H}$-Matrix technique can achieve near-logarithmic complexity, resulting in a significant computational gain.

AHMED is an open-source library distributed under an academic-only license and is accessible at the following GitHub repository: https://github.com/xantares/ahmed.git. It is provided as a cross-platform CMake C++ project. To set it up on Windows, the AHMED CMake project can be configured within Visual Studio with Windows-specific parameters to generate a Visual Studio solution package. Compiling this solution produces the final **.dll** and **.lib** files. The **.lib** files are specifically required in Microsoft Visual C++ to correctly link and access the corresponding dynamic-link library **(.dll)**, which contains the main library code. For Windows, the provided AHMED **CMakeLists.txt** file should be carefully reviewed and adapted to ensure successful compilation. Once the Visual Studio AHMED project is built successfully, it will generate the required files (**AHMED.dll** and **AHMED.lib**), which are then used in the GABES project.

2.3.1.3 BLAS

BLAS (Basic Linear Algebra Subprograms) is a highly optimized library used for fundamental vector and matrix operations, including dot products, vector scaling, and matrix multiplication. It is widely used in scientific computing and high-performance numerical applications to accelerate core numerical routines. Common BLAS implementations include **OpenBLAS** and **Intel Math Kernel Library (MKL)**. OpenBLAS is an open-source, high-performance implementation available at https://github.com/OpenMathLib/OpenBLAS, and Intel MKL is a commercially supported, highly optimized implementation available as part of **Intel oneAPI**.

GABES uses the Intel MKL implementation of BLAS in the AHMED library's overridden functions. But The user is free to use another implementation of BLAS.

2.3.2 Visual Studio project configuration

2.3.2.1 Configuration of the library paths

The Microsoft Visual C++ compiler (MSVC) requires explicit specification of the locations of all library header and binary files. These paths are configured through the project's **Properties** dialog, which can be accessed from the Visual Studio main menu by selecting **Debug**, then **GABES Debug Properties**.

Within the **Configuration Properties** tree, the user must navigate to **VC++ Directories**. For each external library, both the *Include* and *Library* directories must be specified by selecting **Edit** from the corresponding dropdown combo-box. Consequently, the user must have prior knowledge of the locations of the library header files, which are added to **Include Directories**, as well as the binary (.lib) files, which are added to **Library Directories**.

In most cases, header files are located in an **\include** directory, or occasionally in a **\src** directory for projects that do not separate source (.c, .cpp) files from header (.h) files. Binary library files are generally located in a **\lib** directory; however, care must be taken to select the correct platform-specific subdirectory. Typically, libraries provide separate folders for 32-bit (x86) and 64-bit (x64) builds. It is important to note that **GABES targets exclusively 64-bit platforms**, and no configuration is provided for 32-bit builds.

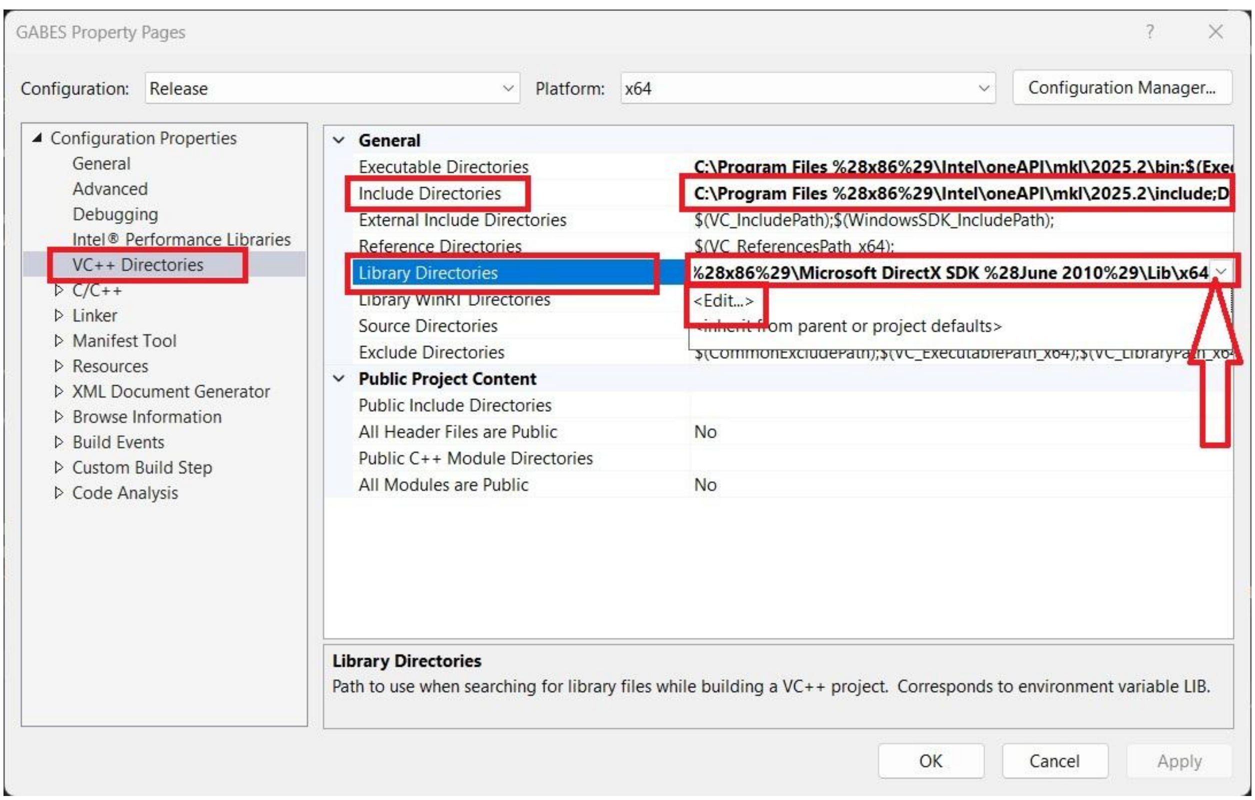


Figure 7: Library paths configuration

2.3.2.2 *Specifying linker input library files*

Once the library paths have been configured, the next step is to explicitly tell the MSVC linker which library **(.lib)** files it should use when building the final executable **(.exe)**. This configuration is performed within the same **Project Properties** dialog.

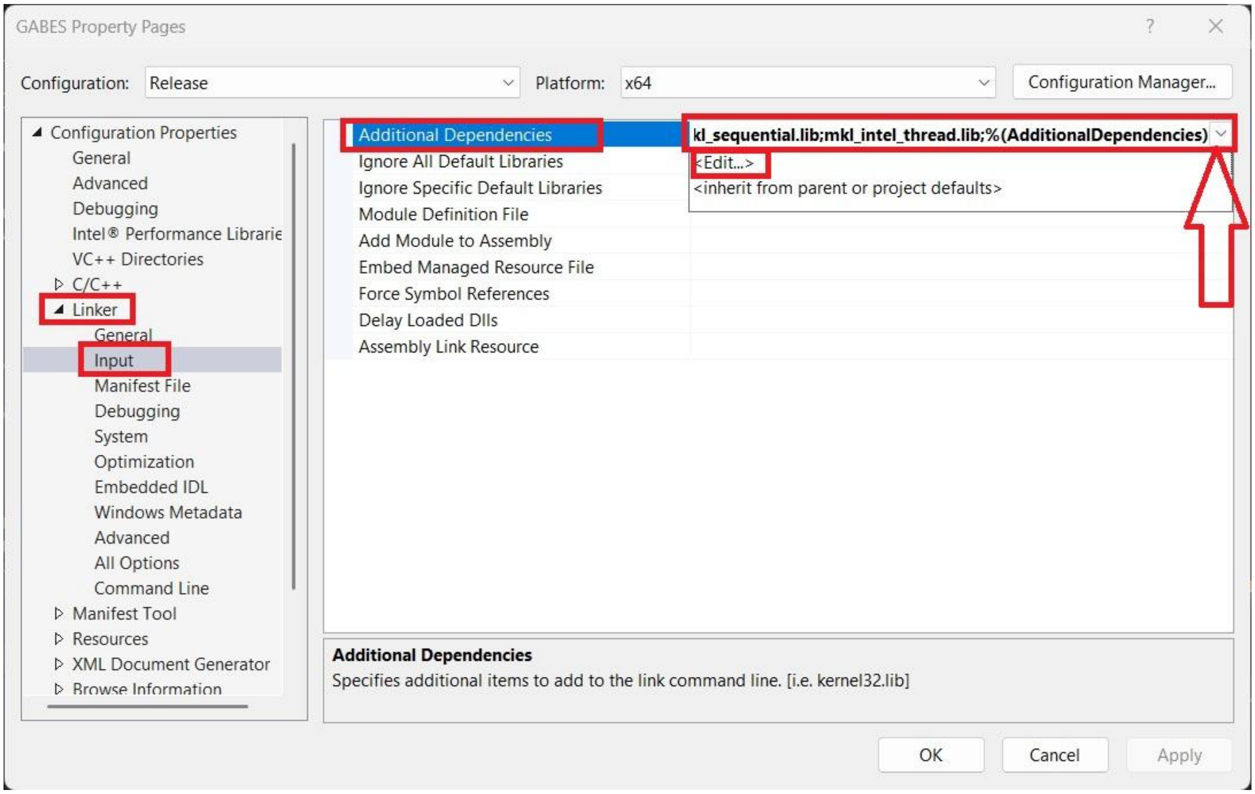


Figure 8: Specifying linker input library files

As shown in Figure 8, navigate to: **Linker** > **Input**. In the **Additional Dependencies** field, you must add the required library files. This can be done by clicking the corresponding dropdown combo-box and entering the library names. In the dialog that appears, add the library files listed in Table 2.

Table 2: Required linker input library files

File	Purpose / Provider
d3d9.lib	Direct3D 9 Core Library
d3dx9.lib	Direct3D 9 Extensions Library (D3DX)
ahmed.lib	AHMED (.lib) file
mkl_core.lib	Intel Math Kernel Library (MKL) Core
mkl_intel_lp64.lib	MKL LP64 Interface (for 64-bit integers)
mkl_sequential.lib	MKL Sequential Layer
mkl_intel_thread.lib	MKL Intel Threading Layer

N.B. If the application's BLAS functionality is supplied by a library other than Intel MKL, the required linker files will differ accordingly. For instance, when using OpenBLAS, you would substitute all MKL libraries with **libopenblas.lib**.

2.3.2.3 Activating OpenMP

OpenMP is used for thread-level parallelism (machine-local parallelism) and is primarily employed in GABES to parallelize loops involved in the computation of coefficient matrices. OpenMP is already integrated into Visual Studio, but it must be explicitly enabled in the project properties.

To activate OpenMP, go to the Visual Studio main menu, select **Debug**, then choose **GABES Debug Properties**. In the project properties window, expand the **C/C++** tree, select **Language**, and set the **OpenMP Support** field to **Yes (/openmp)**, as illustrated in Figure 9. Users should ensure that the correct build configuration (Debug or Release) is selected, since each configuration can have different compilation options.

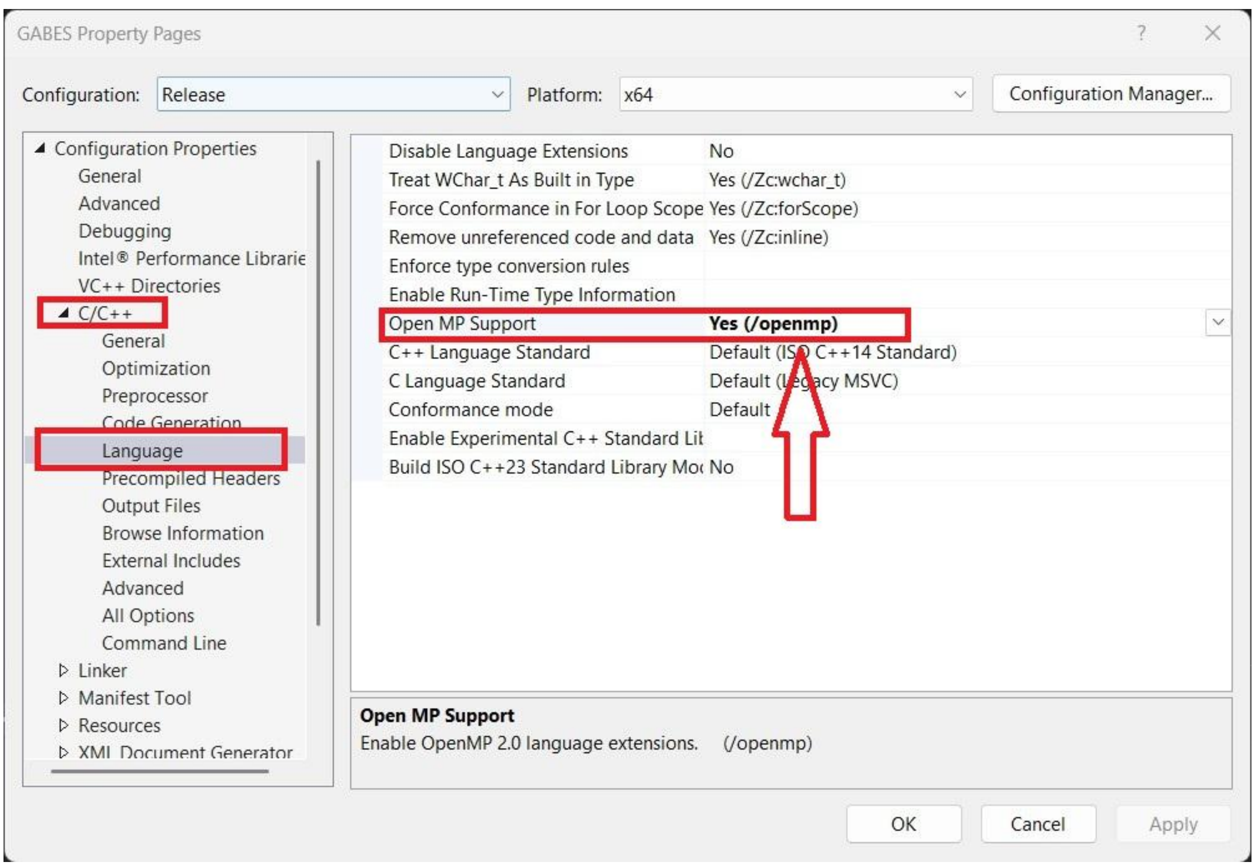


Figure 9: OpenMP Enabling

3 PRESENTATION OF THE APPLICATION

3.1 The Mainframe Window

The **Mainframe** window serves as the container for all graphical components of the application, as illustrated in Figure 10. It includes:

- **Client Area:** the primary workspace for 3D visualization and interaction;
- **Ribbon**: the main toolbar for commands and tools;
- **View Toolbar**: a toolbar for commands and tools to manipulate the 3D model;
- **Model Tree**: a hierarchical display of the various objects used during the modeling phase;
- **Information Panel**: displays context-specific data and parameters;
- **Status Bar**: shows the coordinates and the indices of the selected elements and nodes.

Each of these components will be described in more detail in the following sections.

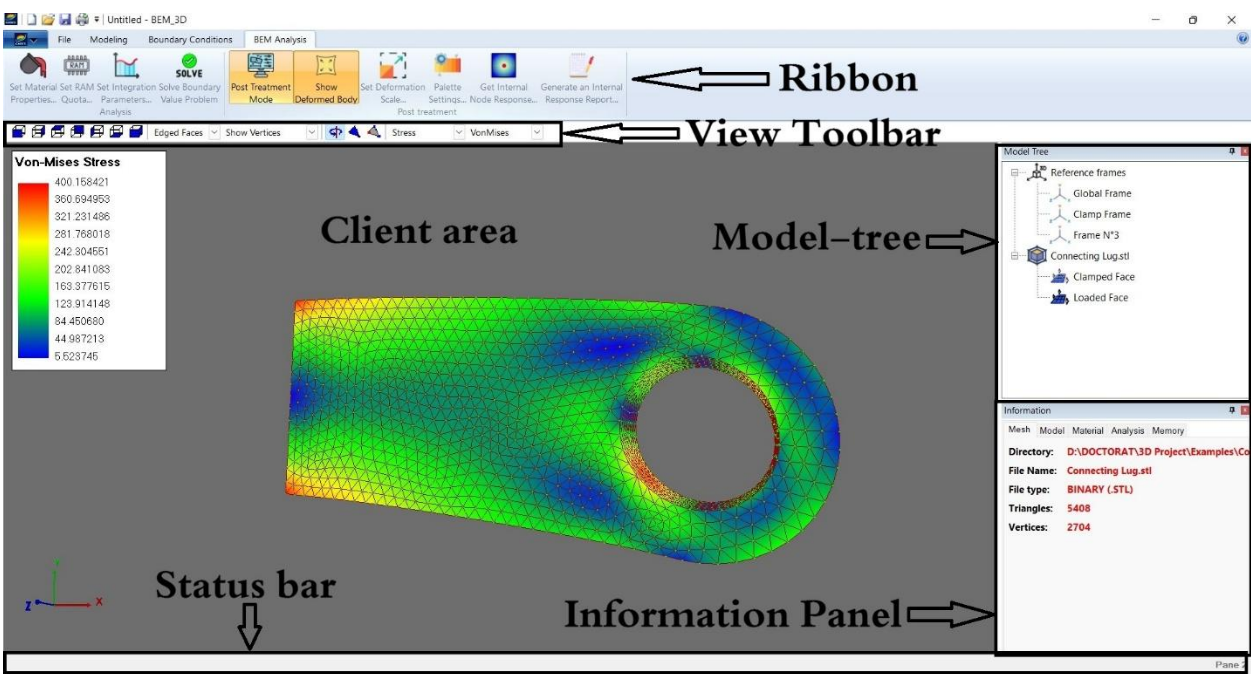


Figure 10: Mainframe Window

3.2 The Client Area

The **Client Area** is the primary space for visualizing the 3D model in both the modelling and post-processing phases. It permits rotation and zooming of the 3D model and contains a small representation of the three coordinate axes (Oxyz), known as the **Gizmo**.

In the **modelling phase**, the Client Area allows interactive selection of triangular mesh elements, vertices, and DOF nodes, which can be useful for various purposes. It also displays several objects (see Figure 2):

- The imported model mesh
- The created coordinate frames
- The created surface subsets
- The applied boundary conditions (simple/double fixations, displacement vectors, and traction vectors)

In the **post-processing phase** (see Figure 3), it displays:

- Various color-mapped field variables (displacements, stresses, and strains)
- A color palette with the corresponding values
- The deformed and undeformed configuration of the body

The model can be visualized in different output modes, including **solid surface**, **edged surface**, and **wireframe**. The visibility of geometrical mesh vertices and DOF nodes can be toggled on or off. All these view options can be controlled and configured using the **View Toolbar**.

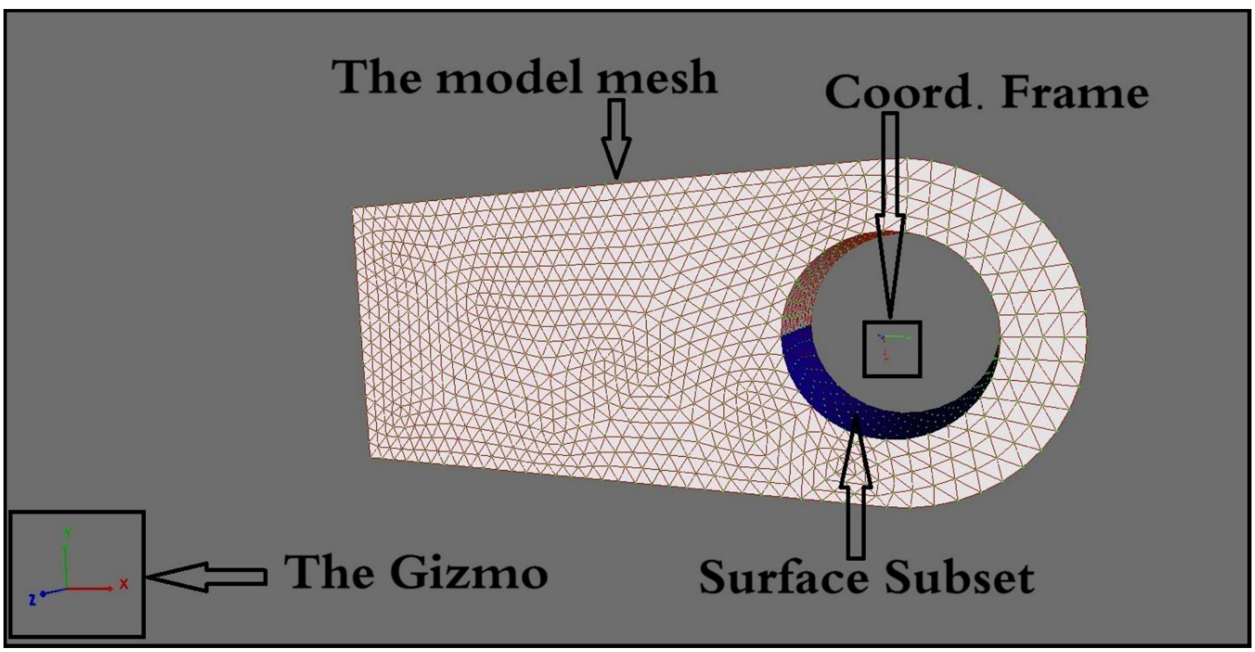


Figure 11: Client Area modeling phase

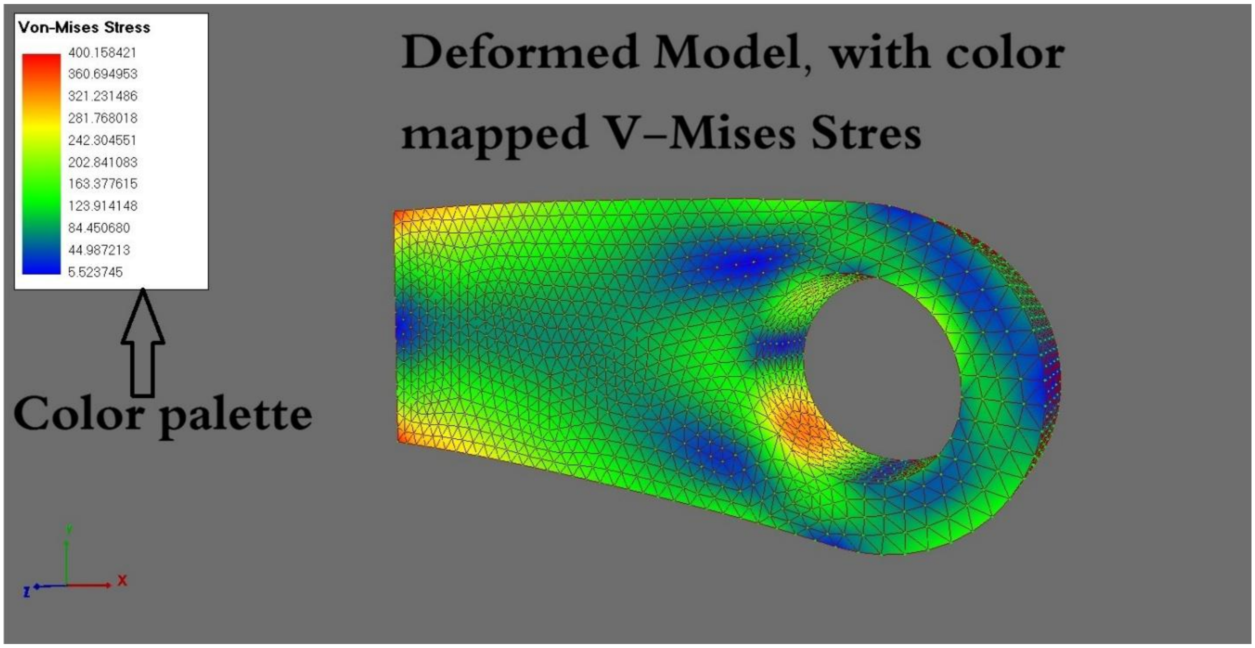


Figure 12: Client Area post-processing phase

3.3 The Ribbon

Modern GUI applications feature a **Ribbon**, which replaces the traditional menu and toolbar system. Commands are provided through buttons with larger icons, organized into panels across multiple separate tabs, making them more flexible and easily accessible.

The majority of **GABES's** commands are embedded in the **Ribbon**, which is organized into four (04) tabs:

3.3.1 File tab

Subdivided into two (02) panels:

- **Geometry panel:** Contains two (02) commands for loading and deleting the model mesh
- **GABES File Management panel:** Contains four (04) commands for creating, opening and saving GABES files.

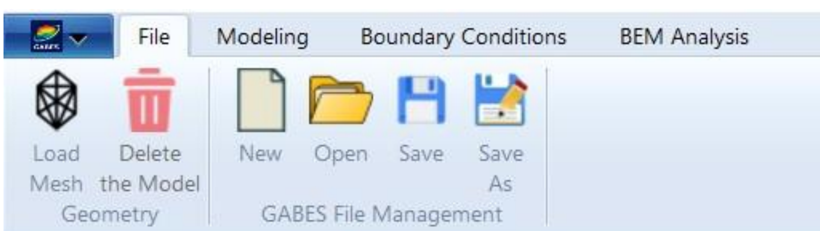


Figure 13: File tab

The **File tab** commands are summarized in Table 1

Table 3: Command buttons of the File tab

Panel	Button	Action
Geometry	Load Mesh	Opens a standard Windows **Open File** dialog to navigate to the location of the mesh file, which can be in one of two formats: **.stl** or **.obj**. The 3D surface mesh is then loaded into **GABES**
	Delete the Model	Deletes the current model and cleans up all generated data
GABES File Management	New	Creates a new **GABES** file and may prompt the user to save the current model if modifications have occurred
	Open	Opens an existing **GABES** file, which may contain all modeling information, including the mesh, reference frames, surface subsets, boundary conditions, and field variable results.
	Save	Saves the current model to an existing **GABES** file, or prompts the user to choose a save location if the file has not yet been created
	Save As	Saves the current model to a new disk location by prompting the user to choose the destination directory.

3.3.2 Modeling tab

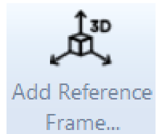
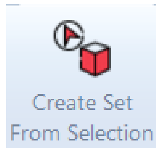
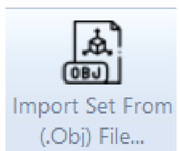
The tab is subdivided into two (02) panels:

- **Frames panel:** Contains a command for creating new reference frames, which are very helpful during surface subset creation and post-processing report generation.
- **Surface Subsets panel:** Contains three (03) different commands for generating surface subsets, which are mandatory for boundary condition assignment.

Figure 14: Modeling tab

The **Modeling tab** commands are summarized in Table 2

Table 4: Command buttons of the Modeling tab

Panel	Button	Action
Frames		Opens a modeless dialog box that allows the graphical, interactive creation of a new frame by defining the location of the origin point and the orientation of the axes.
Surface Subsets		Creates a new surface subset from the currently selected elements. This button becomes active only when a valid selection exists **N.B.** *Elements can be selected interactively in the 3D model using the mouse cursor, with the* ***CTRL*** *key enabling multiple selections.*
		This command opens a standard Windows **Open File** dialog to navigate to the location of a **.obj** file describing the subset geometry. This method can be used with meshes created in **Autodesk 3ds Max**, where individual surfaces can be selected and detached using the 3ds Max **Detach** command. These surfaces can then be exported as Wavefront **.obj** files, which **GABES** can read and convert into surface subsets. **N.B.** *This method is widely adopted in the* ***BESLE*** *[Ref] software.*
		This command implements the third method in GABES for creating surface subsets. It uses previously defined coordinate frames to map model coordinates into the selected frame within one of three coordinate systems: Cartesian, cylindrical, or spherical. The user can then specify coordinate ranges either by entering intervals manually or by using the provided slider bars, with selections updating graphically in real time. This method is particularly efficient for well-structured geometries with planar, cylindrical, or spherical sections, which is typical for most mechanical parts.

3.3.3 Boundary Conditions tab

The tab is subdivided into two (02) panels:

- **Displacement BCs:** Contains commands for assigning various Dirichlet-type boundary conditions, including total encaster (clamping), partial encaster (zero displacement in selected directions), vectorial displacement, and normal displacement. This panel also provides check boxes to toggle the visibility of the created fixations and displacement vectors.

- **Traction BCs:** Contains commands for assigning various Neumann-type boundary conditions, including vectorial traction, normal traction, and pressure. It also provides the **Free Surface** command to release a surface from the previously assigned boundary conditions and restore it to a traction-free state. A checkbox is also available to toggle the visibility of the created traction vectors.

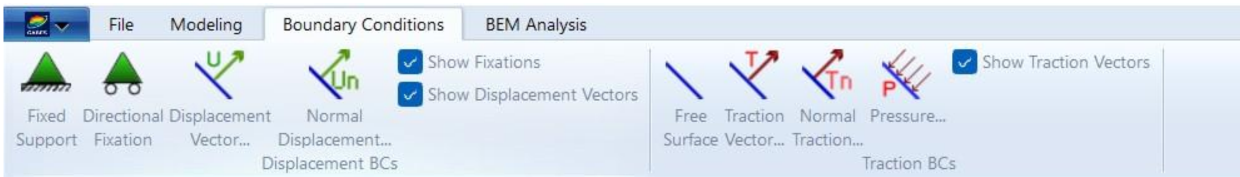
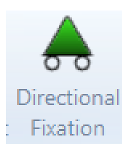
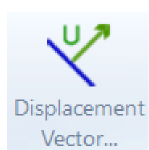
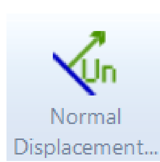
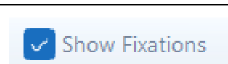
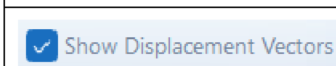
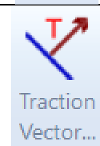
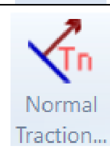
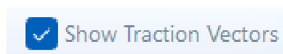


Figure 15: Boundary Conditions tab

These boundary condition assignment tools are only active when a **Surface Subset** is selected in the **Model Tree**. The commands on the **Boundary Conditions tab** are summarized in Table 3.

Table 5: Command buttons of the Boundary Conditions tab

Panel	Button	Action
Displacement BCs		Assigns a fully fixed boundary condition (zero displacement in all three directions) to the selected surface subset.
		Assigns zero displacement in the prescribed directions. The user is prompted with a dialog box to select the directions where zero displacement is applied.
		Assigns prescribed displacements in selected directions. The user is prompted with a dialog box to specify the directions and the corresponding displacement values.
		Assigns a displacement vector normal to the elements with a prescribed magnitude. The user is prompted with a dialog box to specify the displacement magnitude.
		Toggles the visibility of the created fixations on and off.
		Toggles the visibility of the created displacement vectors on and off.

Traction BCs	 Free Surface	Removes the previously assigned boundary conditions from the selected surface, restoring it to a traction-free state.
	 Traction Vector...	Assigns prescribed tractions in specified directions. A dialog box prompts the user to select the directions and enter the corresponding traction values.
	 Normal Traction...	Assigns a traction vector normal to the elements with a specified magnitude. A dialog box prompts the user to enter the traction magnitude.
	 Pressure...	This command is similar to the **Normal Traction** command but acts in the opposite normal direction. Positive pressure is applied inward.
	 Show Traction Vectors	Toggles the visibility of the created traction vectors on and off.

3.3.4 BEM Analysis tab

The tab is divided into two panels:

- **Analysis Panel:** Contains commands for defining material properties, setting numerical integration parameters, managing memory, and launching simulations.
- **Post-Processing Panel:** Contains commands and tools for various post-processing tasks, including enabling or disabling post-processing view mode, showing or hiding the deformed configuration of the body, setting the displacement magnification scale, choosing palette options, and obtaining internal domain responses either for individual points or as tabulated reports for multiple points.

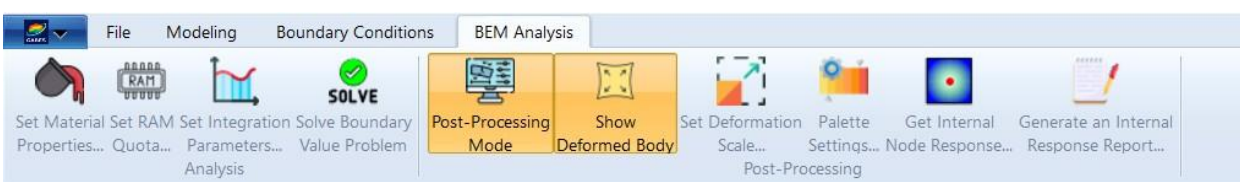
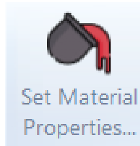
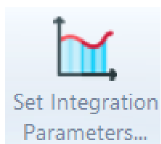
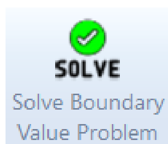
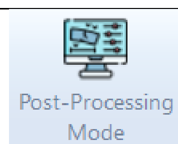
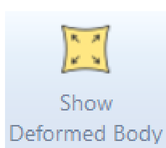
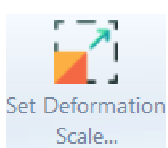
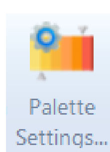
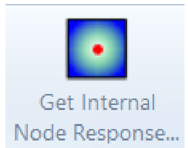
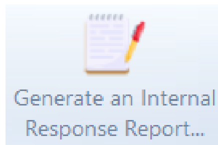


Figure 16: BEM Analysis tab

The commands on the **BEM Analysis tab** are summarized in Table 4

Table 6: Command buttons of the BEM Analysis tab

Panel	Button	Action
Analysis	 Set Material Properties...	Prompts the user with a dialog box to specify the material parameters. For linear elasticity, these include the Young's modulus E, the shear modulus μ, and the Poisson's ratio ν. By default, the parameters are set to the values for structural steel ($E = 200$ GPa, $\nu = 0.3$).
	 Set RAM Quota...	Prompts the user with a dialog box to specify the reserved RAM memory quotas for the BEM system matrices. If a matrix exceeds its allocated RAM quota, the remaining data is stored on the hard drive using the Out-Of-Core (OOC) storage method.
	 Set Integration Parameters...	This command opens a dialog box for configuring various integration parameters, including whether to use the Rigid Body method, selecting the cubature rule, or enabling an adaptive integration criterion.
	 SOLVE Solve Boundary Value Problem	This command is the most important operation in the application. Once the model is fully prepared and all parameters are correctly defined, the user clicks this button to solve the problem. Before execution, a dialog box prompts the user to select the $\mathscr{H}$-matrix error tolerance and the solver to use (H-LU, BiCGStab, or GMRES).
Post-Processing	 Post-Processing Mode	This button enables the Post-Processing View mode
	 Show Deformed Body	This button becomes active when the Post-Processing view mode is selected. After the problem is solved, this command allows the deformed body to be displayed by applying the computed displacement field to the mesh vertices using a uniform magnification scale.
	 Set Deformation Scale...	This button opens a dialog box for setting the deformation magnification scale. It becomes active only in Post-Processing view mode and after selecting the previous **Show Deformed Body** button
	 Palette Settings...	Opens a dialog box for configuring the number formatting displayed in the Color Palette. Tick values can be shown in either floating-point or scientific (exponential) notation, with a user-defined number of decimal digits.

		Opens a dialog box for computing specified field variables at an internal domain point or on a boundary point. The user enters the spatial coordinates of the point and selects the variables to be evaluated (displacements, stresses, or strains). The results are displayed directly in the dialog box.
		This command serves the same purpose as the previous one. However, instead of specifying a single point, the user is prompted to select a point cloud (multiple points), which can be generated automatically or imported from an input coordinate file. After computation, the results are exported to a tabulated report file in CSV format, readable by spreadsheet programs such as Microsoft Excel.

3.4 Dockable panels

Dockable panels are user interface components that can be attached to the edges of the main application window or displayed as independent floating windows, allowing users to customize the workspace layout according to their preferences and workflow. GABES has two distinct dockable panels, the **Model-Tree panel** and the **Information panel**. (See Figure 8).

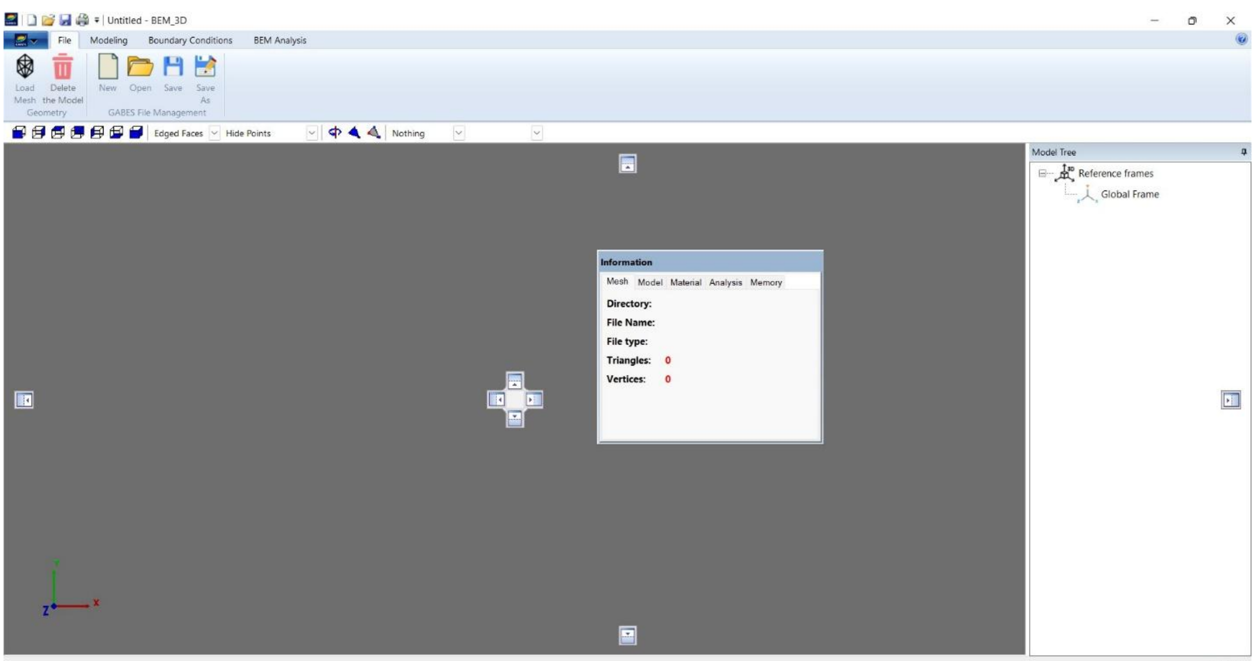


Figure 17: Dockable panels

3.4.1 The Model-Tree panel

The **Model Tree** is a floating and dockable panel that can be attached to any edge of the main frame window (see Figure 8). However, since the model hierarchy typically grows in the vertical direction, it is more practical to dock this panel on the **left or right side** of the main frame, where greater vertical space is available and navigation becomes more convenient.

The Model Tree panel displays the objects created during the modeling phase, including reference coordinate frames, the imported mesh, and the associated surface subsets. The panel is organized as a hierarchical tree structure composed of two (02) root nodes:

- **Reference Frames root**: contains all user-defined reference coordinate frames as well as the default global frame, which is permanently associated with the model's center of gravity and aligned with the center of the Client Area.
- **Mesh root**: becomes available once a mesh is loaded and adopts the name of the mesh file. All generated surface subsets are organized as child nodes of this root.

The Model Tree facilitates the modeling workflow by enabling easy modification of reference frames. It also allows the selection of surface subsets, which activates the boundary-condition tools and simplifies the assignment and modification of boundary conditions.

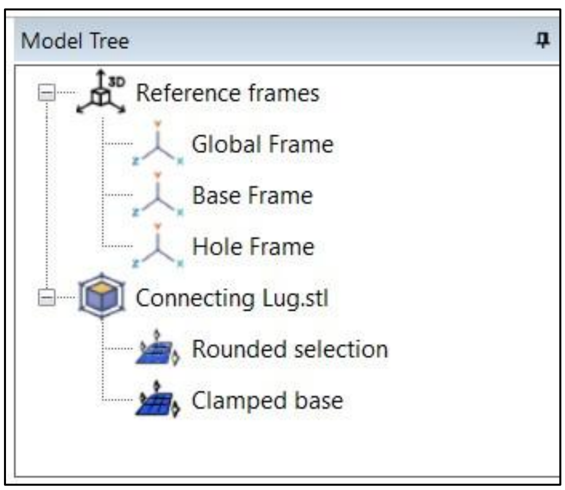


Figure 18: The Model-Tree panel

The objects displayed on the **Model-Tree panel** are summarized in Table 5

Table 7: The Model-Tree panel objects

Object	Description
Reference frames	Root node containing all reference coordinate frames defined in the model.
Global Frame	The global reference coordinate frame automatically created by the application and

	permanently associated with the model's center of gravity.
Hole Frame	A reference coordinate frame created by the user for custom positioning, orientation, or boundary-condition definition.
Connecting Lug.stl	Root node representing the imported mesh; its name is extracted from the mesh file title.
Clamped base	A surface subset created by the user from the loaded mesh, used for selection and boundary-condition assignment.

3.4.2 The Information panel

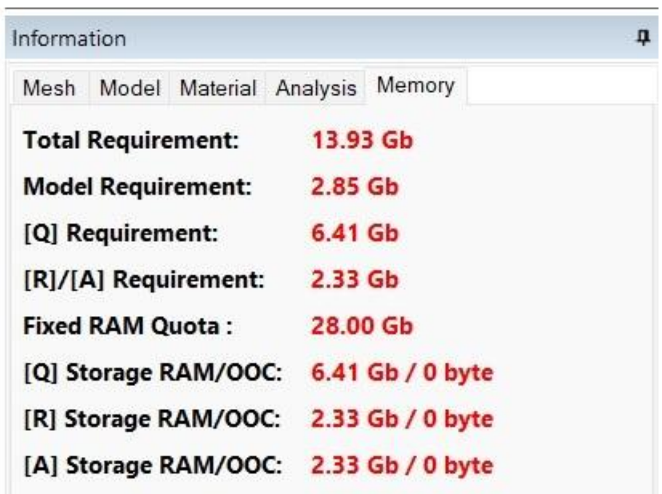


Figure 19: Infromation panel

The **Information panel** is a floating and dockable panel that can be attached to any edge of the main frame window (see Figure 8). It is particularly convenient to dock it below the Model Tree, as this preserves more space for the main view area.

This panel shows essential geometrical and numerical information, along with memory storage requirements. This helps the user understand the size of the model and the system matrices, making it easier to adjust simulation settings efficiently. (See Figure 10)

The various information is displayed in five (05) separated tabs:

3.4.2.1 Mesh tab

Displays mesh related information, like the location directory of the mesh file, the file title, the file type, the number of triangles and the number of vertices. (See Figure 11).

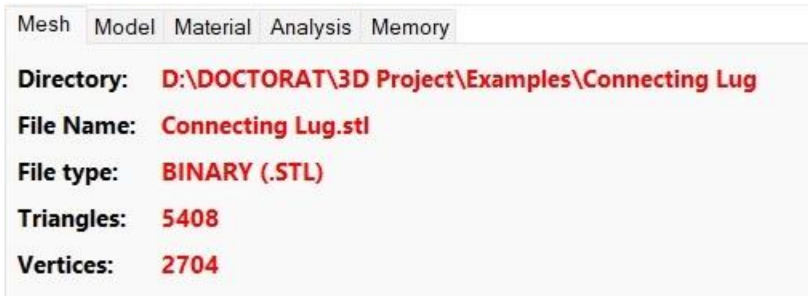


Figure 20: Information panel - Mesh tab

The information displayed under the **Mesh tab** is summarized in Table 6

Table 8: The information displayed under the Mesh tab

Information field	Description
Directory	The current hard disk directory where the mesh file is located.
File name	The name of the mesh file.
File type	The mesh file format, which can be either **STL** (binary or ASCII) or **OBJ** (ASCII).
Triangles	The total number of triangles in the mesh.
Vertices	The total number of vertices in the mesh.

3.4.2.2 Model tab

Displays model related information, like the discretization method, the number of DOF vertices which may differ from the number of the geometrical mesh vertices, the number of continuous and discontinuous elements and the resulting number of the algebraic DOFs which is three (03) times the number of DOF vertices due to the three (03) spatial directions. (See Figure 12)

Figure 21: Infromation panel - Model tab

The information displayed under the **Model tab** is summarized in Table 7

Table 9: The information displayed under the Model tab

Information field	Description
Disc. Type	The discretization method used, which can be Continuous, Discontinuous, or Disc. Sharp Edges. This corresponds to purely continuous elements, purely discontinuous elements, or automatically placed discontinuous elements at sharp edges and corners.
DOF Vertices	The number of degree-of-freedom (DOF) nodes, which may differ from the number of mesh vertices when discontinuous elements are present.
Cont. Elements	The total number of continuous elements in the model.
Disc. Elements	The total number of discontinuous elements in the model.
Algebraic DOFs	The resulting number of algebraic degrees of freedom, calculated as the number of DOF vertices multiplied by the three spatial dimensions. This value is important because it determines the overall size of the numerical system, including both matrices and vectors.

3.4.2.3 Material tab

The tab displays three material parameters: Young's modulus, shear modulus, and Poisson's ratio. In GABES, Young's and shear modulus are unitless, since the application operates in a dimensionless system. The actual units of traction and stress depend on the user's choice of length units when creating the mesh. For example:

- If geometry is interpreted in meters, traction is in Pascals (Pa).
- If geometry is in millimeters, traction is in Megapascals (MPa).

By default, the material values correspond to structural steel, and dimensions are assumed to be millimeters. If the user chooses a different unit system, the Young's modulus and shear modulus must be updated to remain consistent with the dimensionless system. This can be done via the Set Material Properties… button on the BEM Analysis tab of the ribbon.

N.B: Since Young's modulus and shear modulus are closely related, setting one is sufficient; the other will be computed automatically.

Figure 22: Information panel - Material tab

3.4.2.4 Analysis tab

This tab displays the matrix dimensions along with the current numerical integration parameters. It provides users with a clear overview of the system size and the specific integration settings applied during matrix construction, such as the method used to evaluate CPV integrals and whether an adaptive criterion scheme or a fixed cubature rule is employed.

Figure 23: Information panel - Analysis tab

The information displayed under the **Analysis tab** is summarized in Table xx

Information field	Description
CPV Evaluation	The method used to evaluate Cauchy Principal Value (CPV) integrals, which can be either **Rigid Body Displacement** or **Direct Evaluation**. This option can be modified using the **Set Integration Parameters…** button located under the **BEM Analysis** tab in the ribbon
Integ. Scheme	Specifies the integration scheme used to evaluate regular integrals, which can be either an **Adaptive Integration Criterion** or a **Fixed Cubature** with the corresponding number of integration points. This option can also be modified using the **Set Integration Parameters…** button.
[Q] matrix	The dimension of the **Q** matrix, which groups all influence coefficients associated with the displacement U_{ij} kernel. This matrix is generally rectangular rather than square. It becomes a square matrix only when all elements are discontinuous and no nodes are shared between elements.
[R] & [A] matrices	The dimension of the **R** and **A** matrices, which represent, respectively, the grouped influence coefficients originating from the traction T_{ij} kernel and the assembled linear system matrix. These two matrices are always square and share the same dimension.
{b} & {x} vectors	The dimension of the **{b}** and **{x}** vectors, which represent, respectively, the right-hand side vector and the unknown vector of the linear system $[A]\{x\} = \{b\}$.

3.4.2.5 Memory tab

GABES uses **H-Matrix techniques** to accelerate the solution of the linear system but does not yet benefit from H-Matrix memory compression. This approach simplifies the handling of boundary conditions by assembling fully populated matrices. Once assembly is complete, the system is converted into an H-Matrix format, enabling fast solvers while keeping the assembly algorithm simple.

Because full matrices are used during assembly, memory requirements can be significant. To manage this, GABES employs an **Out-of-Core** storage strategy, storing part of the data on disk to avoid exceeding available RAM. The **Memory** tab displays memory requirements in gigabytes, helping users determine how much RAM to allocate to each matrix. RAM allocation can be adjusted via the **Set RAM Quota…** button under the **BEM Analysis** tab of the ribbon. This approach allows simulation of relatively **large meshes** while keeping the matrix assembly algorithm simple.

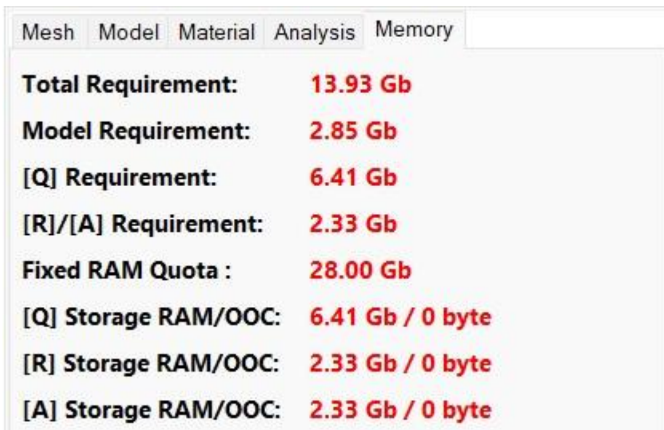


Figure 24: Information panel - Memory tab

The information displayed under the **Memory tab** is summarized in Table xx

Information field	Description
Total Requirement	The total amount of memory required for the simulation.
Model Requirement	The memory required for the internal representation of the mesh, including BEM discretization data and the Direct3D structures used to render the model.
[Q] Requirement	The total memory required to store the **Q** matrix. This typically represents the largest portion of memory usage because the **Q** matrix is generally rectangular and has a large horizontal dimension.
[R]/[A] matrices	The total memory required to store each of the **R** and **A** matrices.
Fixed RAM Quota	The maximum amount of memory allowed in Random Access Memory (RAM). This quota is defined by the user and cannot

	exceed the available system RAM. The specified RAM is distributed among the **Q**, **R**, and **A** matrices.
[Q] Storage RAM/OOC	The actual storage distribution of the **Q** matrix, showing the portion stored in RAM (left) and the portion stored on disk using Out-of-Core storage (right), separated by a slash (/).
[R] Storage RAM/OOC	The actual storage distribution of the **R** matrix, showing the portion stored in RAM (left) and the portion stored on disk using Out-of-Core storage (right), separated by a slash (/).
[A] Storage RAM/OOC	The actual storage distribution of the **A** matrix, showing the portion stored in RAM (left) and the portion stored on disk using Out-of-Core storage (right), separated by a slash (/).

3.5 The status bar

Status bars are user interface components supported by MFC and commonly included in application templates. They are widely used in desktop applications to display contextual command descriptions, application state information, and keyboard state indicators such as Caps Lock or Num Lock.

In GABES, the status bar is used exclusively to display the global coordinates of elements and vertices when they are hit-tested (i.e., when the mouse cursor passes over selectable geometry). This feature provides useful feedback during modeling operations, for example when creating coordinate frames and attaching them to well-defined locations.

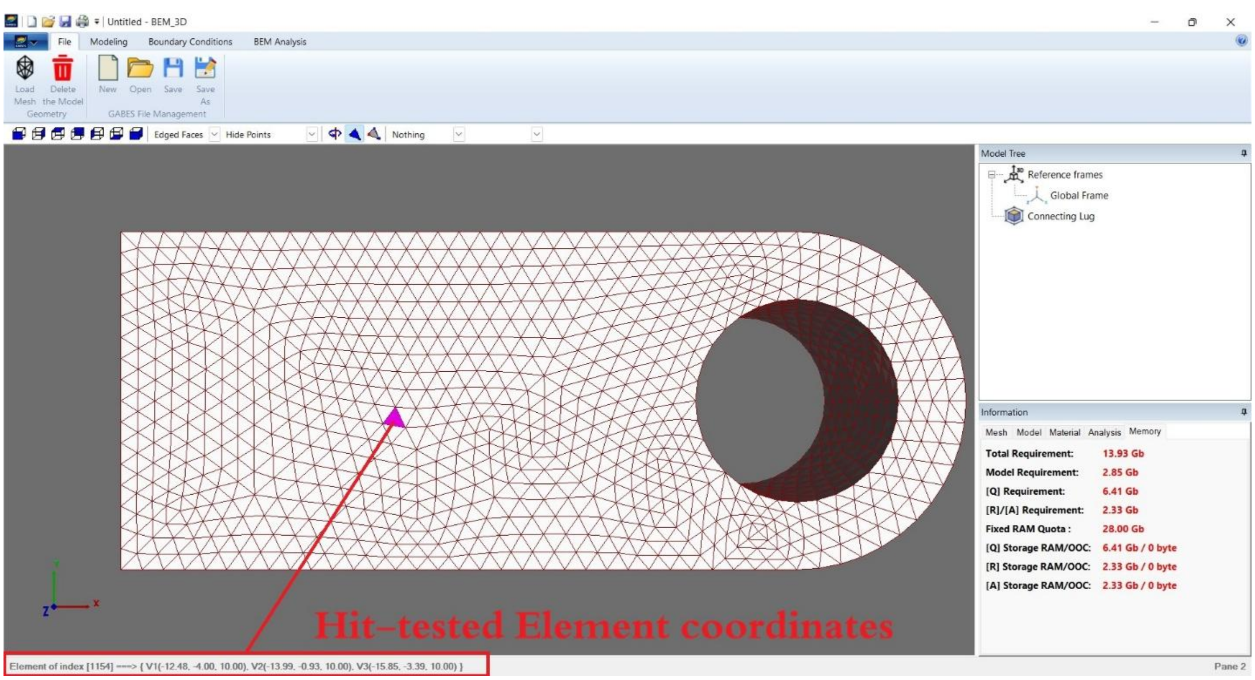


Figure 25: Status bar

4 USING GABES

4.1 Mesh creation

GABES does not implement internal geometry creation or mesh generation algorithms. Instead, it relies on established external software for these tasks. The application operates exclusively on **triangular surface meshes**, which are naturally suited to the Boundary Element Method (BEM) due to its boundary-only discretization approach.

GABES supports common mesh formats such as **Stereolithography (.stl)** and **Wavefront Object (.obj)**, both widely used in CAD, visualization, and multimedia applications.

Polygonal modeling software such as **Blender** and **Autodesk 3ds Max** can directly produce compatible and well-structured meshes by exporting models in **(.stl)** or **(.obj)** format. In contrast, parametric CAD systems such as **SolidWorks**, which are based on **Boundary Representation (B-Rep)** geometry, require conversion to triangular surface meshes before being used in GABES.

Although SolidWorks can export **(.stl)** files, this functionality is primarily intended for visualization or additive manufacturing and provides limited control over mesh quality and refinement. For better mesh control, SolidWorks models should be exported to neutral exchange formats such as **(*.STEP*)** or **(.IGES)**, and then meshed using **Gmsh**, an open-source mesh generator that allows mesh refinement and optimization. The generated meshes can then be exported in **(.stl)** or *.obj* formats compatible with GABES.

Gmsh is available at: https://gmsh.info

4.2 Mesh loading

Once a good-quality mesh is created and exported in **(.stl)** or **(.obj)** formats, it is recommended to store it in a dedicated folder. When the mesh is loaded into GABES, its directory automatically becomes the application's working directory, and all generated files are saved there. Using separate folders for each model helps keep project files organized and easy to manage.

A mesh is loaded using the **Load Mesh** button situated in the **File** tab of the ribbon:

An **Open File** dialog box will appear, allowing the user to browse and select the desired mesh file. Before the mesh is fully loaded, the **Set Mesh Type** dialog box appears, prompting the user to choose the discretization method to be applied to the mesh. In this context, discretization refers to the boundary element type assigned to each triangular mesh element.

GABES uses two types of linear elements: continuous and discontinuous elements. Both element types have three (03) degrees of freedom (DOF) nodes, differing only in their location, as illustrated in Figure 26.

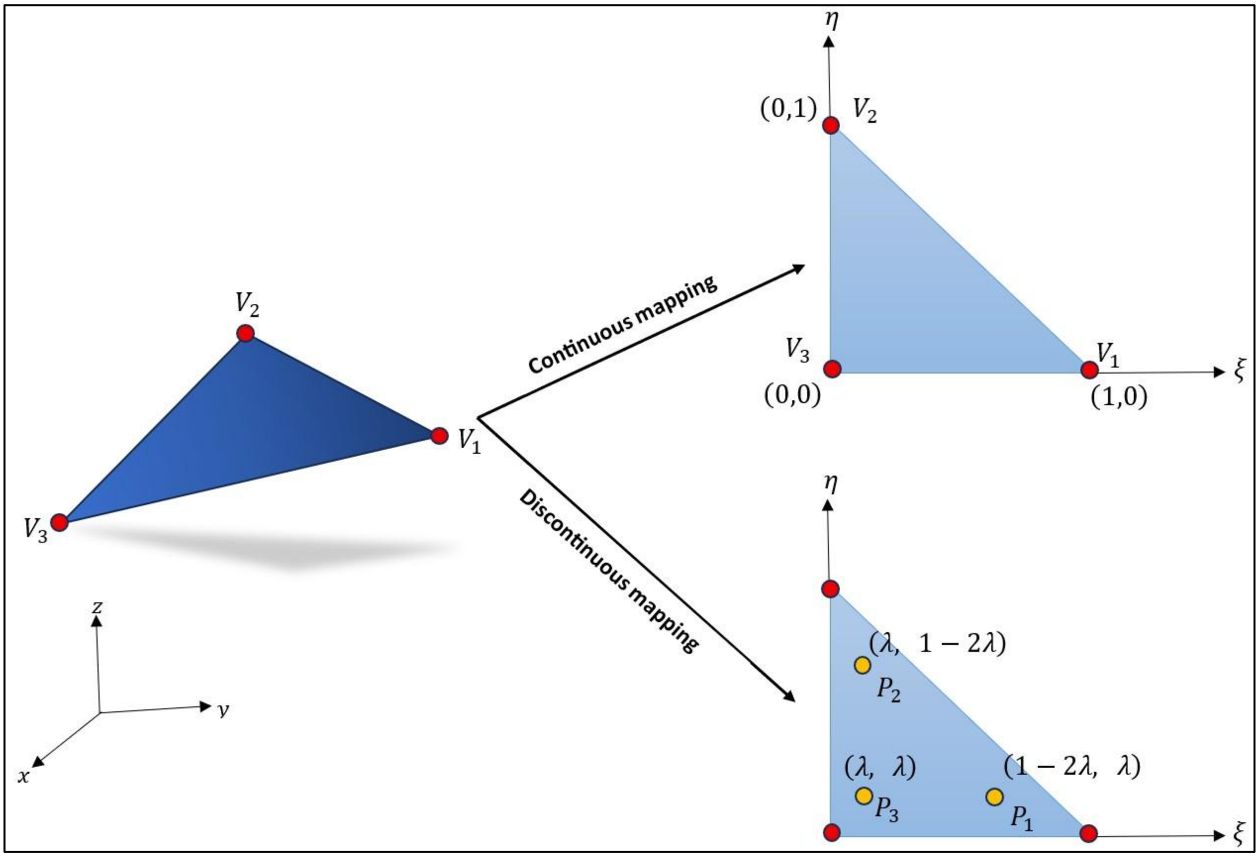


Figure 26: Boundary Element Types

Continuous element nodes coincide with the three geometric vertices of each triangular element, resulting in node sharing between neighboring triangles. In contrast, discontinuous element nodes are located inside the triangle, creating independent degrees of freedom (DOFs) for each element.

Continuous elements, due to node sharing, significantly reduce the overall number of degrees of freedom. However, they tend to yield poor results at sharp edges and corners because of the abrupt variation of the outward normal vector between adjacent elements. On the other hand, discontinuous elements provide better accuracy near edges and corners, and the assembly of the numerical linear system is simpler because no DOF reduction is required due to the absence of node sharing. Their main drawback is the generation of larger linear systems with a higher number of degrees of freedom, which can negatively affect computational performance.

GABES addresses this compromise by automatically assigning continuous elements to smooth boundaries while enforcing discontinuous elements at sharp edges and corners where the angle between adjacent element normals exceeds 45°. This automatic placement of discontinuous elements in sharp regions is the default behavior in GABES. However, this behavior can be overridden by selecting a uniform element type through the **Set Mesh Type** dialog box, which appears before loading a mesh and offers three options: **Purely Continuous**, **Purely Discontinuous**, or **Discontinuous at Sharp Edges**. As shown in Figure 27.

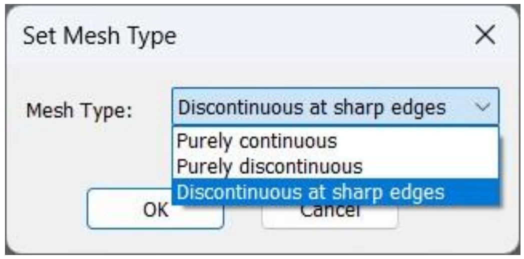


Figure 27: Set Mesh Type dialog box

4.3 Modeling operations

Modeling operations indicate all the operations realized by the user for preparing the model for simulation. In GABES context modeling operations consist generally on two main tasks:

- Reference frames definition
- Surface subsets creation

4.3.1 Reference frames definition

Reference frames are orthogonal, right-handed coordinate systems that include the default global frame as well as any custom frames created by the user. Their primary function is to store and organize the coordinates of three-dimensional objects, such as vertices, nodes, and elements. Reference frames are especially useful for facilitating the interactive creation of surface subsets and for providing intuitive coordinate mapping during post-processing. By allowing the user to define points relative to well-defined geometrical locations, reference frames simplify the interpretation and input of coordinates.

This functionality is particularly important because each loaded model is automatically centered on the screen, with all its coordinates mapped to the global frame based on the model's center of gravity. As a result, the raw global coordinates may not be intuitive for the user. Custom reference frames enable more meaningful and user-friendly coordinate systems, making modeling, selection, and post-processing operations easier and more precise.

4.3.1.1 Create a new Reference Frame

To add a new reference frame, navigate to the **Modeling** tab in the ribbon and select the **Add Reference Frame…** button. The **Create/Modify Reference Frame** dialog box will appear. This is a **modeless dialog**, which means that the main application window remains active, allowing the user to interact with the model, such as selecting vertices and elements graphically, while defining the new reference frame.

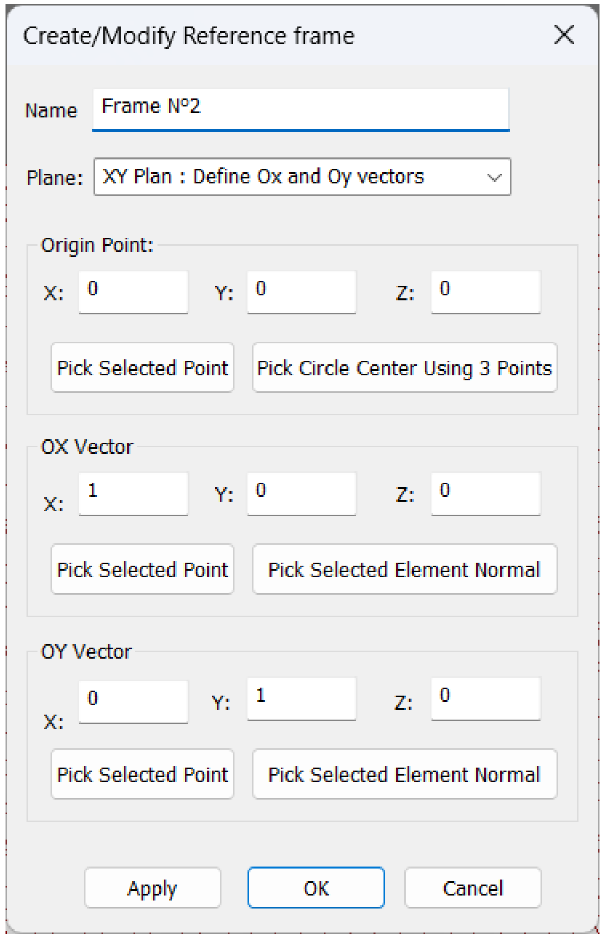


Figure 28: Create/Modify Reference Frame

4.3.1.1.1 The use of the Create/Modify Reference Frame dialog for coordinate system definition

This dialog box is shown in **Figure 28** and allows the creation of right-handed orthogonal coordinate systems using a simplified approach. Normally, defining a 3D coordinate system requires six quantities: the coordinates of the origin point and three Euler angles. However, Euler angles can be difficult for users to visualize. To simplify the process, GABES does not require the Euler angles. Instead, the user only needs to define a single reference plane **OXY**, **OYZ**, or **OZX** by specifying two orthogonal unit vectors with a null dot product. The third unit vector is automatically computed using the vector cross product, ensuring a valid orthogonal right-handed coordinate system.

The procedure for defining a reference frame is as follows: first, define the **origin point**, then select a reference plane using the **Plane** dropdown combo-box, and finally assign the unit vectors for the axes of the chosen plane.

The origin point can be defined either manually or graphically:

1. **Manual Input:** Enter the desired coordinates directly, expressed in the global coordinate system.

2. **Graphical Methods:**

   - **Pick Selected Point:** Use this method to place the frame on a known vertex, such as a corner or edge tip. It requires a prior and valid graphical selection of a single point; clicking the button places the origin of the frame at that selected point.
   - **Pick Circle Center Using 3 Points:** Use this method to place the frame at the center of a circular edge. This is useful, for example, for defining a cylindrical coordinate mapping with the Z-axis aligned along the cylinder axis. It requires a valid graphical selection of three points on the circular edge; clicking the button places the frame origin at the circle center.

The unit vectors can also be defined either manually or graphically:

1. **Manual Input:**
   Enter the three (03) vector components directly.

2. **Graphical Methods:**

   - **Pick Selected Point:** Use this method to orient the vector from the origin toward a graphically selected point, such as a corner or an edge tip. It requires a prior valid graphical selection of a single point. Clicking this button creates a unit vector oriented from the origin toward the selected point.
   - **Pick Element Normal:** Use this method to orient the vector along the outward normal direction of a selected element. It requires a prior valid selection of a single element. Clicking this button creates a unit vector perpendicular to the selected element and oriented outward. This method is useful for defining a vector normal to a known surface by simply selecting one of its elements.

The **Create/Modify Reference Frame** dialog box can also be used to modify an existing coordinate frame, except for the default global frame, which is fixed and tied to the model's center of gravity. A user-defined frame can be modified by simply double-clicking it in the **Model Tree** panel. This action opens the **Create/Modify Reference Frame** dialog box, allowing the user to change the frame's location or orientation.

It is worth noting that all created frames are displayed using a small **gizmo**, allowing users to identify their exact location without obscuring or interfering with the 3D model. The gizmo axes are temporarily scaled only during frame creation, modification, and surface subset creation to improve visibility and facilitate interactive operations.

4.3.2 Surface subsets creation

As their name indicates, **surface subsets** are user-defined portions of the boundary and are mandatory for boundary condition assignment. They provide a flexible and efficient method for defining boundary conditions. All boundary condition tools located under the **Boundary**

Conditions tab of the ribbon are enabled only after a valid surface subset is selected in the **Model Tree** panel.

GABES provides three (03) methods for creating surface subsets:

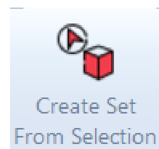
- **Graphical Selection**
- **Import from an (.obj) file**
- **Interactive Range Selection**

All these three methods command buttons are accessible under the **Modeling tab** of the ribbon

4.3.2.1 Graphical Selection

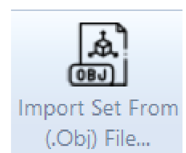
The user can select multiple elements after enabling the **Element Selection** button on the **View** toolbar. To select multiple elements, the **CTRL** key must be held while selecting.

When a valid element selection exists, the **Create Set From Selection** button becomes enabled. Clicking this button automatically converts the current selection into a surface subset with an automatically generated name, which appears in the **Model Tree** under the mesh root. The newly created subset can be renamed directly within the **Model Tree** panel.



4.3.2.2 Import from an (.obj) file

This method can be used with meshes created in **Autodesk 3ds Max**, which provides the ability to create sub-surfaces of the original model using its **Detach** command. These detached surfaces can be exported individually as **Wavefront (.obj)** files. GABES can then convert these detached surfaces into boundary subsets using the **Import Set From (.obj) File** command.



Clicking this button opens a dialog box that allows the user to browse and select the desired file. This method is commonly used in **BESLE BEM** software. Once the subset is created, GABES automatically assigns it a name based on the filename of the imported surface.

When using this method, it is preferable to store the detached surface **(.obj)** file in the same directory as the mesh file, which serves as the default working directory for the model.

4.3.2.3 Import from an (.obj) file

This method is the most practical and versatile among the three. It relies extensively on reference frames, allowing elements to be selected using coordinate ranges defined in Cartesian, cylindrical, or spherical coordinate systems. By placing user-defined reference frames at key geometrical locations on the 3D model, the user can easily determine the coordinate ranges required for selection.

For example, to select a surface with known dimensions, the user can place a reference frame at one corner of the surface and orient its axes so that the selection ranges can be conveniently defined using the X, Y, and Z coordinates. Another example involves cylindrical surfaces, where the user can place a reference frame at the center of the circular base and orient the Z-axis along the cylinder height. This configuration allows cylindrical regions to be easily selected by specifying ranges for the radius, angle, and height. A similar approach can also be applied to spherical geometries.

In addition to manual input, range selection can be performed interactively using slider track bars that define the minimum and maximum range values. The selection is updated graphically as the sliders are adjusted. When entering range values manually, the selection can be refreshed by clicking the **Apply** button.

Range-based subsets are created using the **Create Range Subset…** button, which opens the **Create Range Subset** modeless dialog box, as shown in Figure 29. The selection ranges are defined by choosing a coordinate frame from the **Coordinate Frame** drop-down list, which highlights and scales the frame to improve visibility in the 3D scene. The coordinate system can then be selected (Cartesian, Cylindrical, or Spherical), which updates the minimum and maximum values for each dimension. The ranges can be adjusted interactively using the slider bars or by manually entering values and updating the selection using the **Apply** button. The dialog also allows the user to specify the subset name. Clicking **OK** converts the created selection into a surface subset that appears in the **Model-Tree** panel.

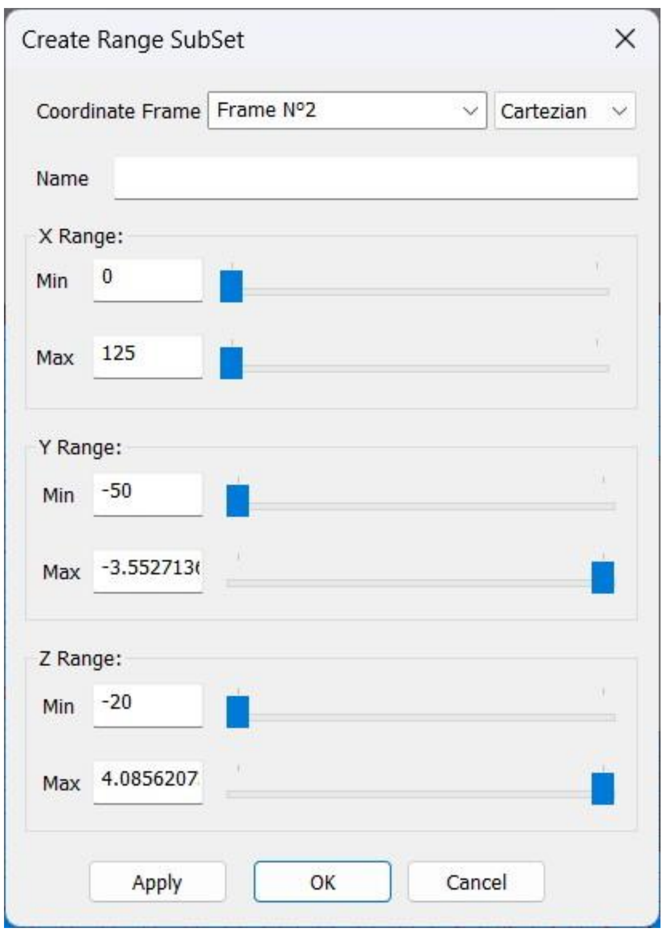


Figure 29: Create Range Subset dialog

4.4 Boundary conditions assignment

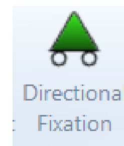
The boundary conditions are simply assigned after the selection of a subset from the Model-Tree. When a subset is selected the boundary condition assignment buttons located under the **Boundary Conditions tab** become activated.

4.4.1 Displacement (Dirichlet) boundary conditions

4.4.1.1 Fixed Support

This condition is applied using the **Fixed Support** button, which imposes zero displacement in all three spatial directions $U_1 = U_2 = U_3 = 0$.

4.4.1.2 Directional Fixation



This condition is applied using the **Directional Fixation** button, which imposes zero displacement along the selected directions. Clicking this button opens the dialog box shown in Figure 30, where the user can select the directions to be restrained.

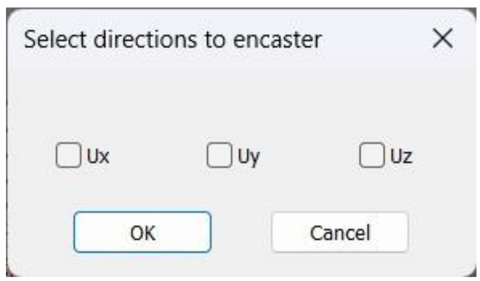
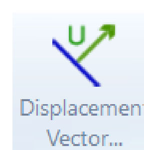


Figure 30: Select directions to encaster

4.4.1.3 Displacement Vector



This tool allows the user to apply specific displacements in chosen directions. Unlike the **Directional Fixation** tool, which restricts movement, this tool imposes a non-zero displacement. To use it, click the **Displacement Vector…** button to open the dialog box shown in Figure 31. Then, check the desired displacement component to enable its edit box and enter the required value.

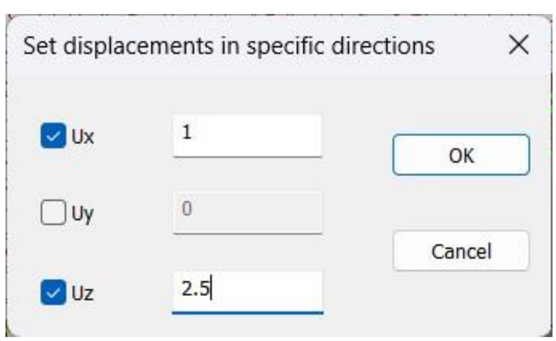


Figure 31: Set Displacements in specific directions

4.4.1.4 Normal Displacement

This tool allows the application of a non-zero displacement along the normal direction of the elements. It is assigned using the **Normal Displacement…** button, which opens the dialog box shown in Figure 32, where the displacement magnitude can be specified. A positive value applies the displacement in the outward normal direction, while a negative value applies it in the inward normal direction.

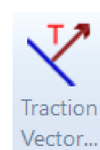
Figure 32: Set the normal displacement magnitude

4.4.2 Traction (Neumann) boundary conditions

4.4.2.1 Free Surface

This command removes all applied boundary conditions and restores the default free-surface condition, characterized by unknown displacement and zero traction. It is assigned by using the **Free Surface** button.

4.4.2.2 Traction Vector



This tool is similar to the Displacement Vector tool; however, it applies traction components in the selected directions. To use it, click the Traction Vector… button to open the dialog box shown in Figure 33. Then, select the desired traction component to enable its edit box and enter the required value.

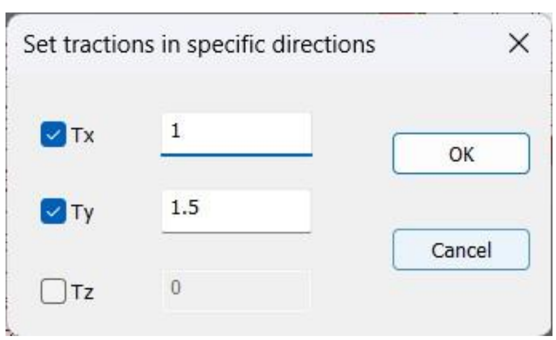
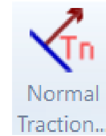


Figure 33: Set tractions in specific directions

4.4.2.3 Normal Traction



This tool applies traction along the element normal direction. It is accessed through the **Normal Traction…** button, which opens the dialog box shown in Figure 34, allowing the user to specify the traction magnitude. Positive values apply traction in the outward normal direction, whereas negative values apply it in the inward normal direction.

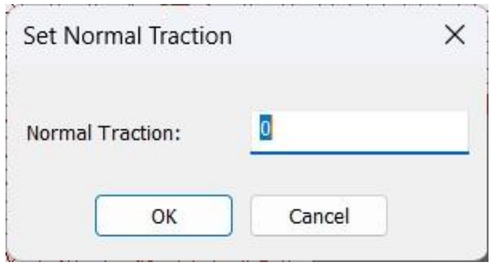


Figure 34: Set Normal traction

4.4.2.4 Pressure

This tool is equivalent to the **Normal Traction** tool but uses the opposite sign convention. In this case, a positive pressure value represents a load acting toward the surface, while a negative value represents a load acting away from the surface.

4.5 Analysis phase

Once the mesh is loaded, and the boundary conditions assigned, the boundary value problem is fully defined and the model become ready for simulation. However, some parameters need to be verified and set by the user before solving the boundary value problem.

4.5.1 Setting options before solving

4.5.1.1 The material properties

In linear elasticity, a material is characterized by Young's modulus E and Poisson's ratio ν. In some cases, the shear modulus μ is used instead of Young's modulus. These elastic material properties can be defined using the **Set Material Properties…** button, which opens the **Set Material Properties** dialog, as shown in Figure 35. By default, the material is configured as structural steel with a Young's modulus of 200,000 MPa, assuming dimensions are expressed in millimeters.

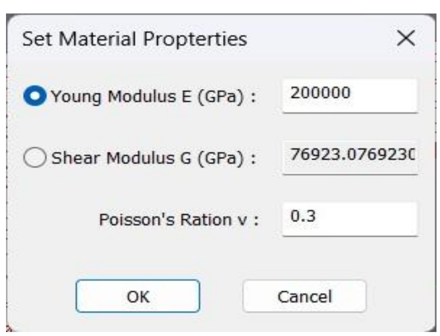
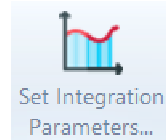


Figure 35: Set Material Properties

4.5.1.2 The numerical integration parameters



Some numerical integration parameters must be defined by the user. These include the method used to evaluate Cauchy Principal Value (CPV) integrals, which can be computed either directly or using the Rigid Body Displacement approach. The user must also specify the integration scheme for evaluating regular integrals, either through an adaptive criterion or a fixed cubature order. These options can be configured using the **Set Integration Parameters…** button, which opens the dialog box shown in Figure 36.

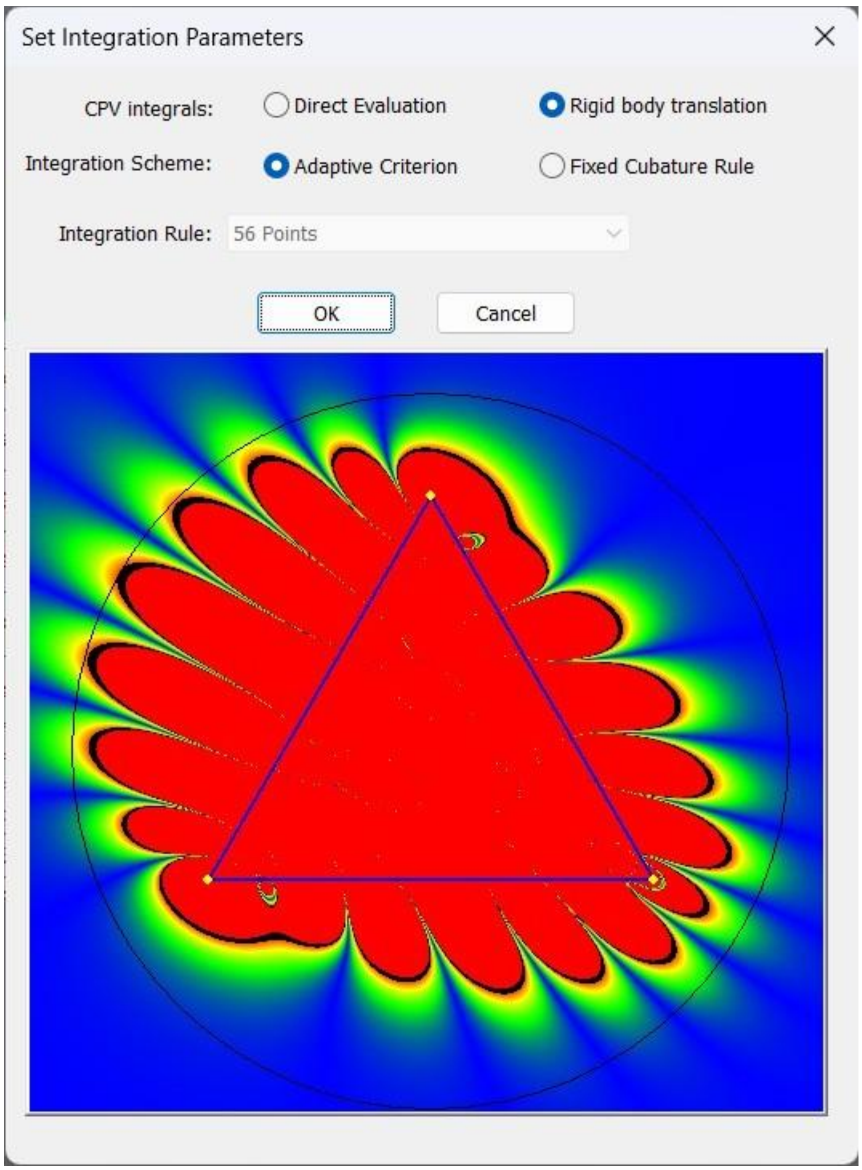


Figure 36: Set Integration Parameters

4.5.1.3 The RAM Quotas Allocation

The RAM quotas can be set for each matrix using the **Set RAM Quota** dialog (See Figure 37), accessible by clicking the **Set RAM Quota**… button

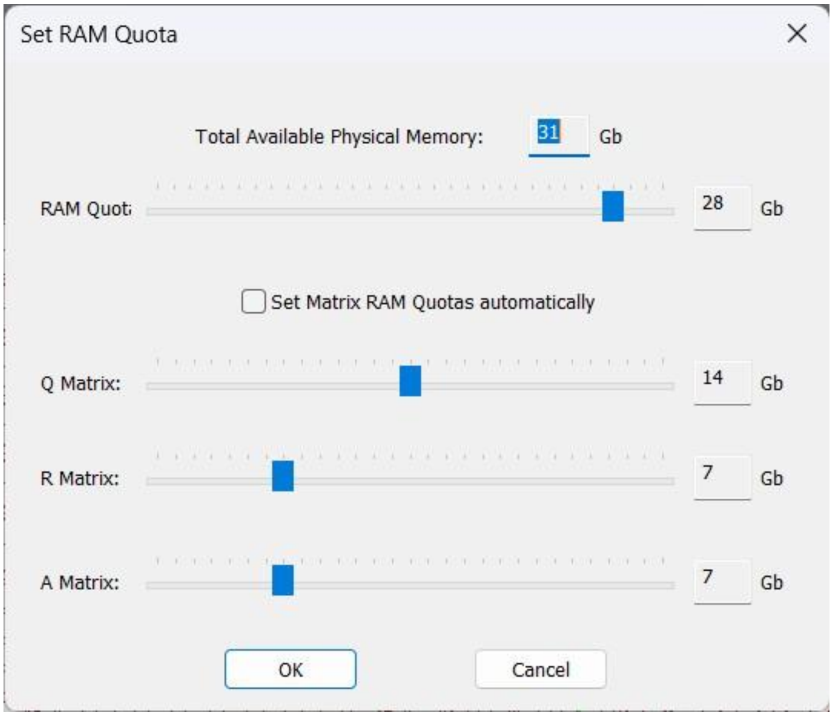
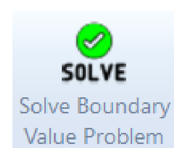


Figure 37: Set RAM Quotas

4.5.2 Solving the linear system



The simulation is started by clicking the **Solve Boundary Value Problem** button. This opens a dialog box where the user can define the H-Matrix error threshold and select the solver type. The available options include the direct **H-LU** solver or one of the iterative solvers (**GMRES** or **BiCGStab**), for which the maximum number of iterations must be specified (see Figure 38).

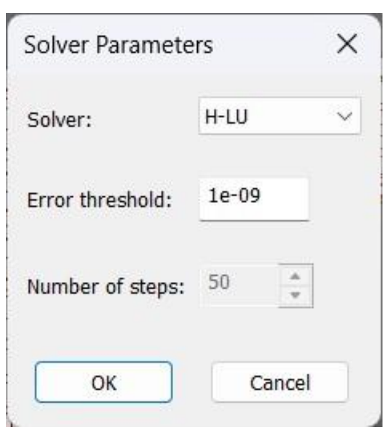


Figure 38: Solver parameters