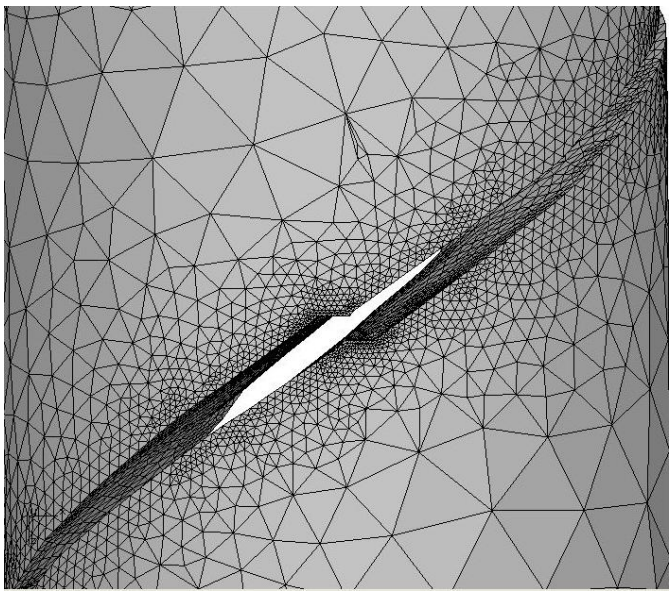
FRANC3D

NASTRAN™ Tutorial

NASTRAN is a registered trademark of NASA



Version 9.2

Fracture Analysis Consultants, Inc
https://www.fracanalysis.com/

Revised: Oct 2026

Table of Contents:

1.0 Introduction 4

2.0 Tutorial 1: Crack Insertion and Growth in a Cube 6

2.1 Step 1: Create the NASTRAN FE Model 6

2.2 Step 2: Reading NASTRAN FE Model into FRANC3D 7

Step 2.1: Importing Complete NASTRAN FE Model 7

Step 2.2: Select the Retained Items in the Local FE Model 9

2.3 Step 3: Importing and Subdividing the Model 12

2.4 Step 4: Insert a Crack 19

Step 4.1: Define a new Crack from FRANC3D Menu 19

Step 4.2: Insert Cracks from Files 25

2.5 Step 5: Static Crack Analysis 28

Step 5.1: Select Static Crack Analysis 28

Step 5.2: Select NASTRAN Analysis Options 30

2.6 Step 6: Compute Stress Intensity Factors 32

Step 6.1: Select Compute SIFs 32

2.7 Step 7: Manual Crack Growth 34

Step 7.1: Select Grow Crack 34

Step 7.2: Specify Fitting and Extrapolation 37

Step 7.3: Specify Crack Front Template 39

2.8 Step 8: Automatic Crack Growth 41

Step 8.1: Open FRANC3D Restart File 41

Step 8.2: Select Crack Growth Analysis 41

Step 8.3: Specify Growth Rules 42

Step 8.4: Specify Fitting and Template Parameters 44

Step 8.5: Specify Growth Plan 45

Step 8.6: Analysis Code Settings 46

2.9 Step 9: SIF History and Fatigue Life 48

Step 9.1: Select SIFs Along a Path 49

Step 9.2: Select SIFs For All Fronts 51
Step 9.3: Select Fatigue Life Predictions 51
Appendix A: NASTRAN Keywords 62

1.0 Introduction

This tutorial introduces the fracture simulation capabilities of FRANC3D Version 9 and NASTRAN; we use Siemens Simcenter Nastran 2206, but MSC versions of NASTRAN should work also. The NASTRAN version must be set in the Preferences. FRANC3D is introduced by analyzing a single surface crack in a NASTRAN finite element (FE) model of a cube.

Subsequent tutorials (see the FRANC3D Tutorials #2-15 document) build on this first example and describe additional capabilities and features of the software. It is intended that the user perform operations as they are presented, but you can experiment and consult other reference documentation whenever necessary.

The FRANC3D menu and dialog button selections are indicated by bold text, such as **File** menu. Window regions along with dialog options, fields and labels are underlined. Model names and file names are indicated by *italic text.*

The FRANC3D main window is shown in Fig 1.1. When you start the tutorial, the first thing to do is set the working directory using the **File →Work Directory** menu option, Fig 1.1.

Before or after a model is imported, you can set the FE model units, Fig 1.2. The selected units are applied to plots and figures that FRANC3D displays and are used during crack growth in combination with fatigue crack growth rate data, which also has units.

You can also set your preferences; it is helpful to set the path to the NASTRAN executable. You can view the preferences using the **Edit →Preferences** menu option. The Preferences dialog is described in Section 5.4 of the FRANC3D Reference document.

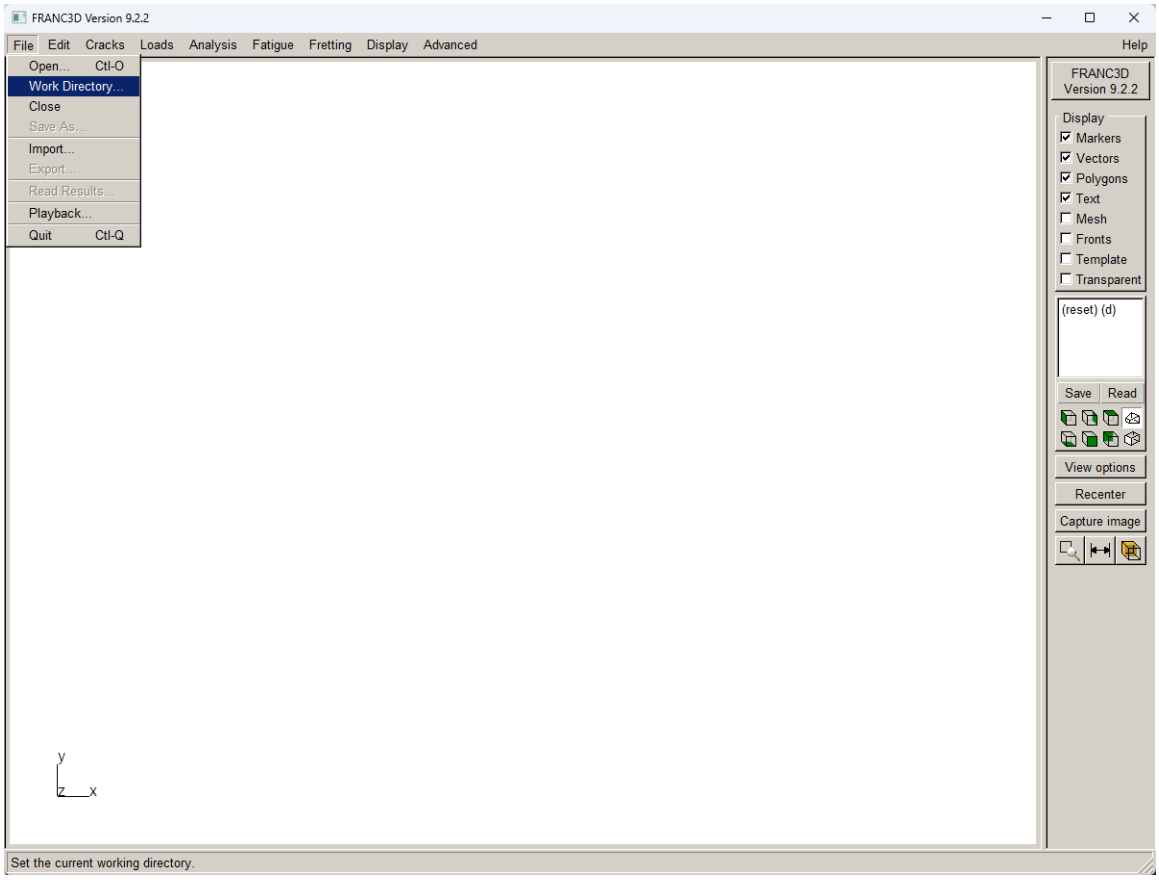


Figure 1.1 FRANC3D main window – Set Work Directory menu option.

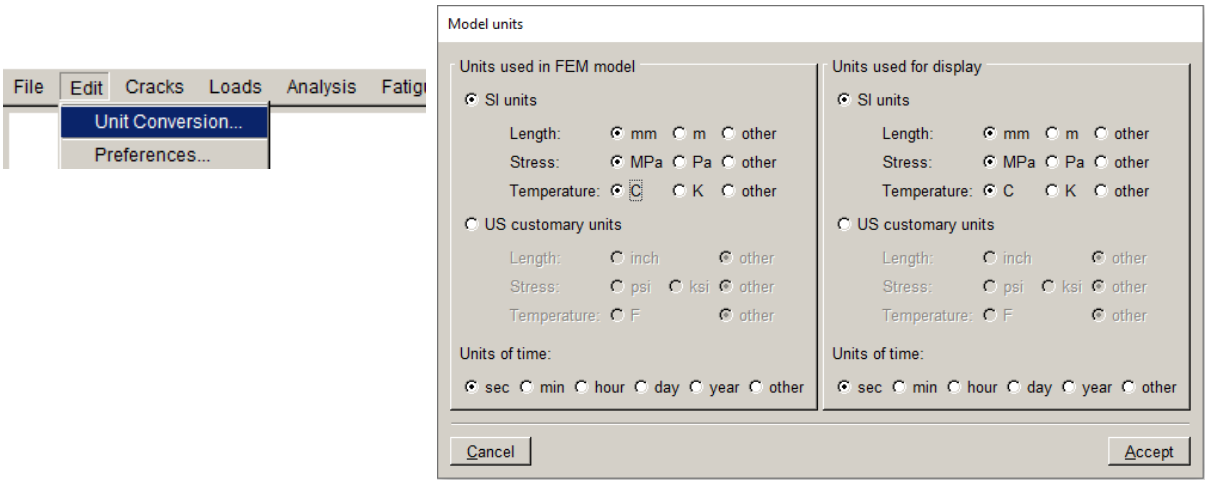


Figure 1.2 FRANC3D FE model unit conversion menu option and dialog.

2.0 Tutorial 1: Crack Insertion and Growth in a Cube

In this tutorial, a single surface crack in a cube under far-field tension is modeled. It is assumed that the user is familiar with a pre-processor for NASTRAN. Once the FE model is created, the FRANC3D steps to: 1) read the model, 2) insert a crack, 3) perform the NASTRAN analysis, and 4) compute stress intensity factors, are all described.

2.1 Step 1: Create the NASTRAN FE Model

First, you must create a cube model using any pre-processor for NASTRAN. We outline the steps to create the FE model with boundary conditions to ensure that you have a model that can be used with FRANC3D.

1. Create a 10x10x10 cube geometry; units of length are mm.
2. Subdivide the edges for meshing, using 10 to 20 subdivisions.
3. Define the element type as quadratic; use brick or tetrahedral elements to mesh the volume.
4. Define the material properties; elastic modulus (E) is 10000 and Poisson's ratio is 0.3; units for E are MPa.
5. Boundary conditions consist of displacement constraints on the bottom surface and uniform traction (a negative pressure) on the top surface. The bottom surface is constrained in the y-direction, the bottom left edge is also constrained in the x-direction, and the point at the origin is also constrained in the z-direction. The top surface traction is 10 MPa.
6. Write the *.bdf* or *.nas* file (*Nastran_Cube.nas*).

   Note that the SUBCASE information in the *.bdf* file should include the PUNCH option for the displacement:

```
SUBCASE 1
   TITLE = NX Nastran Static Analysis Set
```

DISPLACEMENT(PLOT,PUNCH) = ALL

The resulting model should appear as in Fig 2.1. The symbols for the boundary conditions are displayed attached to the mesh model.

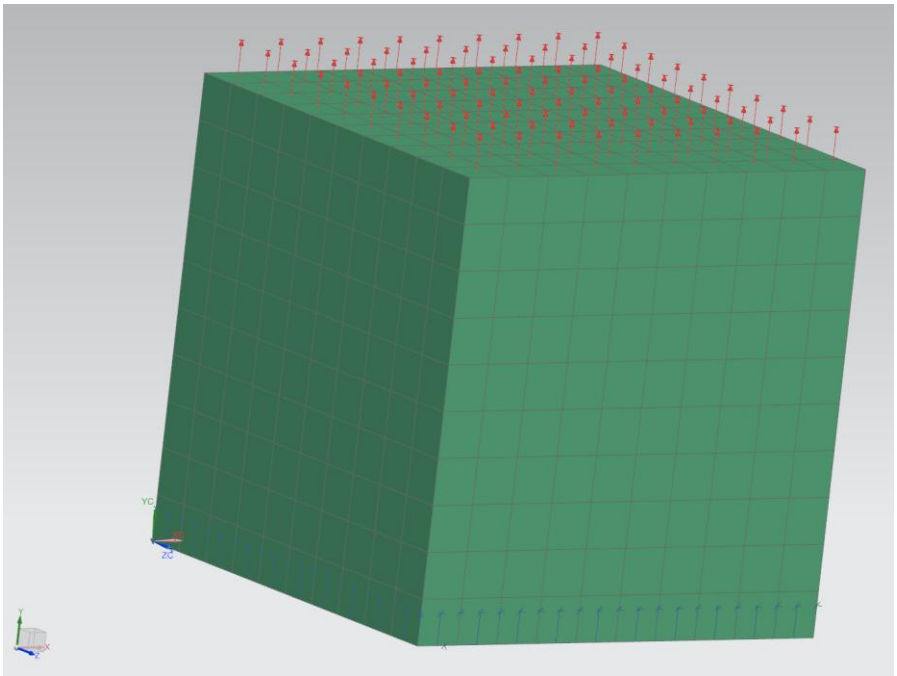


Figure 2.1 NASTRAN cube with brick element mesh and boundary conditions.

2.2 Step 2: Reading NASTRAN FE Model into FRANC3D

Once the NASTRAN FE model is created and the *.nas* (*bdf)* file is saved, we proceed with importing the FE model into FRANC3D. We use the *Nastran_Cube.nas* file that was created in the previous step.

You can choose to do either Step 2 or Step 3 (see Section 2.3).

Step 2.1: Importing Complete NASTRAN FE Model

Start with the FRANC3D graphical user interface (Fig 1.1) and select **File →Import,** Fig 2.2. In the dialog shown in Fig 2.3, choose Complete Model and select **Next**.

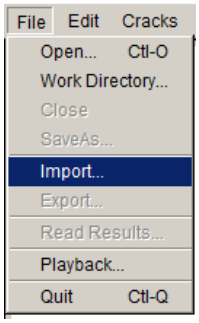


Figure 2.2 File Import menu option.

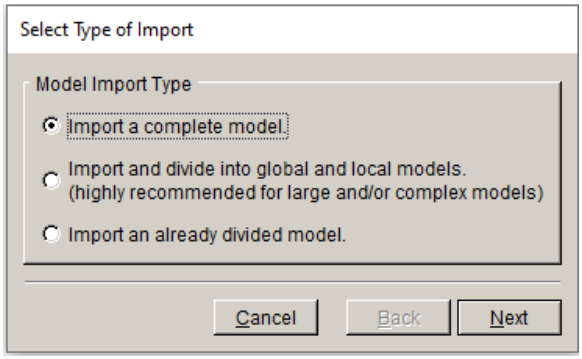


Figure 2.3 Model Import Type dialog

Switch the Mesh File Type radio button in the Select Import Mesh File window, Fig 2.4, to NASTRAN and select the file name for the model, *Nastran_Cube.nas*, and then select **Next**.

Note that the mesh file type can be set in the Preferences so that you do not have to change the mesh file type radio button if you always use NASTRAN.

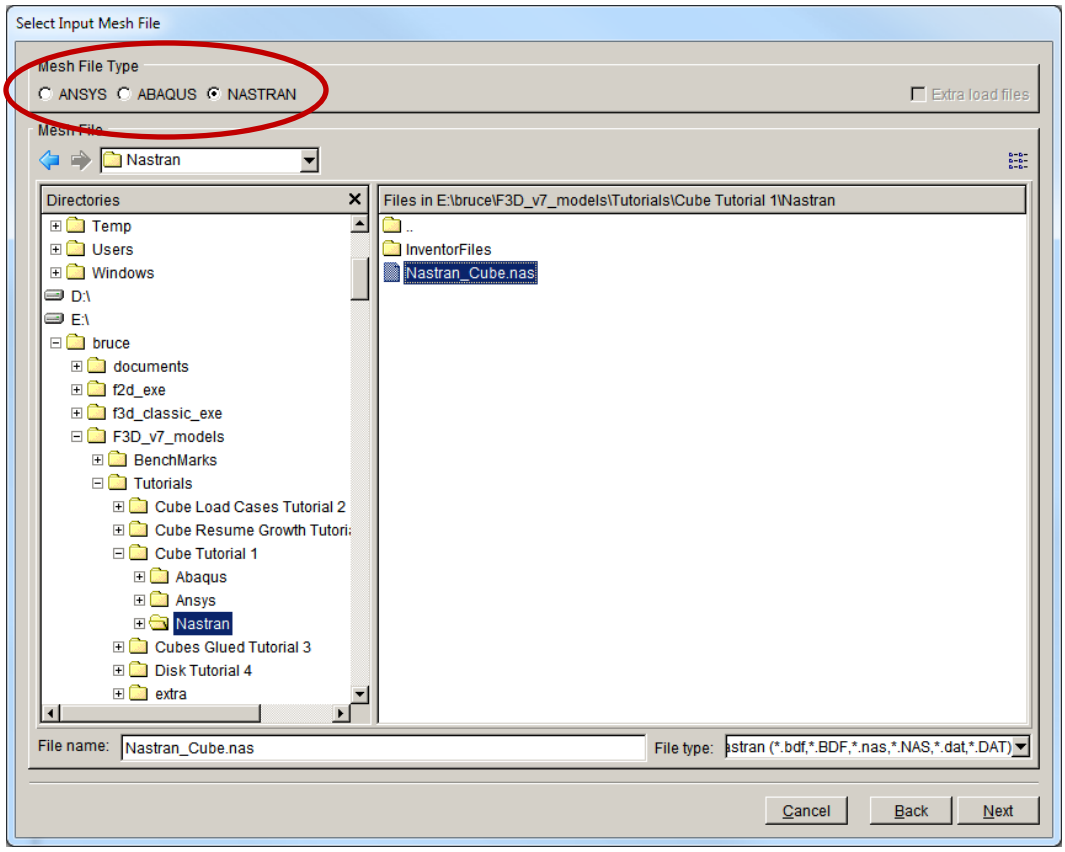


Figure 2.4 Select Input Mesh File dialog box.

Step 2.2: Select the Retained Items in the Local FE Model

The next panel, Fig 2.5, allows you to choose the mesh surface facets that are retained from the NASTRAN *.nas* file. Surfaces with boundary conditions are blue, Fig 2.6, and turn red when selected, Fig 2.7. You can rotate the model to view all surfaces. We retain the surfaces with boundary conditions (top and bottom of the cube) by choosing Select all. The boundary conditions are transferred automatically to the new mesh once the crack is inserted.

Select **Finish** to proceed. The Units dialog will be displayed; if you have already set the units using the **Edit** menu, just select **Finish**.

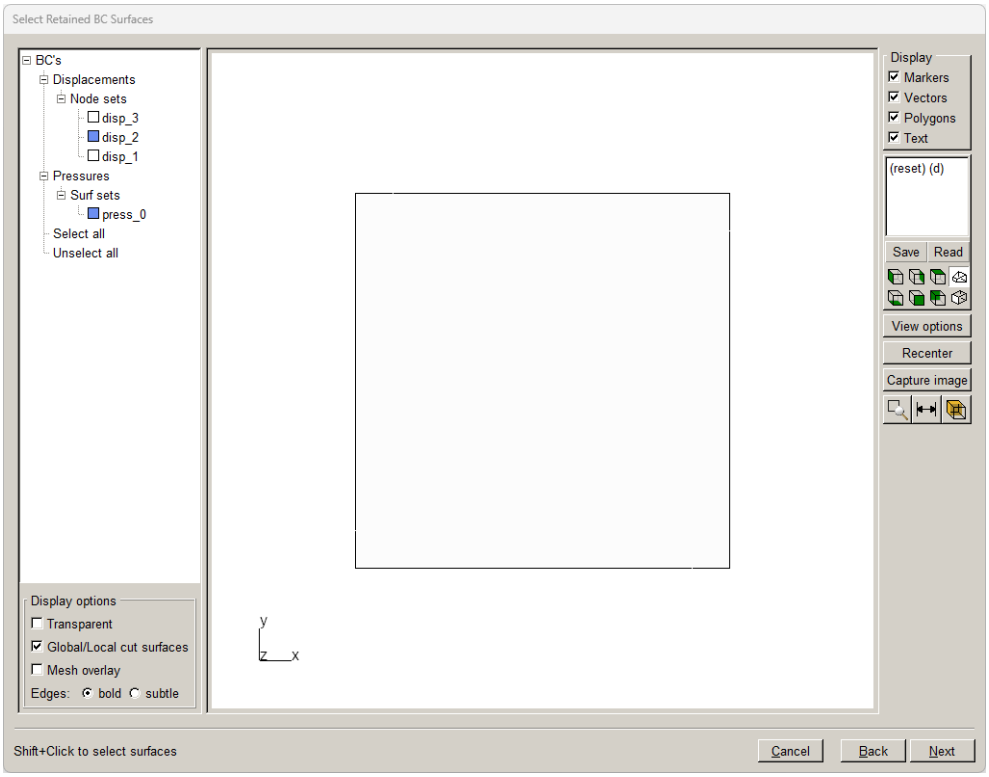


Figure 2.5 Select Retained BC Surfaces wizard panel.

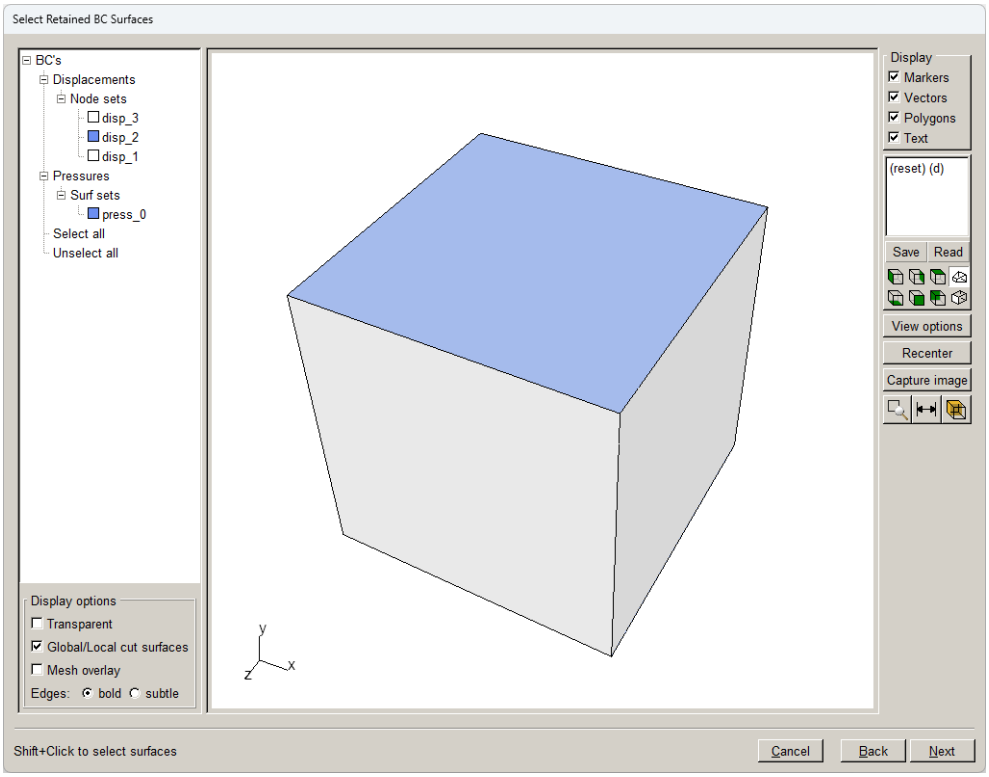


Figure 2.6 NASTRAN Model retain BC Surfaces wizard panel - unselected surface.

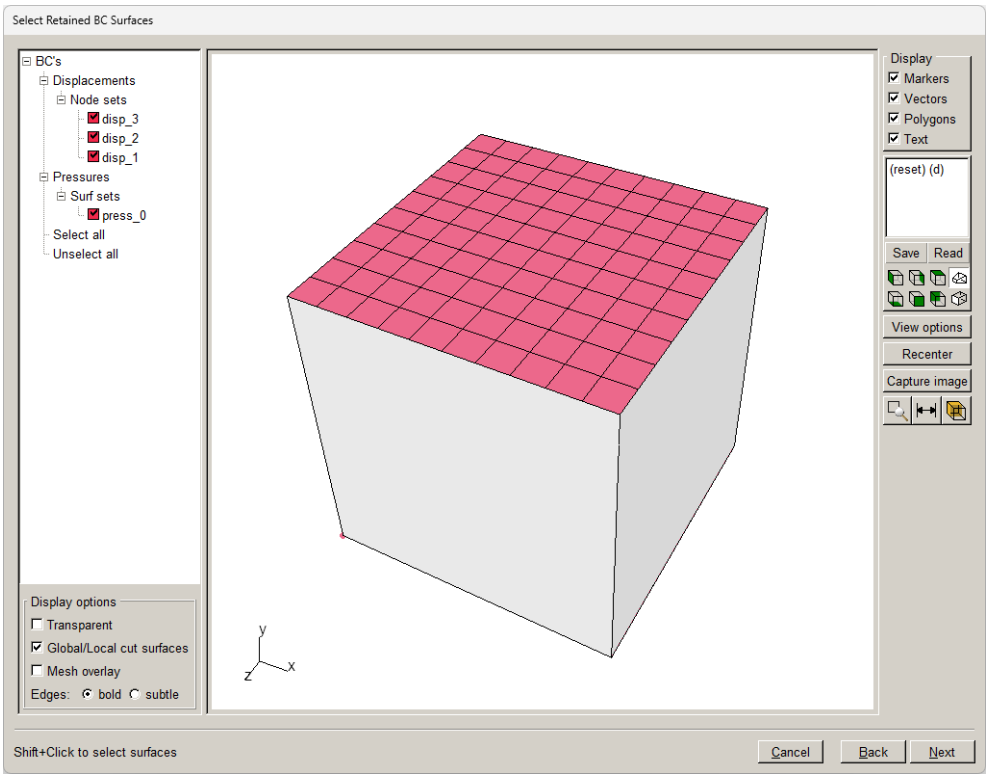


Figure 2.7 NASTRAN Model retain BC Surfaces wizard panel – selected surfaces.

Surface Mesh NOT Retained

The surface mesh facets do not have to be retained, and if a crack is inserted into a surface that has boundary conditions, then the surface mesh must not be retained. In such a case, the boundary condition data is mapped to the remeshed surface. In practice, transfer of boundary condition data is simpler and more precise if the surface mesh is retained, but sometimes this is not possible. FRANC3D will automatically transfer the boundary condition data to the new mesh in either case.

The model is displayed in the FRANC3D modeling window, Fig 2.8. You can turn on the surface mesh, or manipulate the model view by rotating, *etc*. Fig 2.8 shows the retained mesh on the top surface (retained bottom surface is not visible) where the boundary conditions are applied. See Section 2 of the Reference document for a description of the view manipulation.

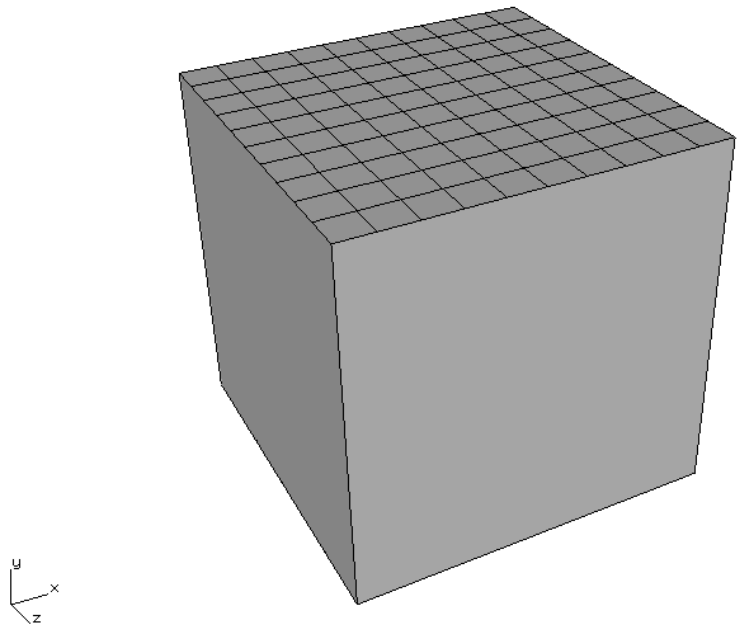


Figure 2.8 NASTRAN model imported into FRANC3D with retained facets on top of the cube.

If you have not set a working directory (using the **File →Set Work Directory** menu option), FRANC3D might present the dialog shown in Fig 2.9 prior to displaying the model. Select **Yes** to set the directory as the folder where the NASTRAN *.nas* file resides. Many files can be created during a crack growth simulation, and it is best to keep them together in one folder.

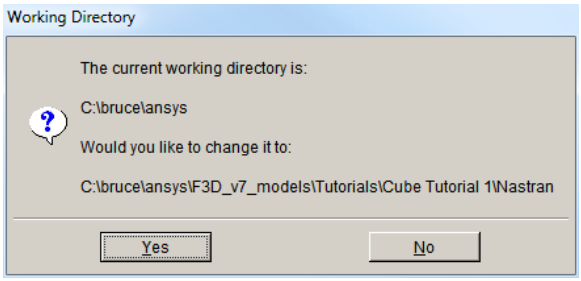


Figure 2.9 Set Working Directory dialog.

2.3 Step 3: Importing and Subdividing the Model

If you did Step 2, close the model using the **File →Close** menu option before starting this step.

The NASTRAN model can be split into smaller parts before inserting the crack. Go to **File →Import** and choose the Import and divide into global and local models radio button, Fig 2.10, and choose the *Nastran_Cube.nas* model, Fig 2.11. Remember to set the Mesh File Type radio button to NASTRAN.

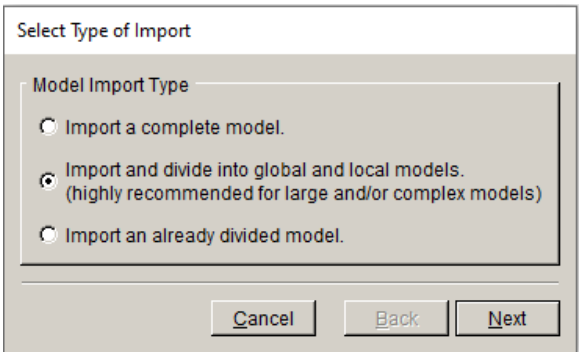


Figure 2.10 Model Import Type dialog with Import and divide selected.

Instead of the retain BC surfaces dialog (see Fig 2.5), the Define a Local Submodel window, Fig 2.12, appears. Figs 2.13 and 2.14 show two of the submodel selection tools. Using the options on the left, a submodel is selected; elements to be retained in the submodel appear red, and the selection is confirmed using the **Crop** button. See Section 4.5 of the Reference document for a more complete description of the submodeling tools.

For this tutorial, the rectangular Rubberband Box is used to create a smaller portion of the cube, Fig 2.15, which has an exposed face for crack insertion.

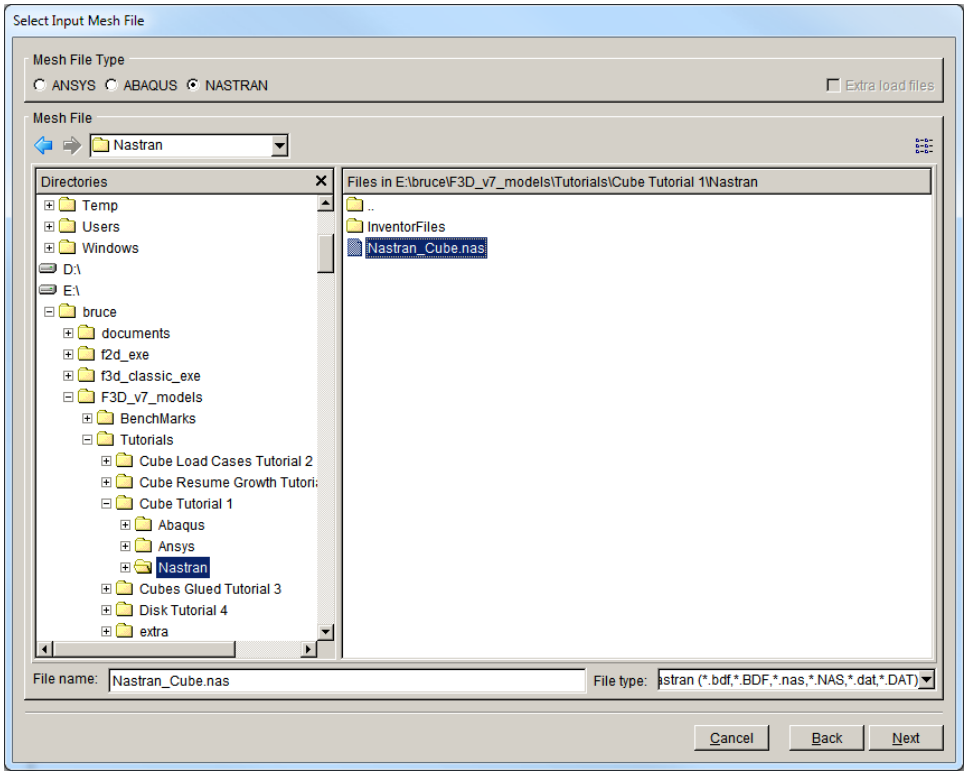


Figure 2.11 Select Input Mesh File selection.

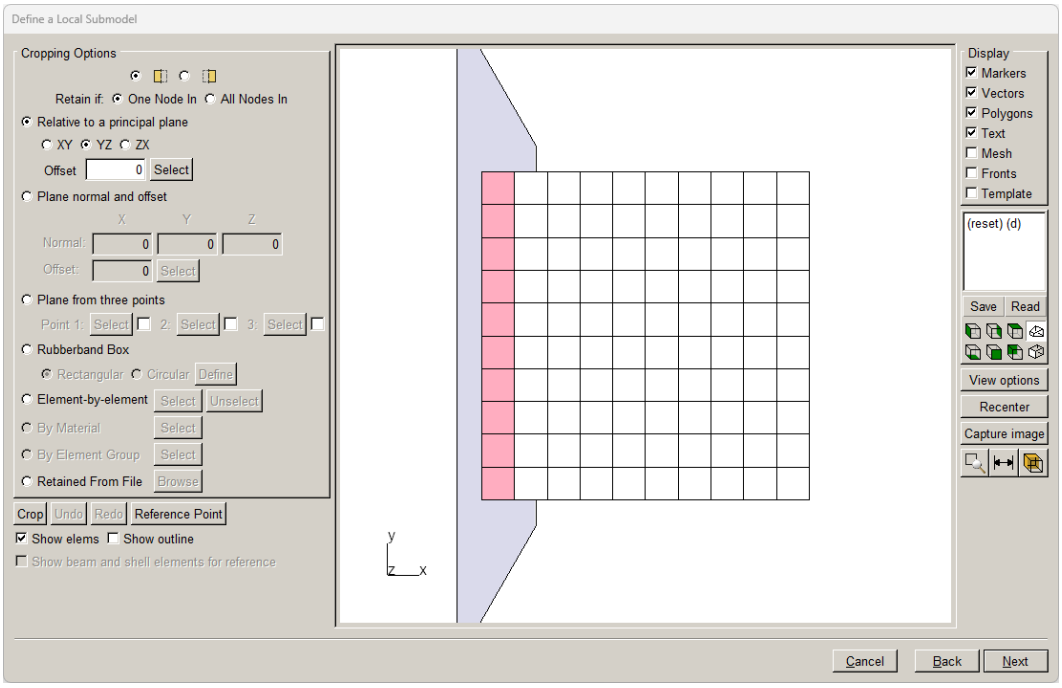


Figure 2.12 Define a Local Submodel Window

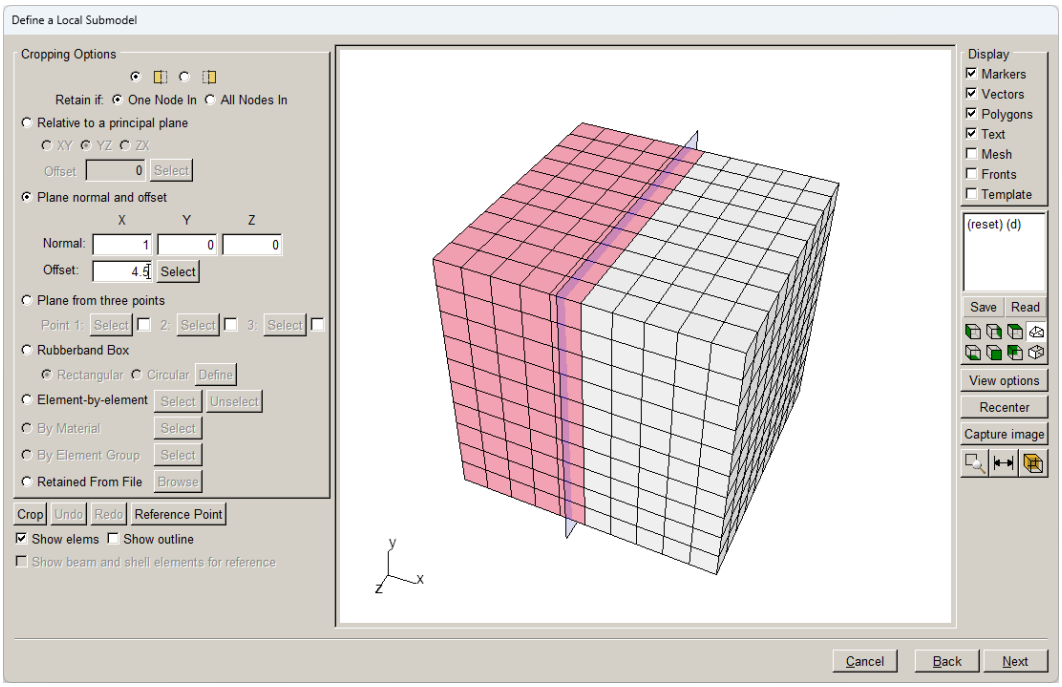


Figure 2.13 Plane Cutting tool – **Crop** will retain the red elements.

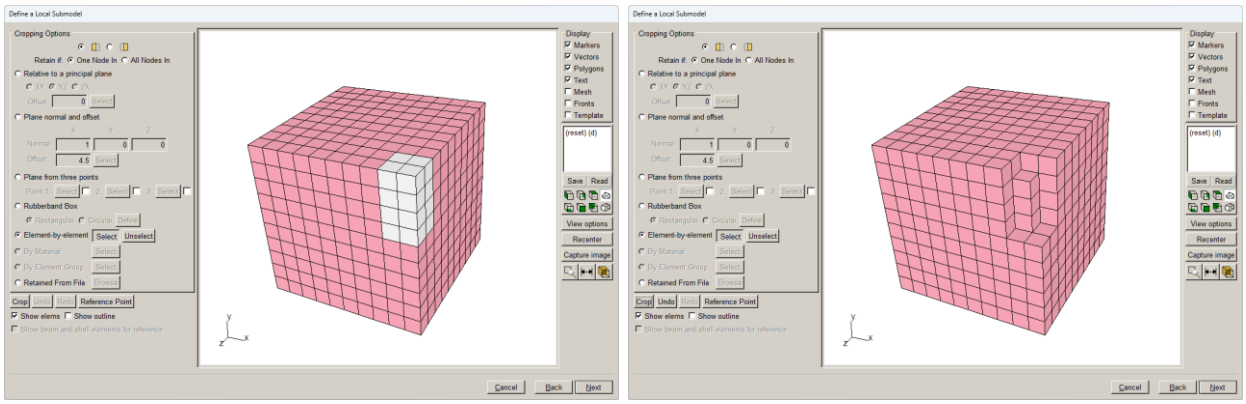


Figure 2.14 Element-by-Element cropping by selecting elements to remove.

We are inserting a surface crack normal to the y-axis (loading direction) and located on the +z surface, and the selection in Fig 2.15 is designed for this. Use the Rectangular option of the Rubberband Box to select and **Crop** a small portion of the model.

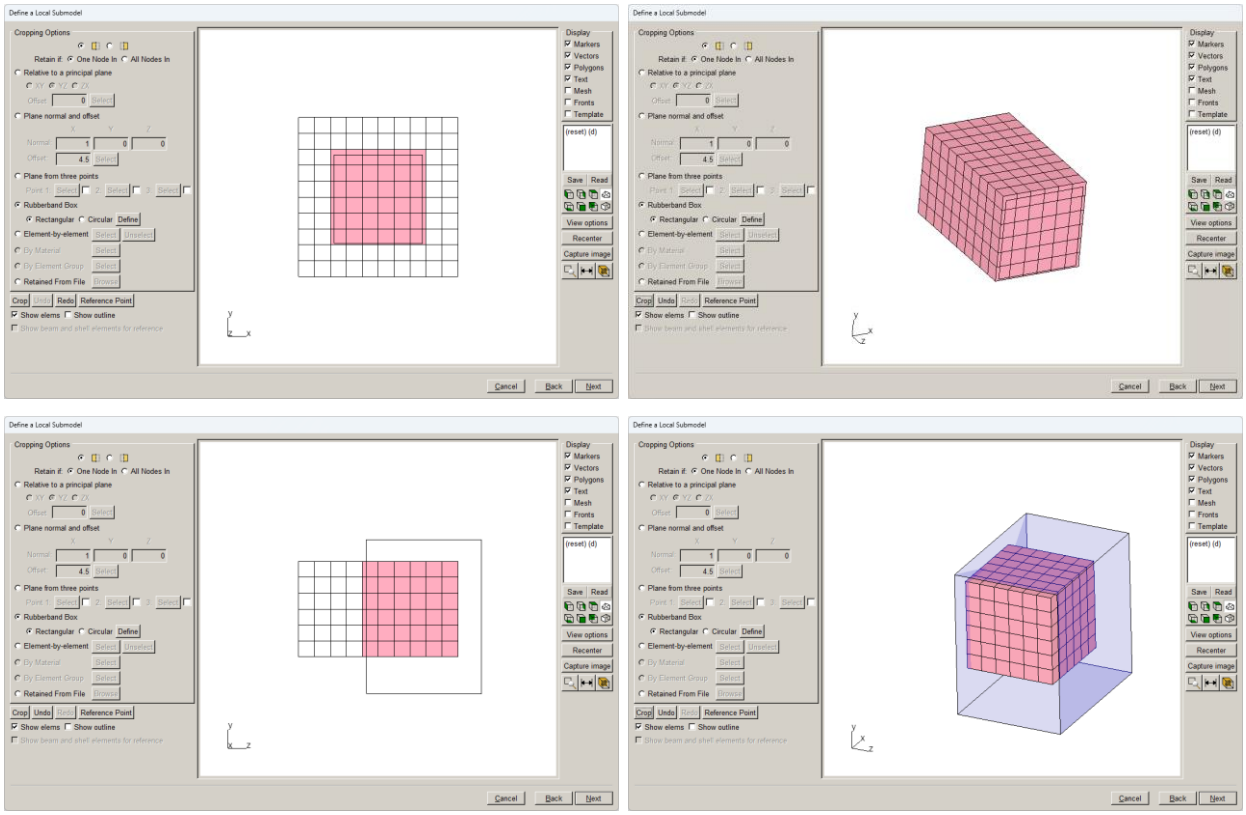


Figure 2.15 Rubberband Box tool used to select a submodel.

Once the elements have been selected and cropped, select **Next**. The Save Files dialog appears, Fig 2.16. Choose the names of the local and global models and their location; we use the default file names and the working directory set earlier. Select **Next**.

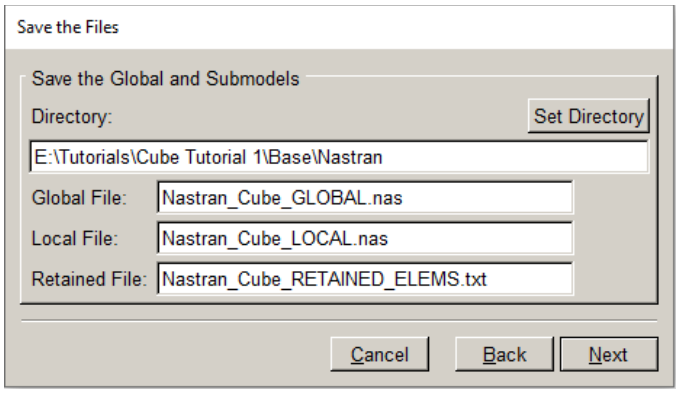


Figure 2.16 Local/Global model save window.

The Select Retained BC Surfaces dialog is displayed next, Fig 2.17, and shows the retained cut-surface. The cropped selection in Fig 2.15 avoids all surfaces with boundary conditions. Click **Next** to proceed.

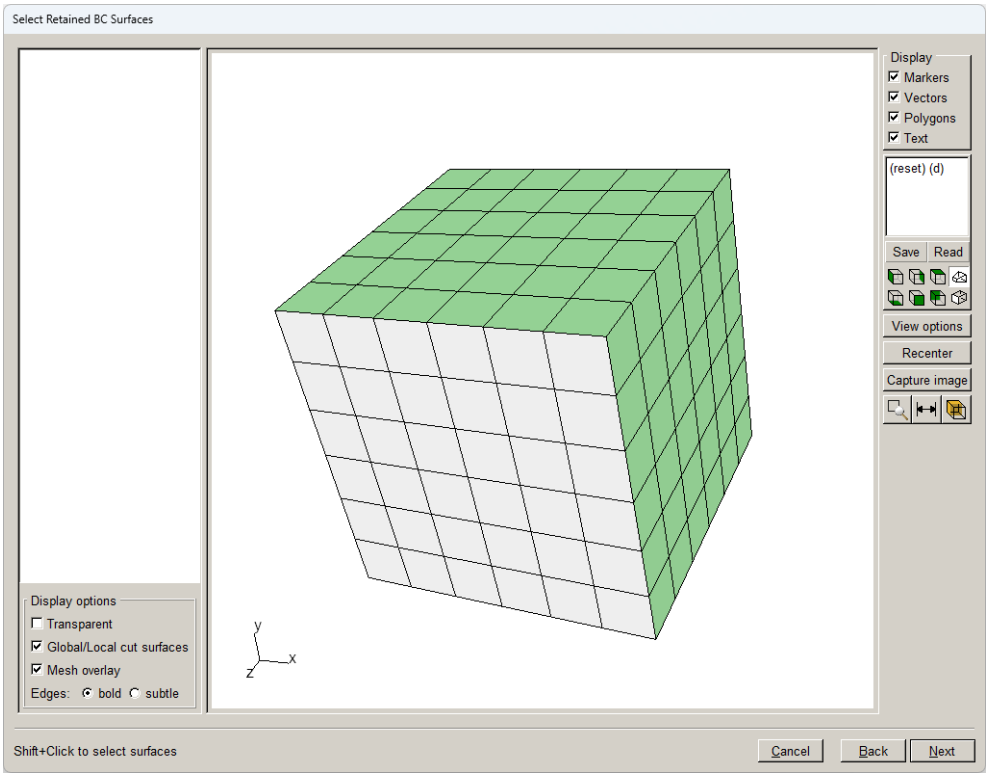


Figure 2.17 Select Retained BC Surfaces wizard panel – showing the retained cut-surface.

The model units, Fig 2.18, must be set. As defined in Step 1, the units for length are mm and the units for stress are MPa. The temperature and time units are not needed for this tutorial.

Select **Finish** and the local model is displayed in the main FRANC3D window, Fig 2.19. The cut-surface mesh facets are retained, while the front surface where the crack will be inserted does not have retained facets.

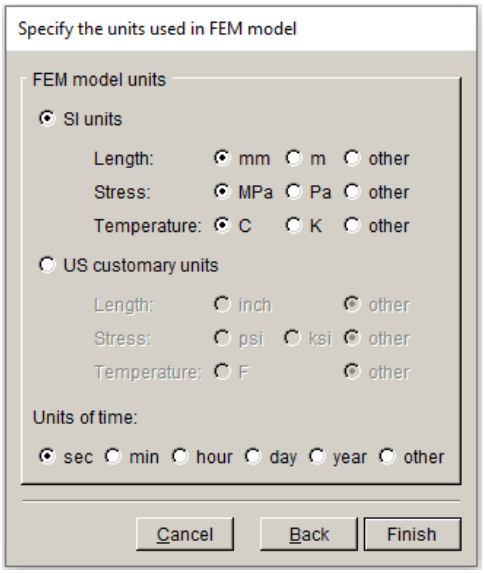


Figure 2.18 FE Model units

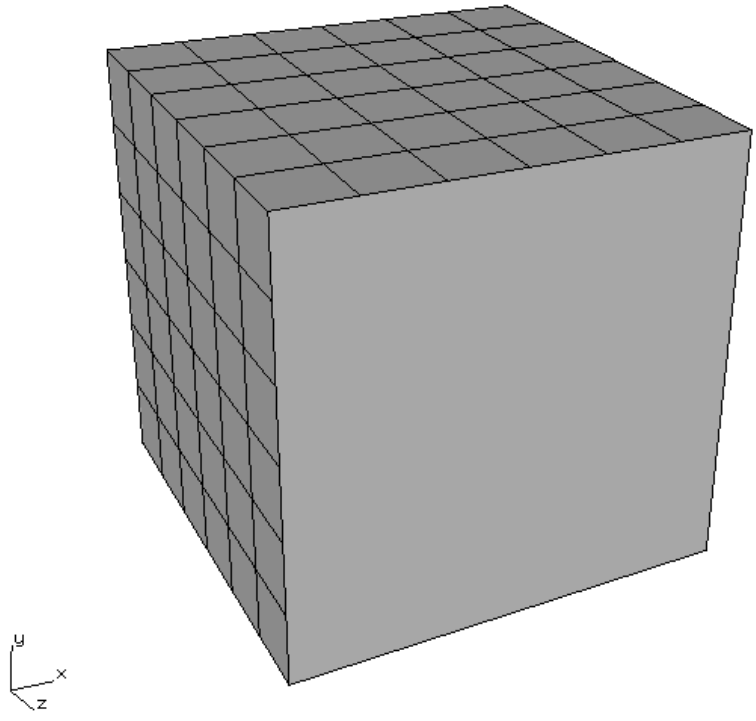


Figure 2.19 Local model with retained mesh facets on cut-surface

2.4 Step 4: Insert a Crack

We now insert a half-penny surface crack into the model. The local submodel from Section 2.3 is used, as opposed to the full model from Section 2.2. In Step 4.1, we describe how to define a new crack, and in Step 4.2, we describe how to insert a flaw from a file.

Note: if you want to try both types of crack insertion, you will have to re-import the uncracked model to insert a crack from a file; inserting a crack from a file will be used in subsequent steps and other tutorials.

Step 4.1: Define a new Crack from FRANC3D Menu

From the FRANC3D menu, select **Cracks →New Flaw Wizard,** Fig 2.20. The first panel of the wizard, Fig 2.21, allows you to select the flaw type; a crack with zero volume is the default. We select the Save to file and add flaw radio button so that the *.crk* file is saved for subsequent tutorials. Note that the Symmetry surface crack option is only displayed if symmetry cracks are turned on in the Preferences (see Section 5.2 in the Reference).

Select **Next**. The second panel of the wizard, Fig 2.22, lets you choose the crack shape; the default is the ellipse, which is what we want in this case. Select **Next**. The crack is a circle with radius=1, centered on the cube's front (+z) face, and parallel to the xz-plane (normal to y).

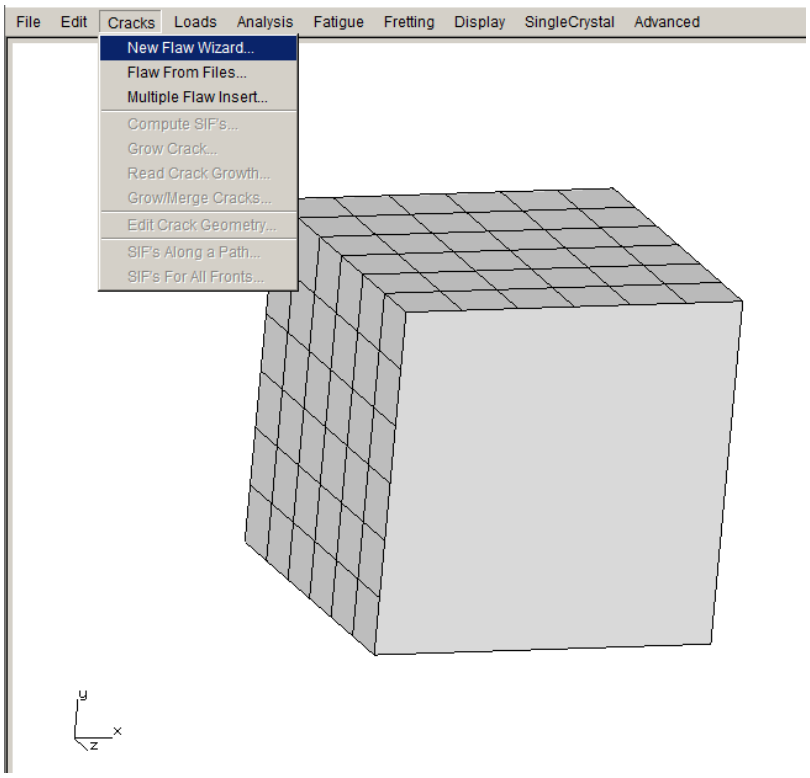


Figure 2.20 New Flaw Wizard menu item selected.

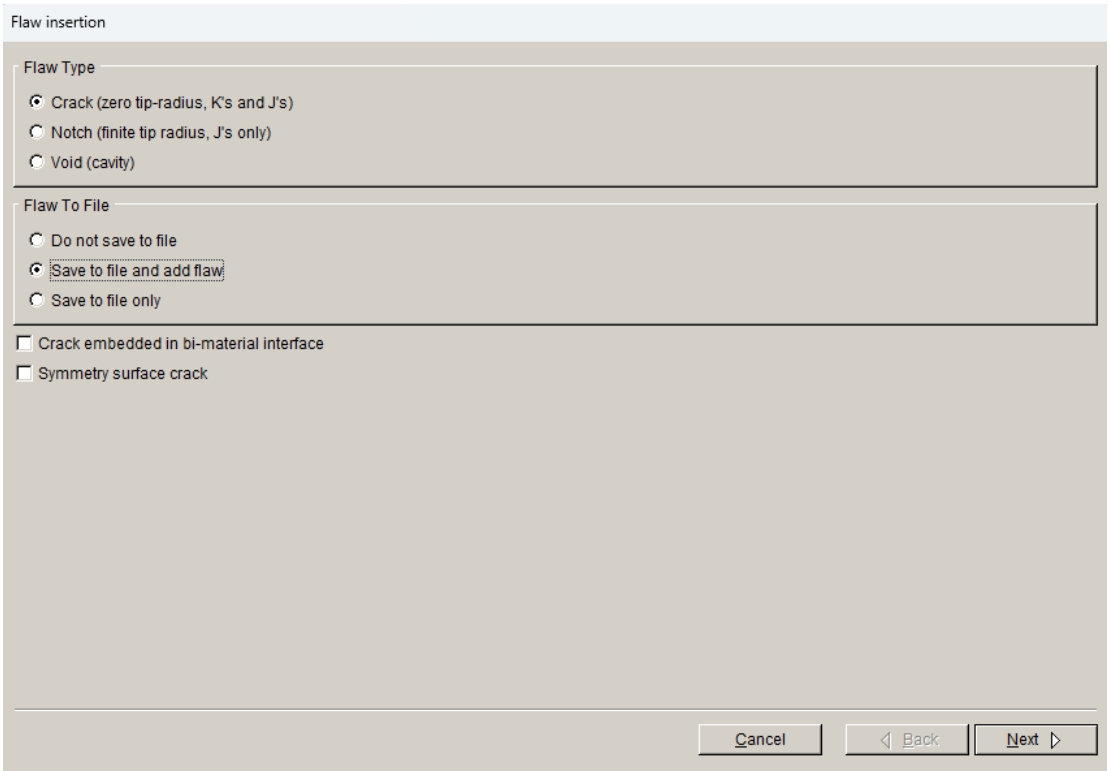


Figure 2.21 New Flaw Wizard first panel – choose Crack (zero volume flaw).

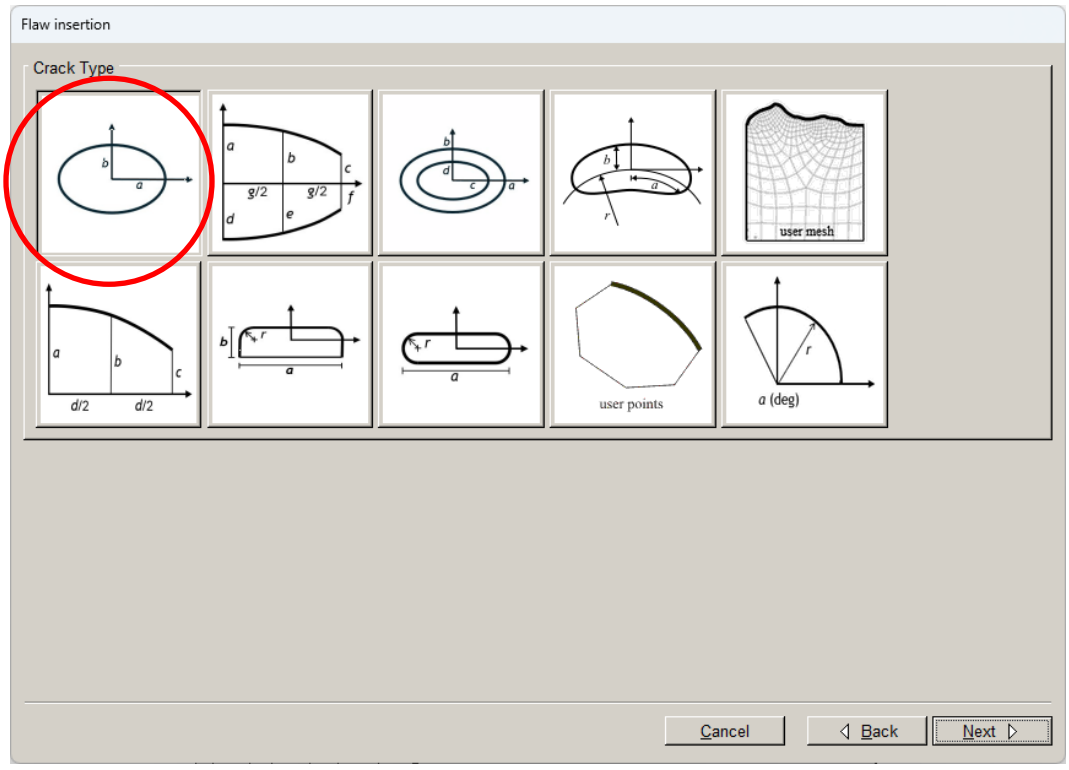


Figure 2.22 New Flaw Wizard second panel – choose ellipse shape.

The third wizard panel lets you set the dimensions for the ellipse, Fig 2.23; set a=1 and b=1. Select **Next**.

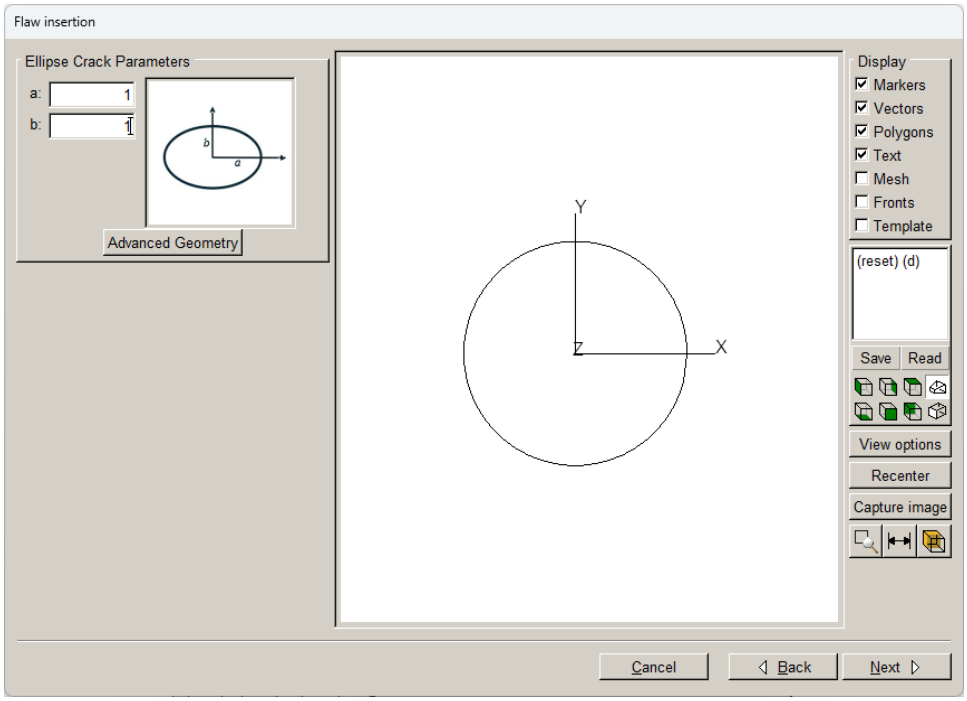


Figure 2.23 New Flaw Wizard third panel – set ellipse dimensions.

The fourth panel lets you set the crack location and orientation, Fig 2.24. We place the crack at the center of the front face at coordinates (5,5,10) and rotate the crack 90 degrees about the Global x-axis. Select **Next**.

The final panel allows you to set the crack front mesh template parameters, Fig 2.25; use the defaults. In practice, the default template radius is based on the crack dimensions and might need to be adjusted depending on the model and crack. Select **Finish**.

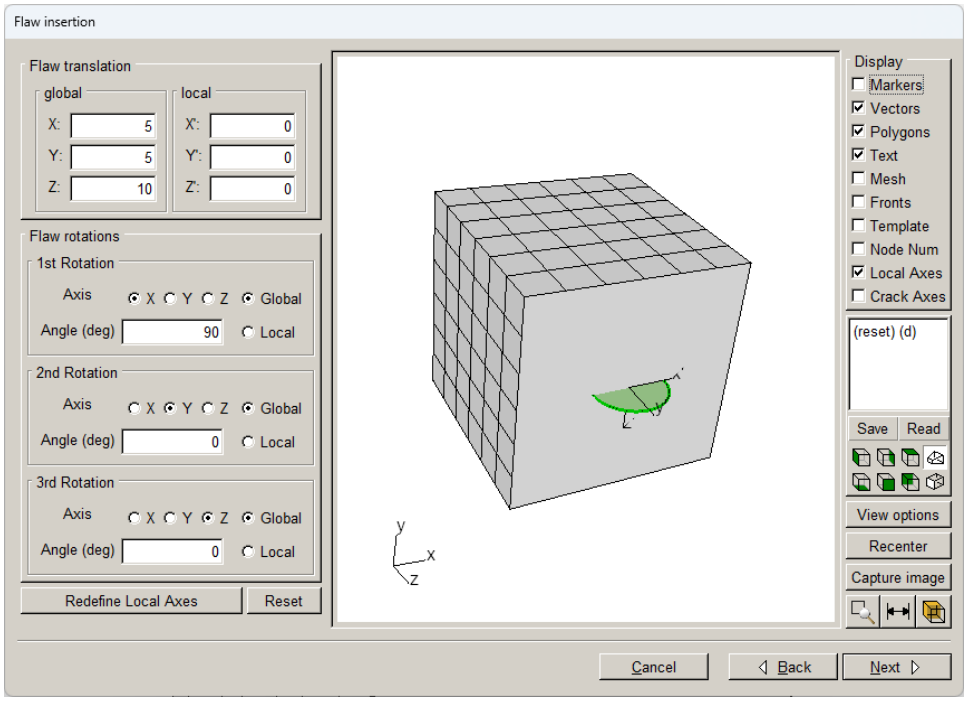


Figure 2.24 New Flaw Wizard fourth panel – set crack location and orientation.

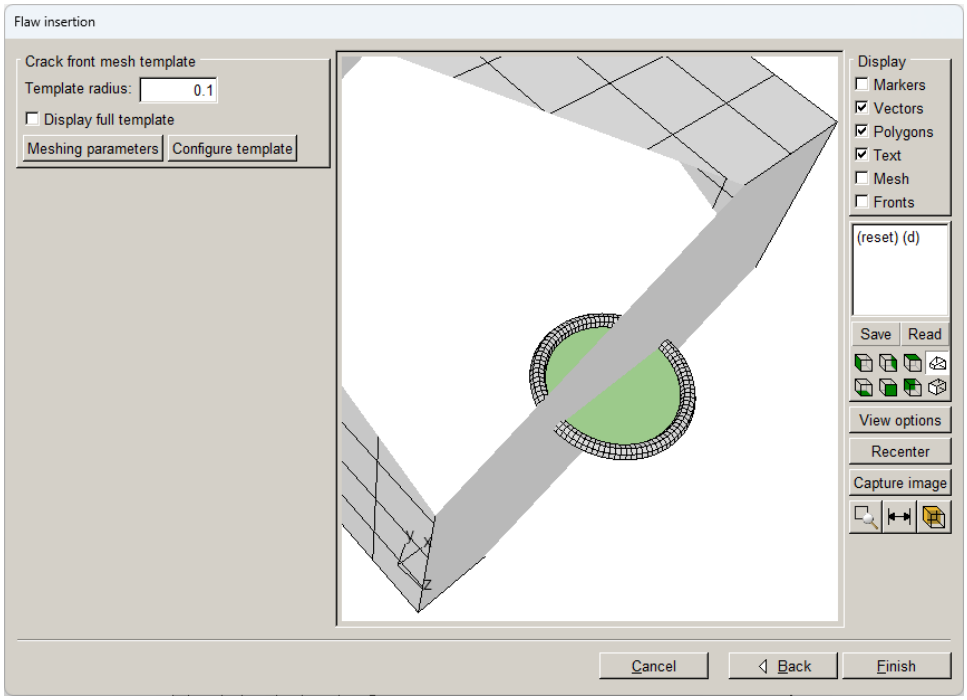


Figure 2.25 New Flaw Wizard final panel – set template mesh parameters.

At this point, you will be asked to specify the file name to save the crack information, Fig 2.26; we call it *Cube_Crack.crk*. Select **Accept**.

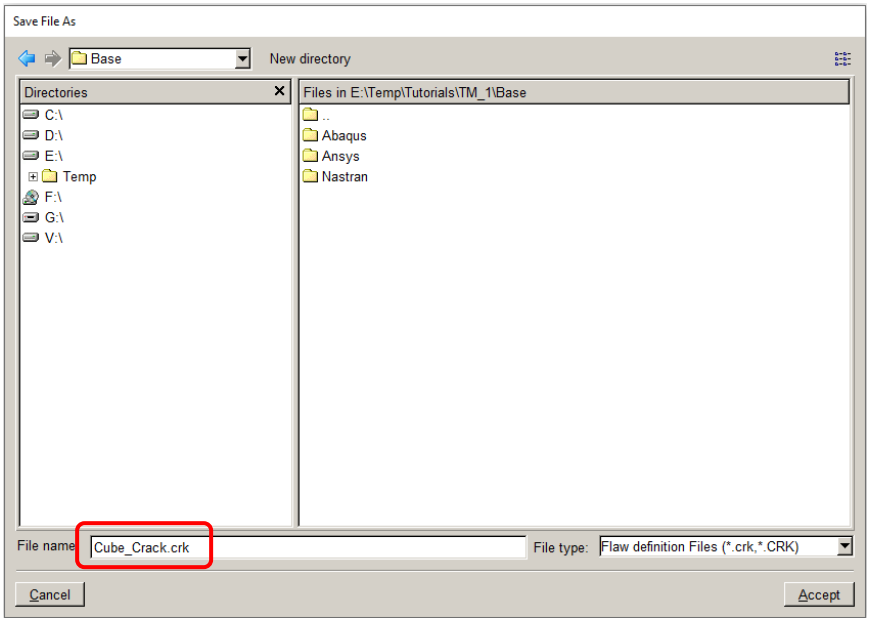


Figure 2.26 New Flaw Wizard final panel – save file as.

The crack is inserted into the model and then remeshing occurs. A Flaw Insertion Status dialog is displayed during this process, Fig 2.27. There are three stages: geometric intersection of the crack with the model, surface meshing and volume meshing. The resulting model is displayed in the main FRANC3D window, Fig 2.28; the surface mesh is turned on in this image.

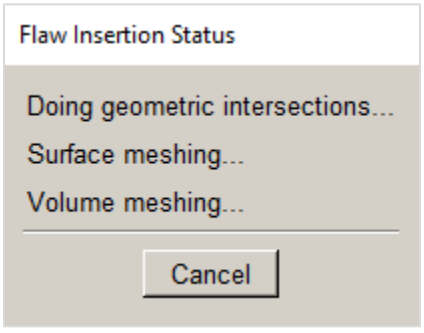


Figure 2.27 Flaw Insertion Status window.

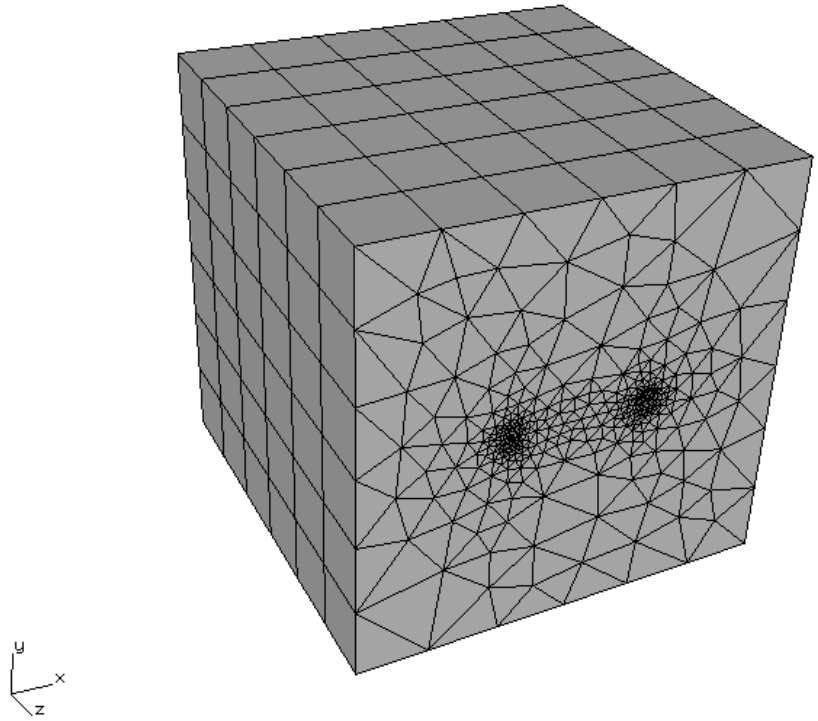


Figure 2.28 Remeshed cracked local model with surface mesh turned on.

Step 4.2: Insert Cracks from Files

If you want to try this step, you must **Close** the model from Step 4.1 and re-import the original uncracked (local) model.

From the FRANC3D menu, select **Cracks →Flaw From Files,** Fig 2.29. The first panel, Fig 2.30, lets you select the *.crk* file. Choose the *Cube_Crack.crk* file name and select **Accept**. This crack was created using the New Flaw Wizard with the save to file option in Step 4.1. The flaw being added is a circle with radius=1, centered on the cube's front face, and parallel to the xz-plane.

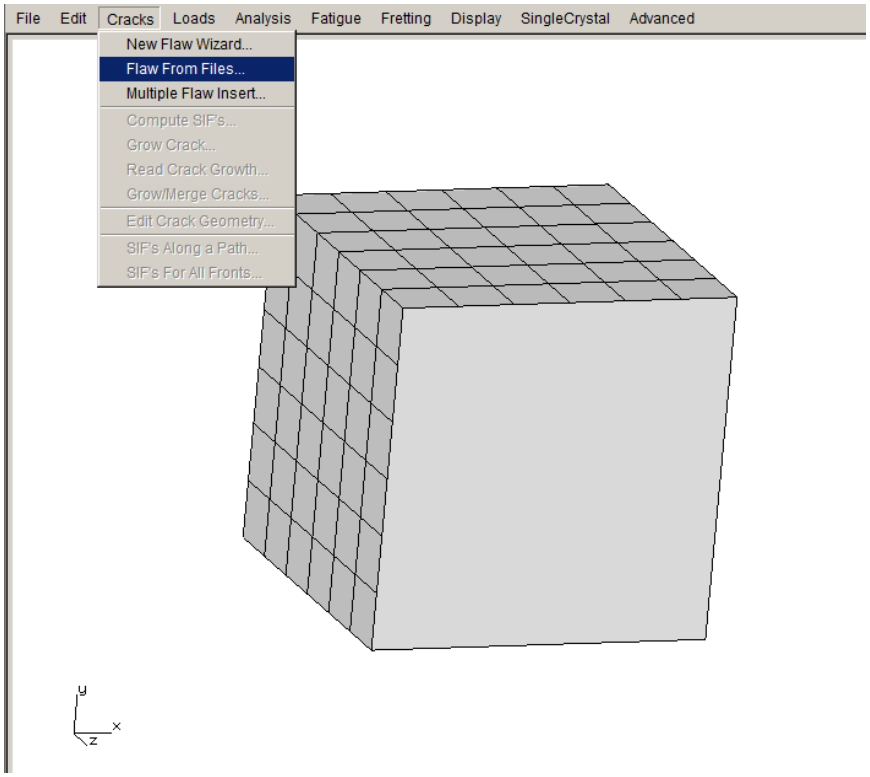


Figure 2.29 Flaw From Files option in Crack menu.

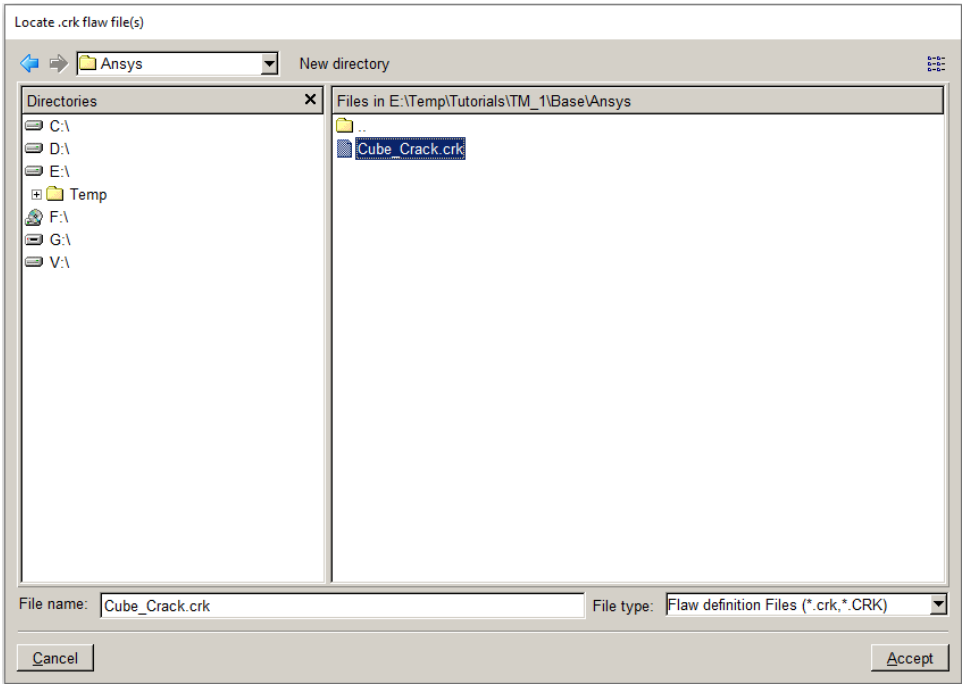


Figure 2.30 Flaw from file dialog to select .crk file.

The next panel of the wizard, Fig 2.31, allows you to adjust the location of the flaw. The orientation cannot be changed. The final panel, Fig 2.32, allows you to set the template mesh parameters. Use the default settings and select **Finish** to complete the insertion and remeshing.

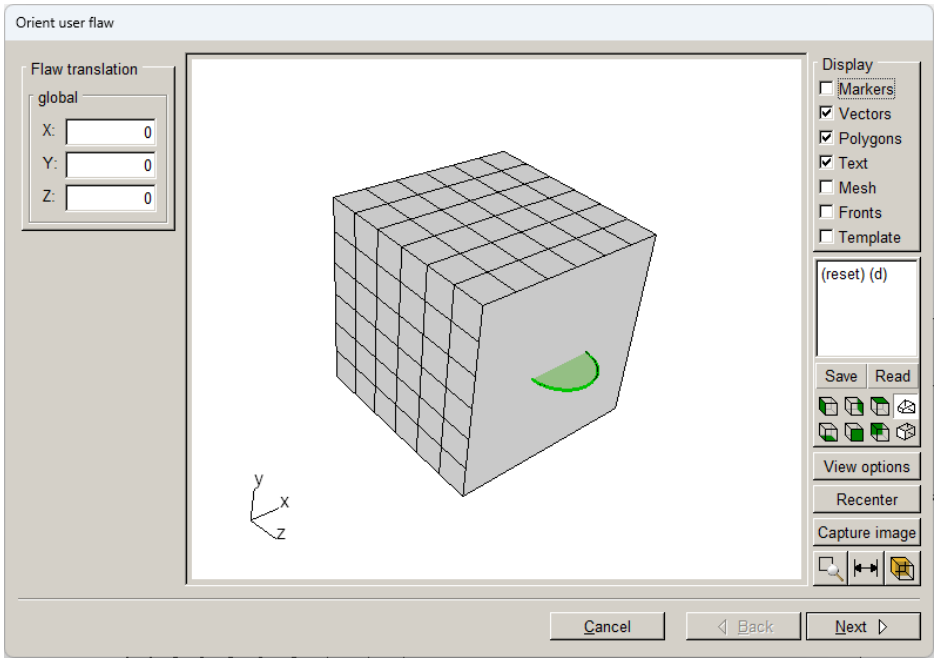


Figure 2.31 Flaw wizard panel to set location.

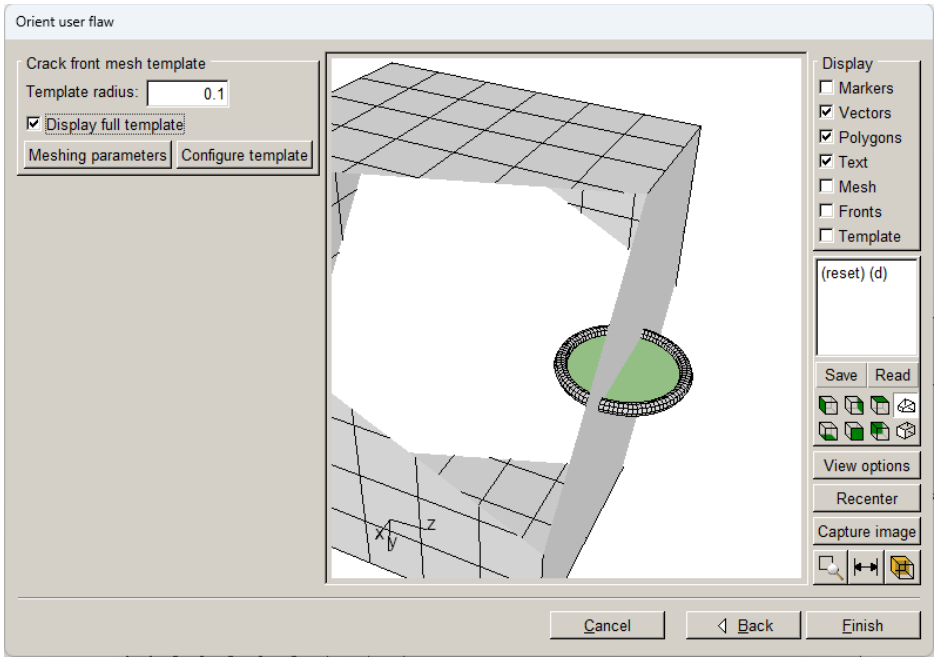


Figure 2.32 Flaw geometry shown in the model with crack template mesh.

2.5 Step 5: Static Crack Analysis

Once the crack is inserted and the model is remeshed, we perform a stress analysis using NASTRAN, which will provide the displacement results that are needed to compute stress intensity factors. Typically, you would run a "static crack analysis" of the initial crack prior to running automated crack growth; this allows you to verify that the crack model is consistent with the uncracked model. You can compare displacement contours using the NASTRAN post-processor, for the original uncracked and cracked models to verify that the crack model is valid.

You might also insert the crack with a bigger or smaller template radius to verify that the computed SIFs are accurate; there should be no notable change in SIFs for a reasonable template.

Step 5.1: Select Static Crack Analysis

From the FRANC3D menu, select **Analysis →Static Crack Analysis**, Fig 2.33. The first panel, Fig 2.34, requests that you enter the file name for the FRANC3D data. We call it *Nastran_Cube_Subdivide.fdb*; select **Next** once you enter a File Name.

Note that you cannot use the initial uncracked *Nastran_Cube.nas* or *Nastran_Cube_LOCAL.nas*, as a new *.nas* (or *.bdf)* will be created during the analysis. The original uncracked *.nas* files are reused for each step of crack growth.

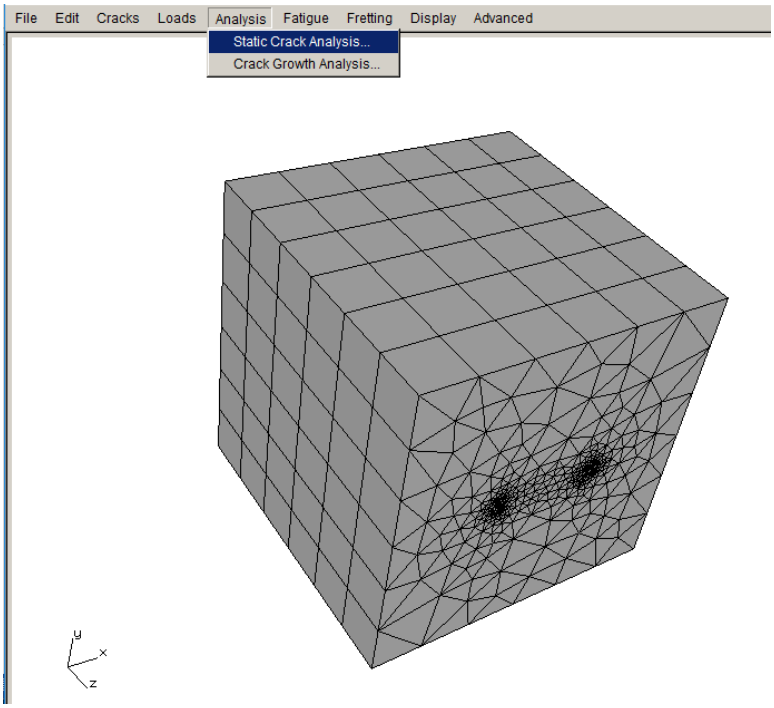


Figure 2.33 Static Crack Analysis menu

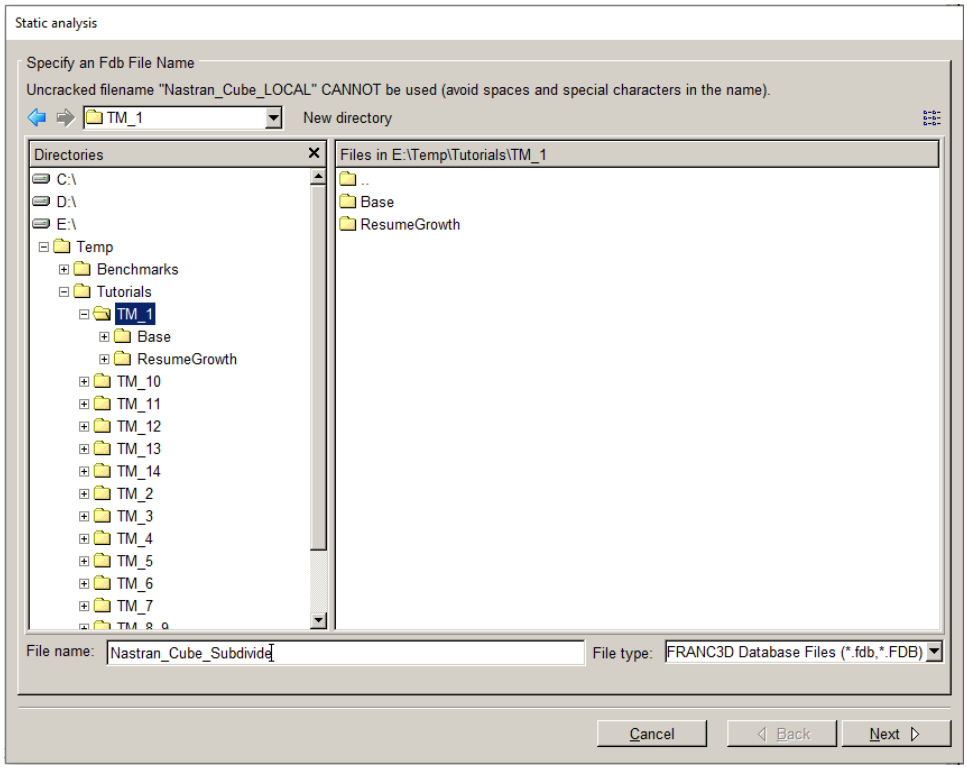


Figure 2.34 Static Analysis wizard first panel – File Name.

Step 5.2: Select NASTRAN Analysis Options

The next panel of the wizard, Fig 2.35, allows you to specify the NASTRAN output and analysis options. The model is a sub-model and must be connected to the global model, so the Connect to global model box is checked automatically if the FRANC3D submodeler tool was used. The next panel, Fig 2.36, allows you to set the local + global model connection. If the model being analyzed is a full/complete model, the panel in Fig 2.35 is the final panel as there is no global model, and the **Next** button will be a **Finish** button.

The panel in Fig 2.36 shows the options for tying together the local and global models. AUTO_CUT_SURF and GLOBAL_CONNECT_SURF are the sets/surfaces created by the FRANC3D submodeling tool and should be automatically selected. Click **Finish** to start NASTRAN running (in batch/background mode). FRANC3D writes files and then attempts to execute NASTRAN based on the NASTRAN Executable information (Fig 2.35). You can monitor the FRANC3D CMD window and the NASTRAN *.f06* and *.log* files for errors or messages.

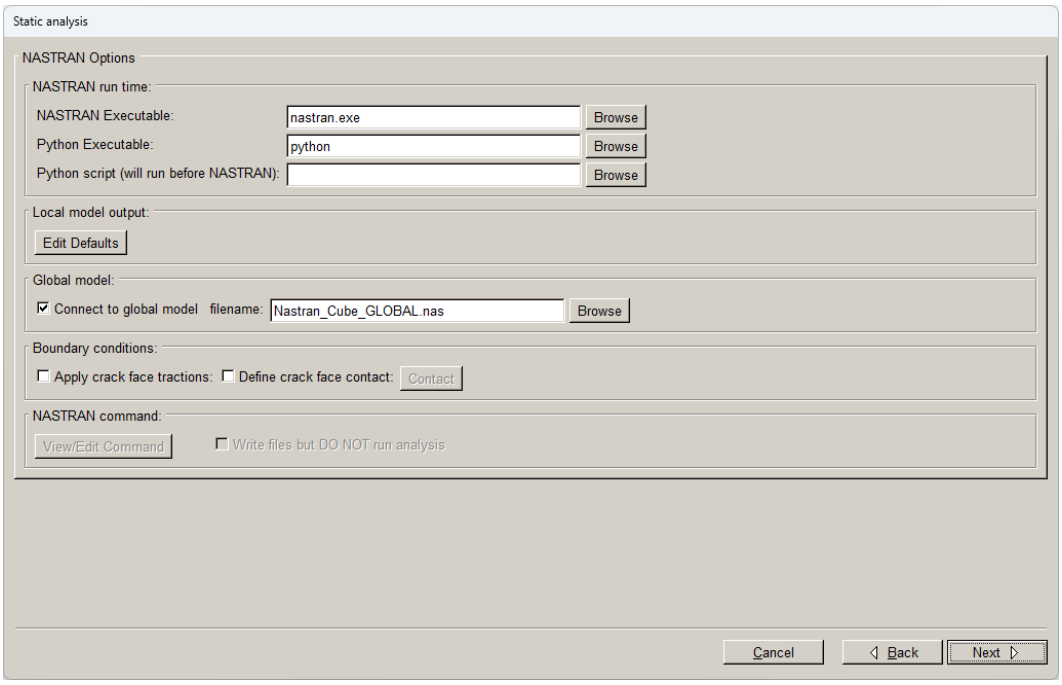


Figure 2.35 Static Analysis wizard third panel – NASTRAN output options.

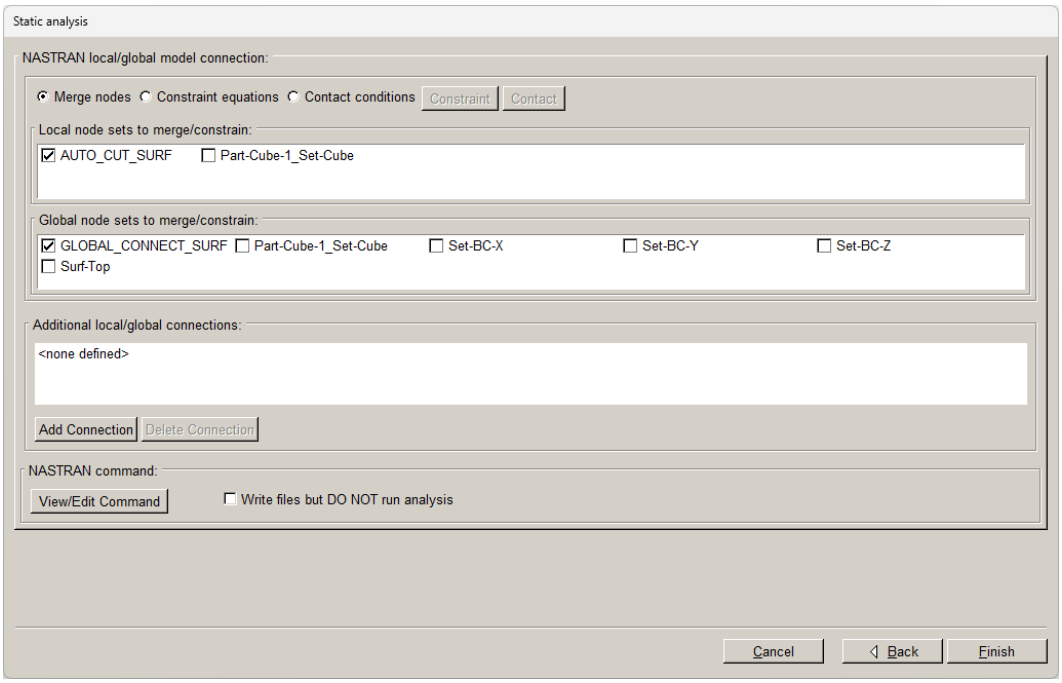


Figure 2.36 Local/Global model connection

Note that if NASTRAN does not run, you will see a message in FRANC3D indicating that SIFs cannot be computed because there are no displacements, or you might see a message indicating that FRANC3D cannot find or open the *.dtp* file. You can review the NASTRAN files for errors. The NASTRAN executable (and path to it) will depend on the installation and operating system. For MS Windows, for a typical installation, the executable is simply "nastran.exe" (not nastranw.exe") and is in the folder "C:\Program Files\Siemens\SimcenterNastran_2022\bin". For Linux, the extra command line flag "batch=no" is required to run the analysis in the foreground.

Choosing Write files but DO NOT run analysis in Fig 2.36 will create all the files without running NASTRAN, if the analysis needs to be run later or on a different computer.

If you need to run analyses on a different computer, you must bring the results file (*.pch*) back to the current folder, and then you can use the **File** – **Read Results** menu option to import the results into FRANC3D.

2.6 Step 6: Compute Stress Intensity Factors

Step 6.1: Select Compute SIFs

When the NASTRAN analysis is done (and the *.pch* file read back into FRANC3D), we can compute SIFs. From the FRANC3D menu, select **Cracks →Compute SIFs**, Fig 2.37. The Stress Intensity Factor dialog is displayed, Fig 2.38. You should use the M-integral, but you can check that the Displacement Correlation results are similar. Select **Accept** and the SIFs Plot window is displayed, Fig 2.40. You can view the three modes of stress intensity factor (SIFs) modes using the upper tabs: KI, KII and KIII; mode II and III values should be negligible compared to mode I.

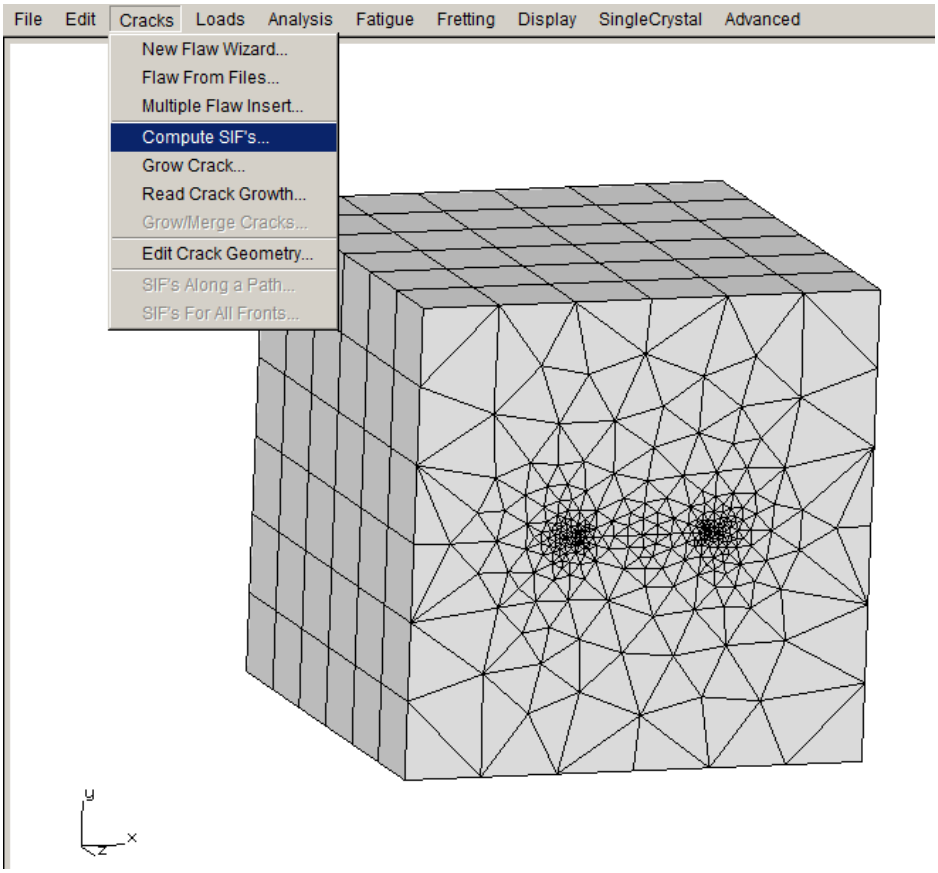


Figure 2.37 Compute SIFs selected from the Cracks menu.

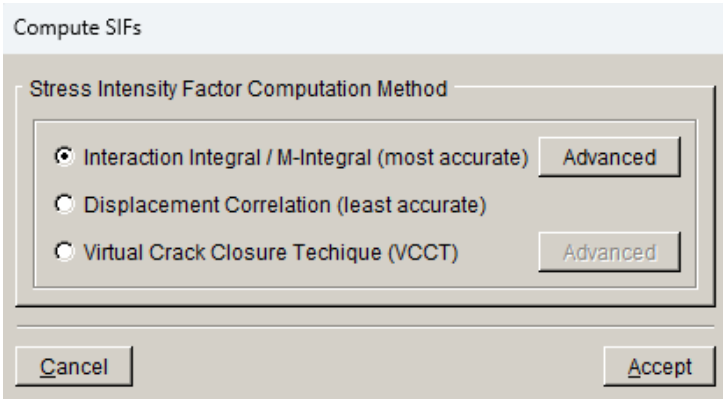


Figure 2.38 Compute SIFs dialog

The units for SIFs are MPa√mm. The SIFs are plotted versus the normalized crack front length, starting from point A – see the image in the left pane of Fig 2.39. SIFs are computed at element midside nodes along the front, so the values do not start at exactly zero.

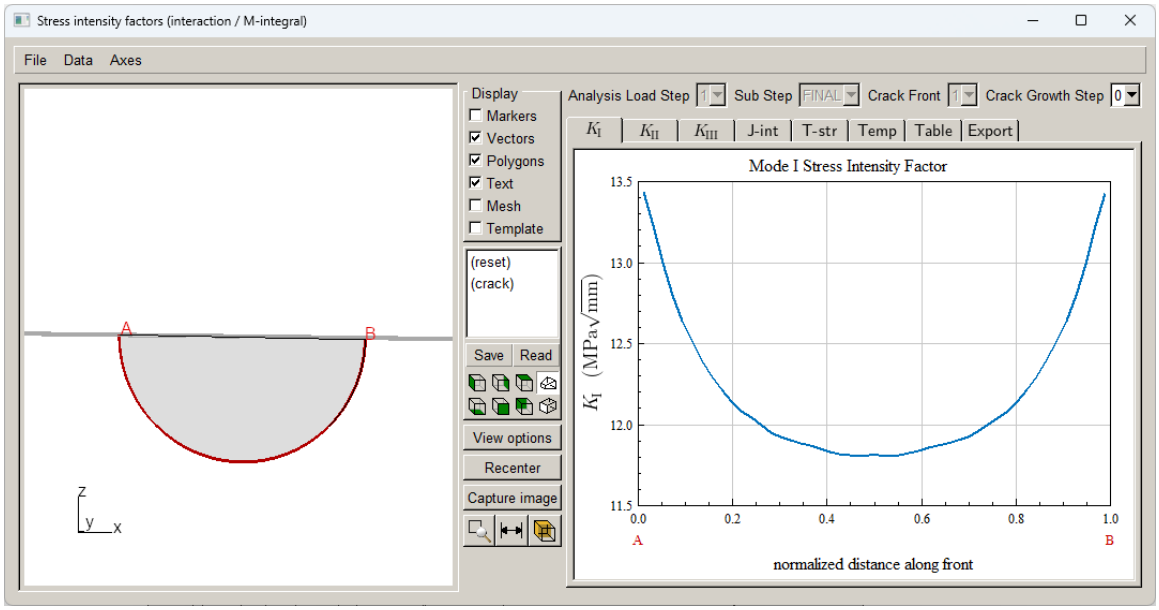


Figure 2.39 Stress Intensity Factor plot

2.7 Step 7: Manual Crack Growth

We will "manually" propagate the crack front; this lets us examine the predicted crack growth to determine suitable parameters for fitting and extrapolation before proceeding with automated crack growth, which is described in Section 2.8.

Step 7.1: Select Grow Crack

From the FRANC3D menu, select **Cracks →Grow Crack**, Fig 2.40. In the first panel, Fig 2.41, switch to Quasi-Static crack growth and select **Next**. Use the Max Tensile Stress theory to determine the kink angle, Fig 2.42; in this model, crack growth is mostly planar. Select **Next**. Use a Power Law relationship to determine relative growth among points along the crack front, Fig 2.43; we will use the default power n=2. There is only one load step in the model. Select **Finish** to proceed. You will be asked if you want to save the growth parameters to a file, Fig 2.44, you can select **No**.

This quasi-static growth model can be considered a simplified Paris model; it is simpler to use than the subcritical growth model. Consult the Reference document for descriptions of crack extension models.

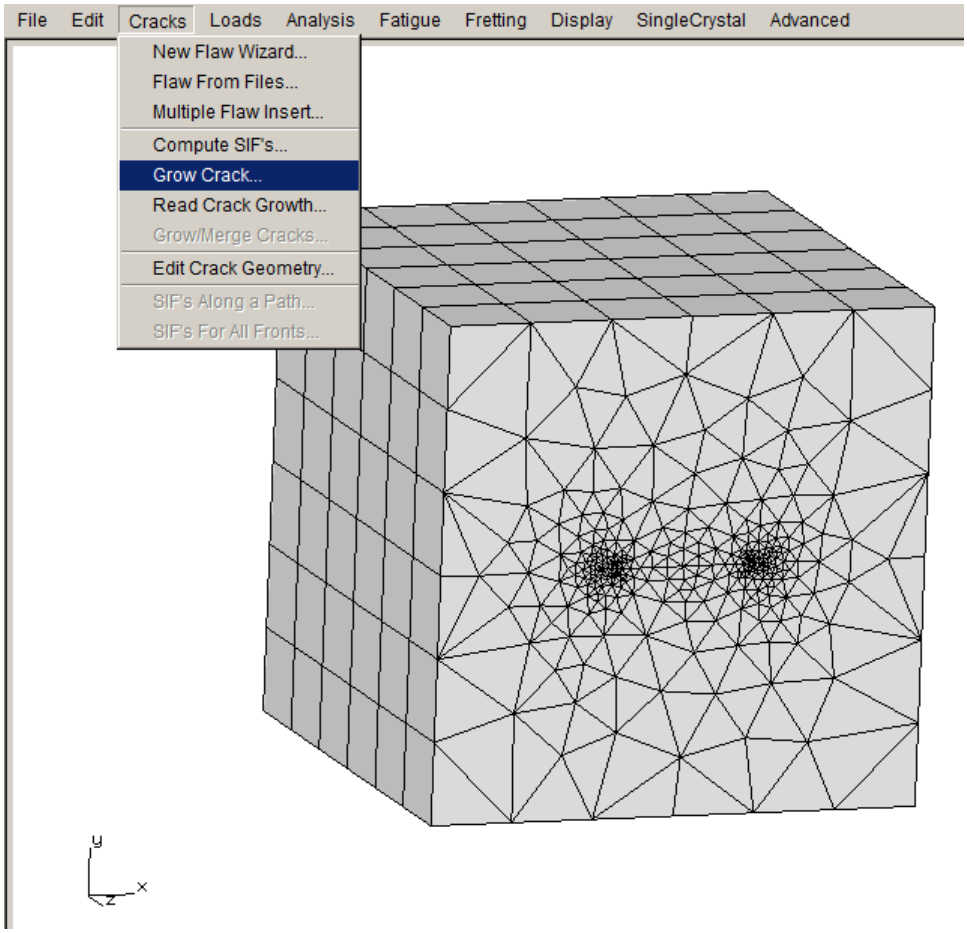


Figure 2.40 Grow Crack menu.

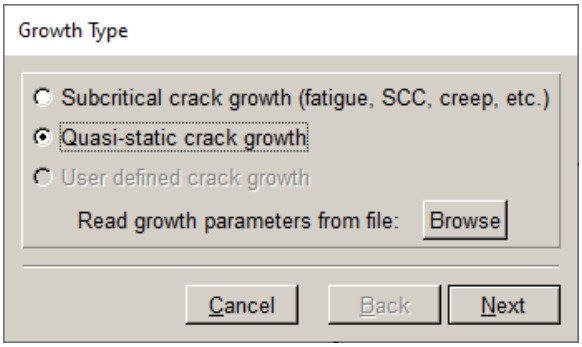


Figure 2.41 Crack Growth wizard – first panel.

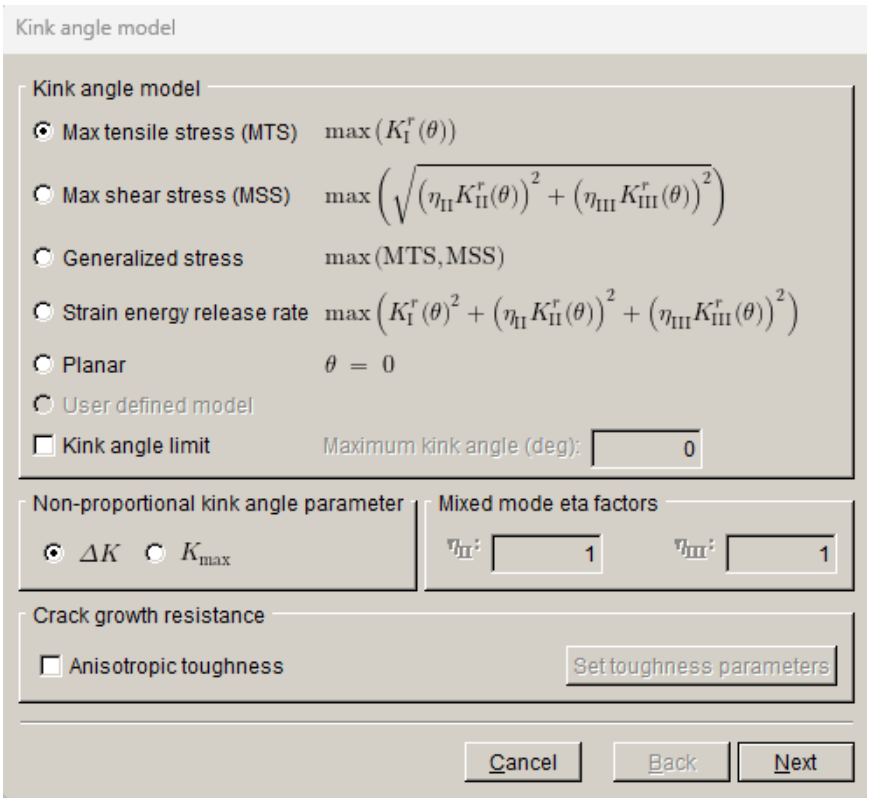


Figure 2.42 Crack Growth wizard – second panel using quasi-static growth.

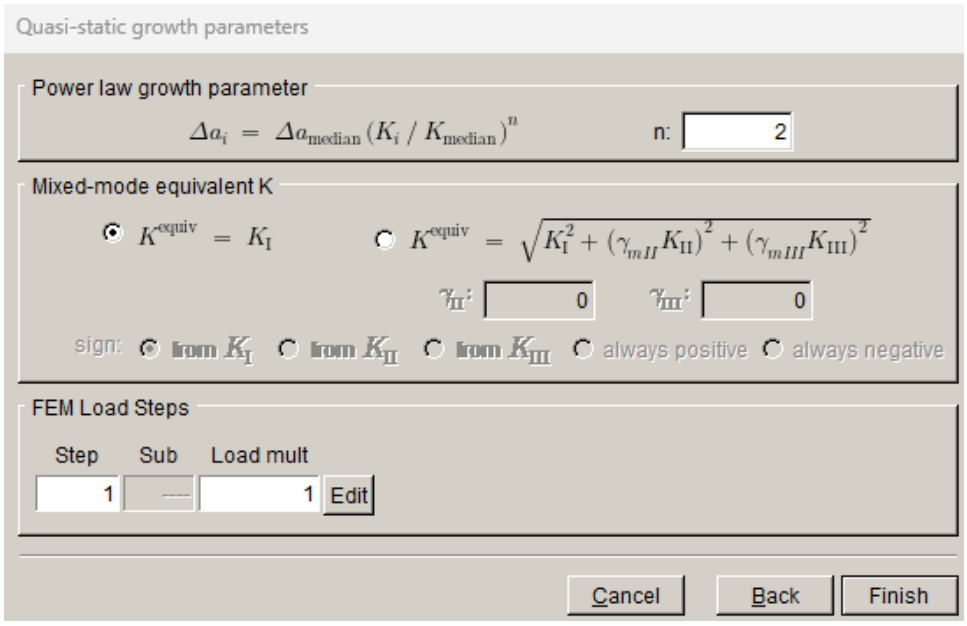


Figure 2.43 Crack Growth wizard – third panel using quasi-static growth.

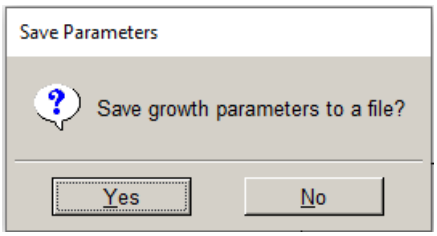


Figure 2.44 Save growth parameters dialog.

Step 7.2: Specify Fitting and Extrapolation

The next panel, Fig 2.45, allows you to specify the crack front point fitting parameters. You can double click on the (crack) view to see the crack surface; it should be the default view.

The median extension is set to 0.15; this value usually is set automatically based on the initial template radius so your value might be different. We change it to 0.12 and turn on Mesh and Markers (top right checkbox in Fig 2.45). The green boxes represent the computed new front points, and the blue line is the curve-fit, Fig 2.46. Note that the scale node is 0.5; the computed extensions are sorted from minimum to maximum, and the scale node can vary from 0 to 1. For a scale node of 0.5, the median extension is set to the user-specified value, and all other extensions are scaled accordingly.

A Fixed order polynomial with order set to 3 and extrapolation set to 3% at both ends of the crack front is the default; the fitting parameters might need to be adjusted as the crack grows. A 4^{th} order polynomial provides a slightly better fit in this case; the guidelines for curve-fitting are to use the simplest fit that is reasonable. The extrapolation is automatically adjusted to make sure the end points of the fitted-front fall outside the model surface. We stick with the default 3^{rd} order polynomial; select **Next** to proceed.

Note that the fitted (blue) curve must be extrapolated so that both ends fall outside of the model, but it should not be extrapolated too much. Normally, FRANC3D will automatically find reasonable values for extrapolation if Allow fit adjustment is turned on.

The median extension must be enough to provide finite space between the current and new fronts along the entire front to define new crack geometry. It is best to specify the extension (and have cycles computed) as opposed to specifying the cycles and having extension computed.

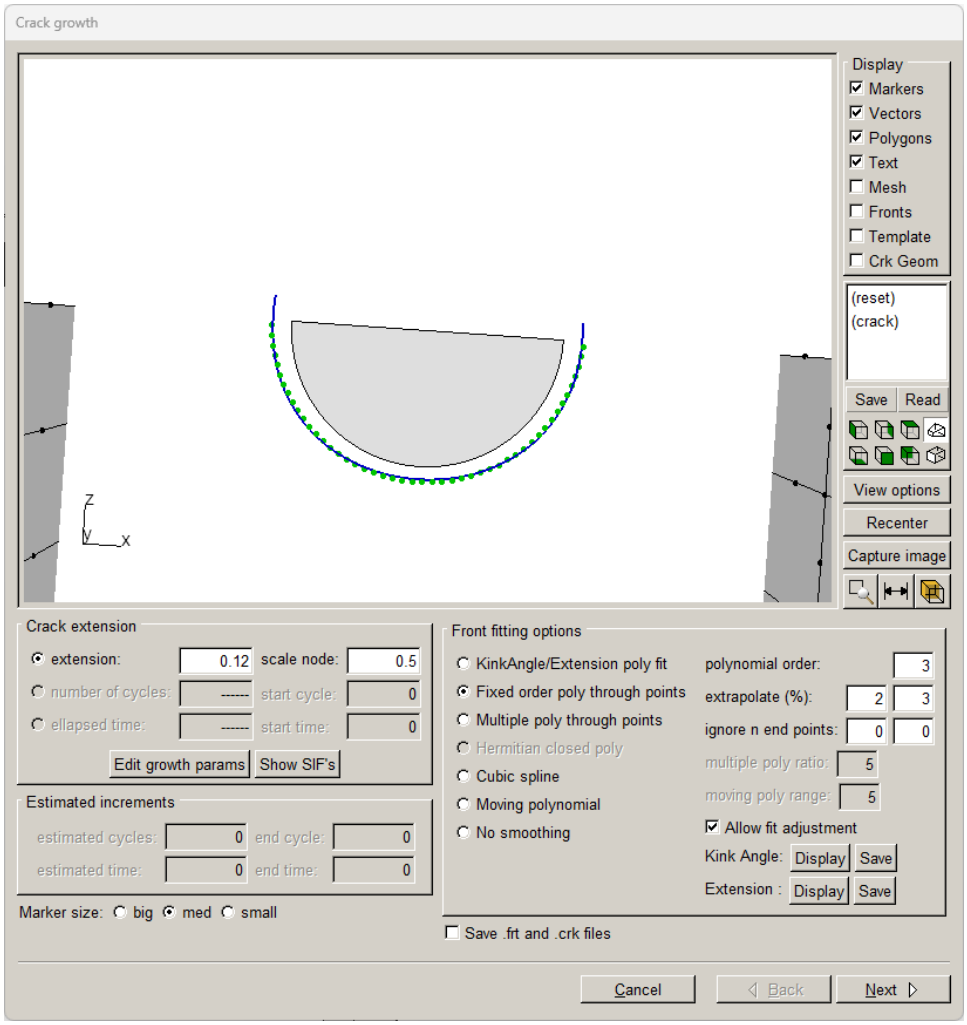


Figure 2.45 Crack growth wizard panel for crack front fitting options

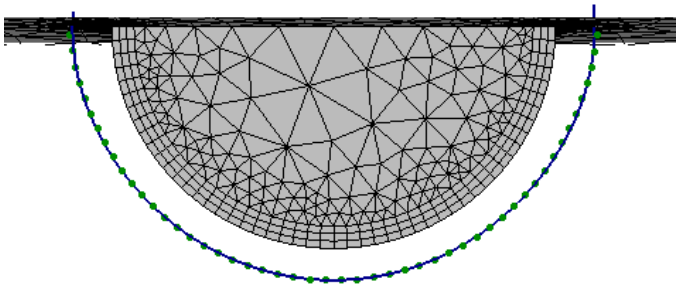


Figure 2.46 Curve-fit through front points.

Step 7.3: Specify Crack Front Template

The final panel, Fig 2.47, allows you to specify the crack front mesh template parameters. The crack front mesh template extends beyond the model surface, which corresponds with the front-fit extrapolation. The template radius can be set as an absolute value or as a % of the median extension; use an absolute value of 0.1, which is the same as the initial crack template radius. Select **Next** on this panel when ready to proceed with crack insertion and remeshing.

Note that the crack geometry for the extended crack is inserted into the original uncracked model, so do not overwrite, or remove the original uncracked model files.

A template radius that is slightly less than the crack extension, Fig 2.48, works best for cases where there is kinking or turning of the crack front.

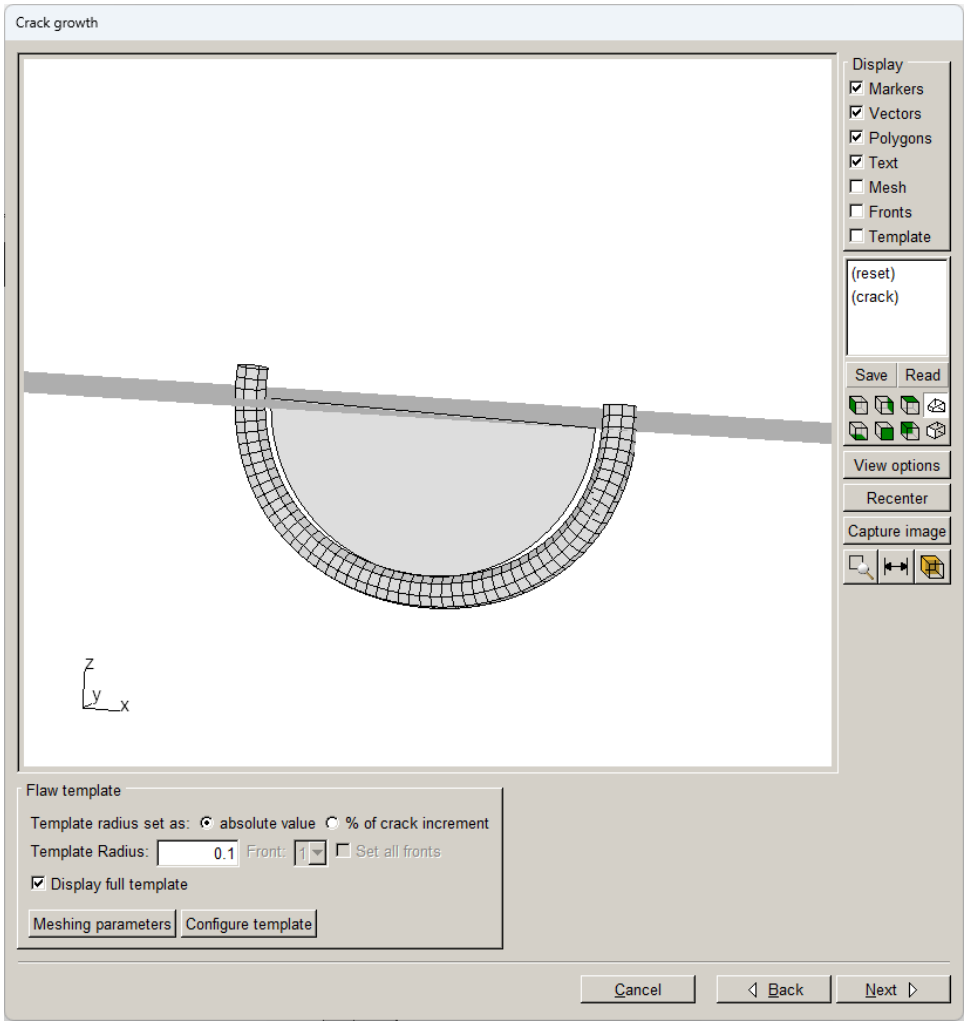


Figure 2.47 Crack growth wizard panel for mesh template options

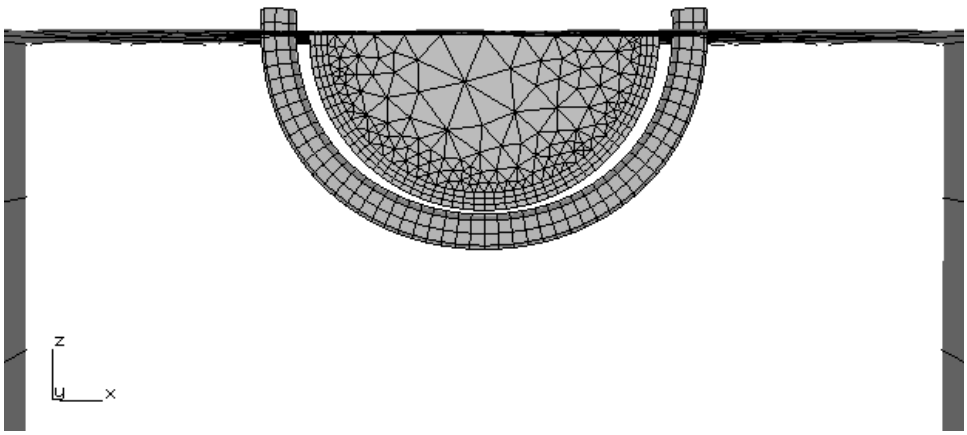


Figure 2.48 Template radius is smaller than extension.

The resulting model can be analyzed as was done for the initial crack (see Step 5 above). Note that you will want to give this model a different name, such as *Nastran_Cube_ step_001*, so that you do not overwrite the initial crack files. You could continue with another manual step of growth followed by static analysis, or you can try automated crack growth, which is described next.

2.8 Step 8: Automatic Crack Growth

This section describes the steps for automatic crack growth starting from the initial crack model. We start with the existing FRANC3D model, using the model created in Section 2.4 and 2.5. In most cases, we would simply do automatic growth after the manual step described in Section 2.7, but this will not show all the dialogs, so we restart from the initial crack model.

Step 8.1: Open FRANC3D Restart File

Start with the FRANC3D graphical user interface (see Fig 1.1) and select **File →Open**. Select the file name specified in Section 2.5, called *Nastran_Cube_Subdivide.fdb*. Select **Accept**. The model will be read into FRANC3D (along with the results that were created when running the static analysis). We will ignore the fact that we already propagated the initial crack in the previous step (but we will use the settings). The next section describes the steps for automatic crack growth analysis.

Step 8.2: Select Crack Growth Analysis

From the FRANC3D menu, select **Analysis →Crack Growth Analysis**, Fig 2.49. The first panel of the wizard, Fig 2.50, allows you to choose the method for computing SIFs. We use the default M-integral and select **Next**. If this panel does not show up, the SIF method is chosen automatically based on your earlier choice when plotting SIFs.

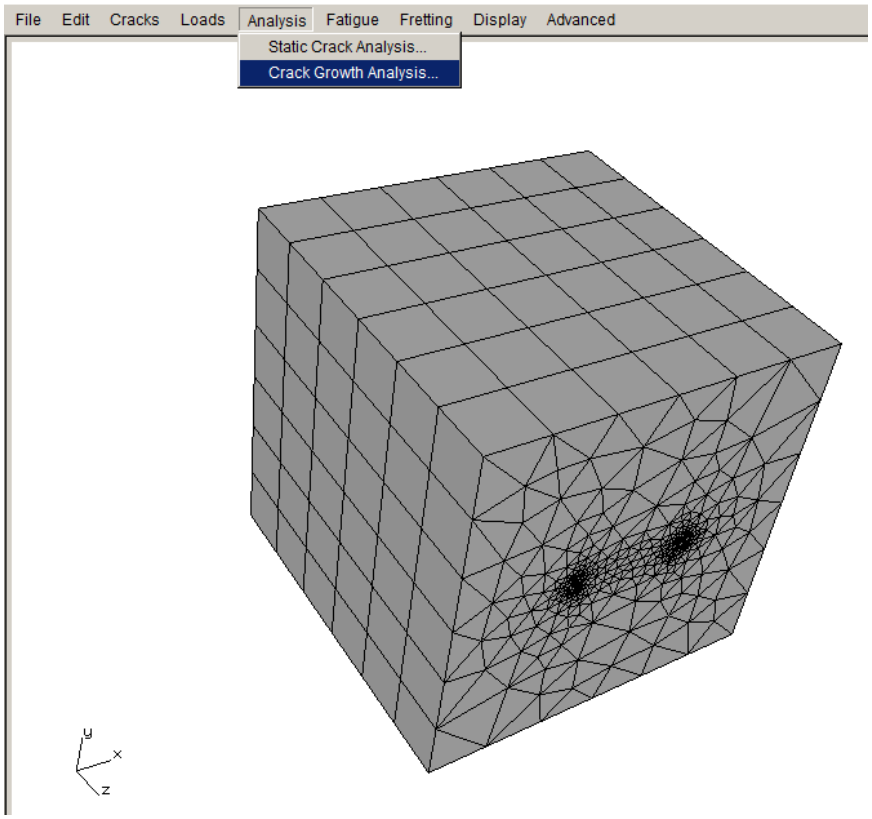


Figure 2.49 Crack Growth Analysis menu

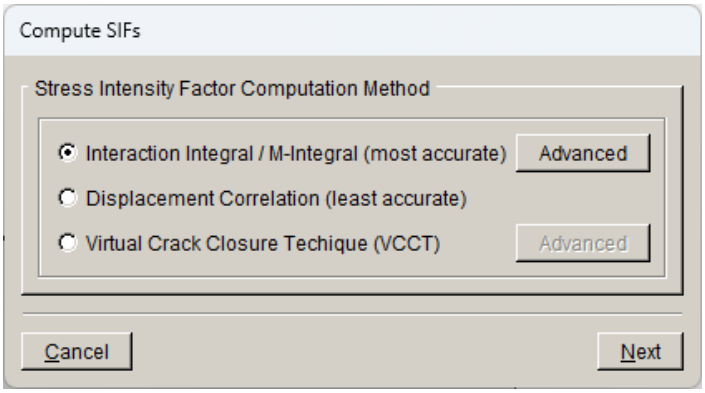


Figure 2.50 Crack Growth Analysis wizard – first panel.

Step 8.3: Specify Growth Rules

The second panel, Fig 2.51, lets us set the crack growth type; we use quasi-static growth. Select **Next** to display the third panel, Fig 2.52; use the default MTS theory to compute the kink angle.

Select **Next** to display the fourth panel, Fig 2.53; use the default settings for the quasi-static growth; there is only one FEM load step to select. Select **Next**. You will be asked if you want to save the growth parameters to a file; you can select **No**.

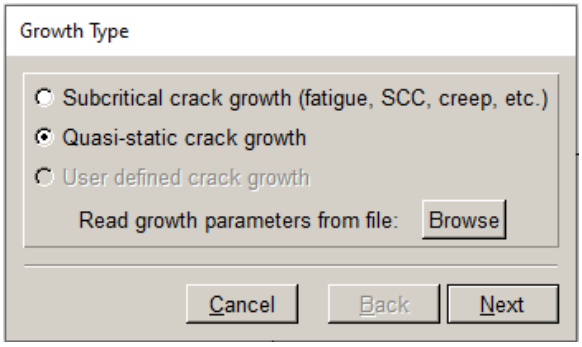


Figure 2.51 Crack Growth Analysis wizard – second panel.

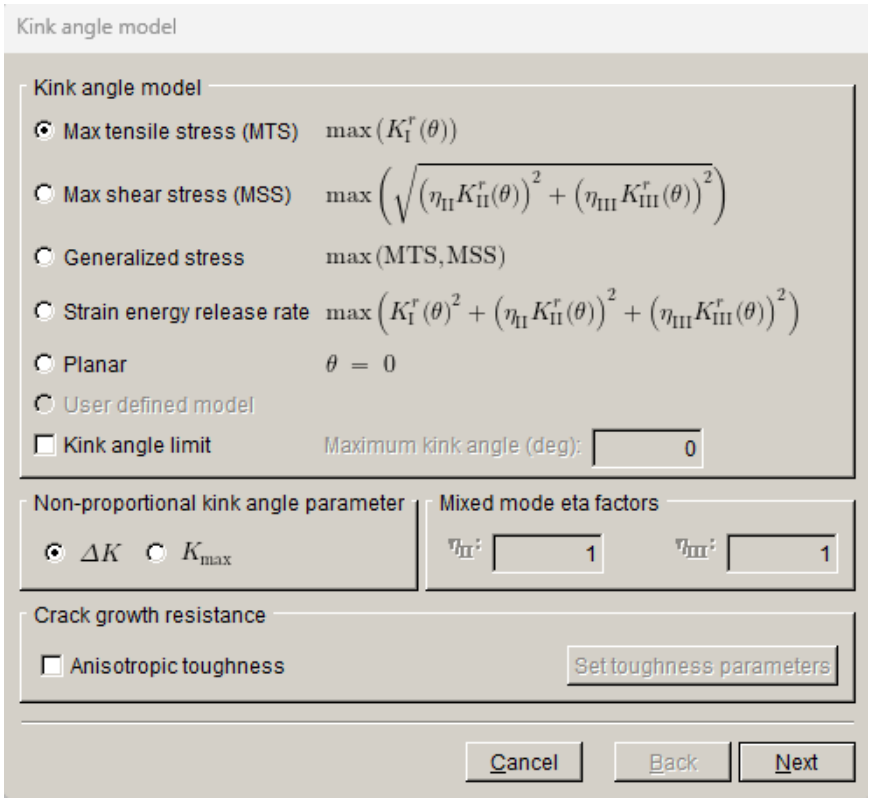


Figure 2.52 Crack Growth Analysis wizard – third panel.

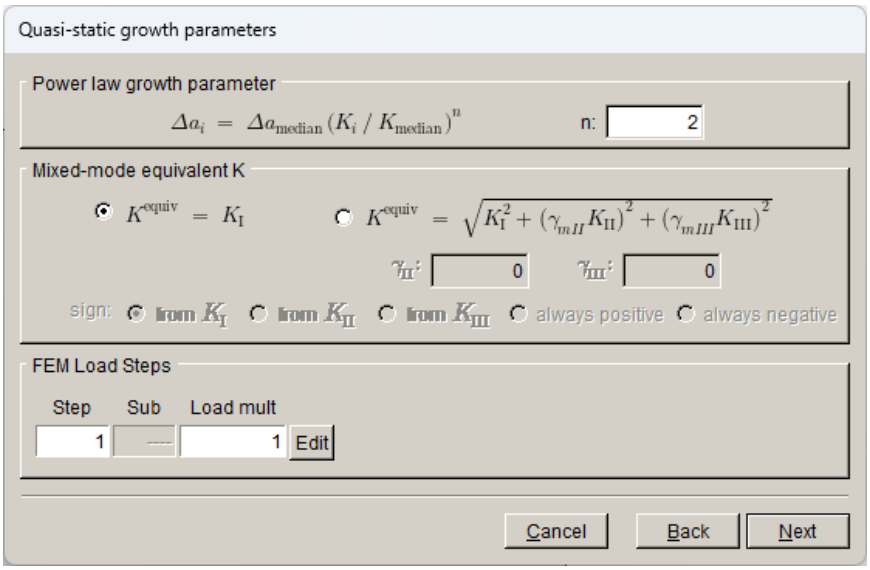


Figure 2.53 Crack Growth Analysis wizard – fourth panel.

Step 8.4: Specify Fitting and Template Parameters

The fifth panel, Fig 2.54, lets us set the crack front fitting and mesh template parameters. Use the default 3rd order polynomial and 2% extrapolation. Set the template radius to an absolute value of 0.1. Select **Next**.

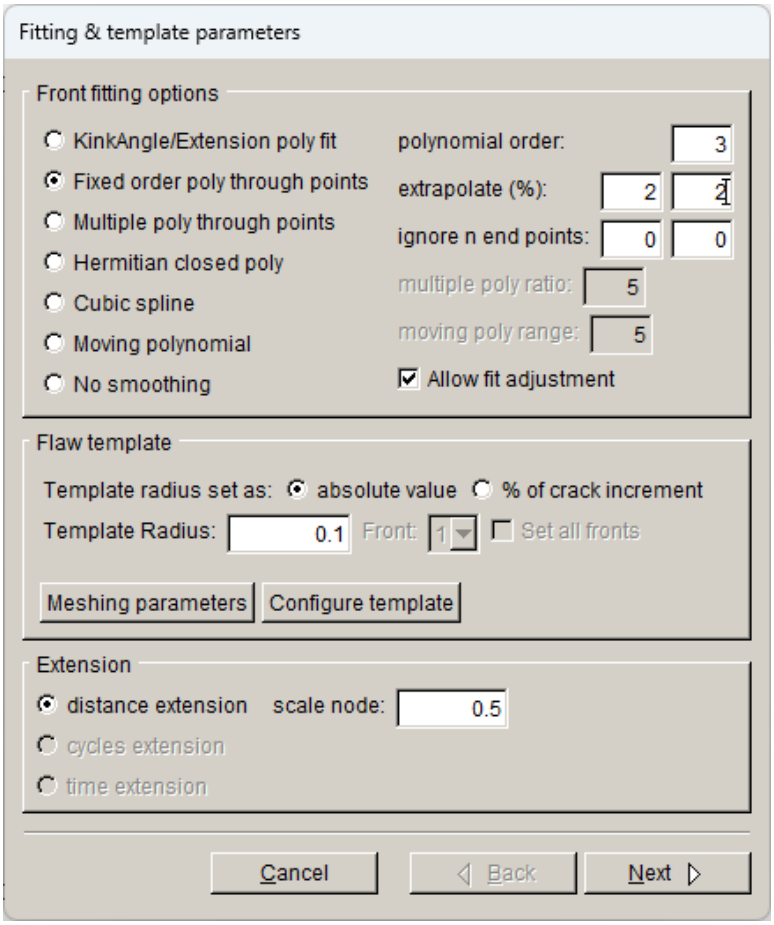


Figure 2.54 Crack Growth Analysis wizard – fifth panel.

Step 8.5: Specify Growth Plan

The sixth panel, Fig 2.55, lets us define the amount of extension for each step of crack growth. Set the number of steps to 5 and the constant median extension to 0.2. Select **Next**.

Note that we are taking a fairly large crack growth step; this is to limit the number of steps and time for this tutorial. Large steps can sometimes lead to issues with oscillations in the shape of the crack fronts (see Section 9 of the User's Guide), but for this model 0.2 is acceptable.

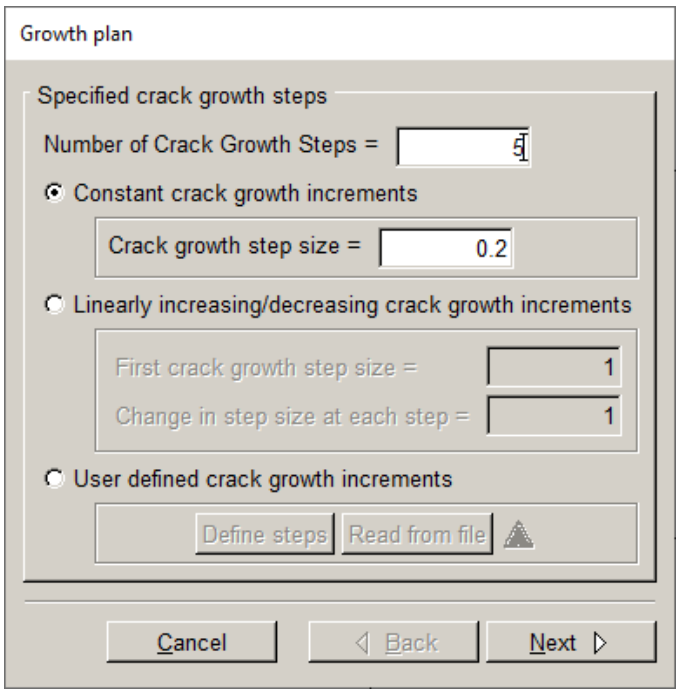


Figure 2.55 Crack Growth Analysis wizard – sixth panel.

Step 8.6: Analysis Code Settings

The seventh panel, Fig 2.56, shows the Base filename and analysis settings. The Current crack growth step defaults to 0 – representing the initial crack. If we "manually" propagated the crack first, we should set this to 1. FRANC3D will extend the initial crack based on the growth rule defined in the previous panels and then name the resulting set of files as *Nastran_Cube_Subdivide_STEP_001*.*. Subsequent file names will have the step number incremented as the automatic analysis proceeds.

Some options will be specific to your site. You should have already set the NASTRAN executable path in the FRANC3D Preferences. The Global model settings should be set automatically. The local + global connection is displayed in the next panel; select **Next**.

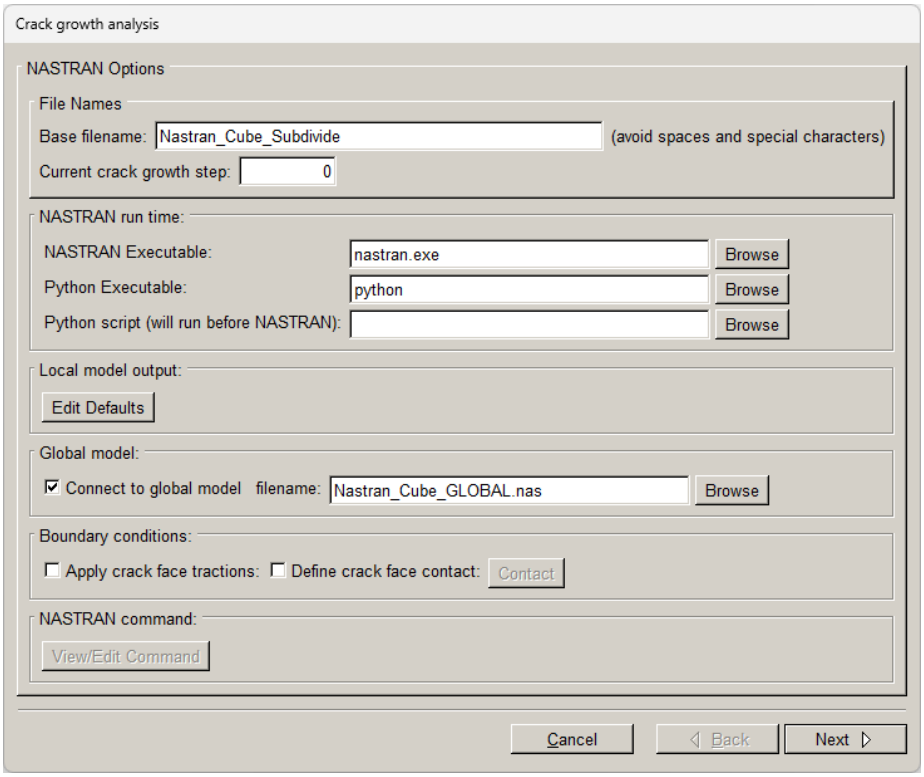


Figure 2.56 Crack Growth Analysis wizard – seventh panel.

The eighth panel, Fig 2.57, displays the local + global connection. The default settings use node-merging of the cut-surfaces between the local and global portions. Select **Finish** to start the automated crack growth process.

FRANC3D extends the initial crack first. It then saves the *.fdb*, *.bdf* and other files for the first crack step model with the name *Nastran_Cube_Subdivide_STEP_001*, and then NASTRAN starts in the background. If the analysis stops at any stage, it can be restarted from the last crack step. All the required restart *_STEP_#* files are retained. The model for any step can be read into FRANC3D to view the SIFs or to restart the analysis with a modified crack growth increment (for example).

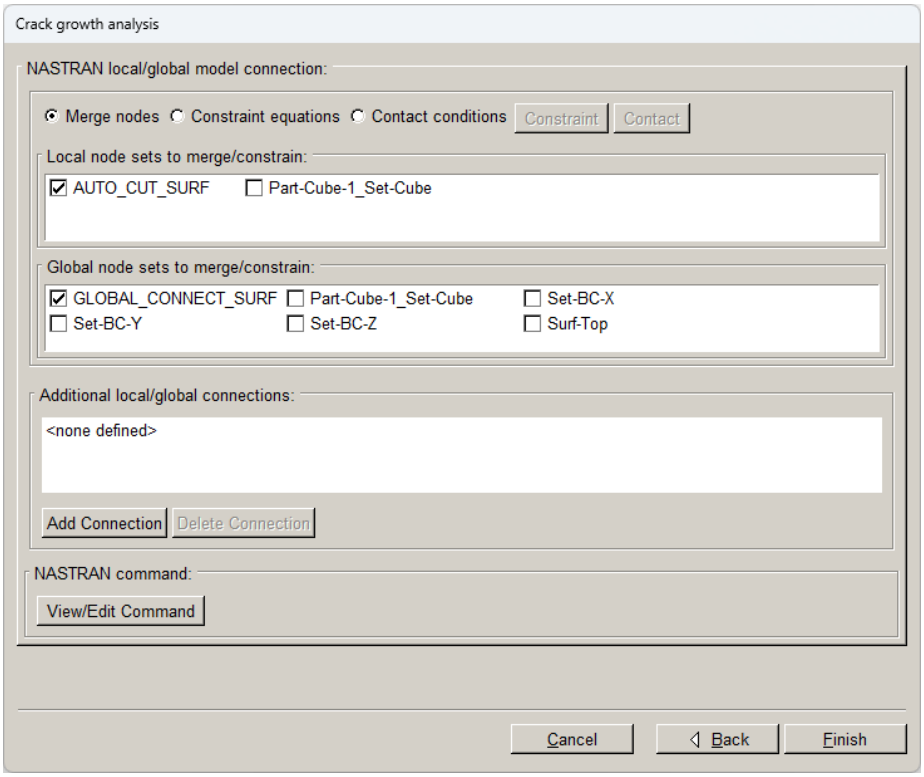


Figure 2.57 Local + Global connection options

A summary of the various files can be found in Section 3 of the User's Guide.

2.9 Step 9: SIF History and Fatigue Life

When the automatic crack growth analysis from Step 8 has finished, the SIF history can be displayed. If you still have FRANC3D open and the crack growth from Step 8 is done, you can proceed with Step 9.1. Otherwise, you can restart FRANC3D and read in the *.fdb* file for the last step, using the **File → Open** menu option.

Step 9.1: Select SIFs Along a Path

From the FRANC3D menu, select **Cracks →SIFs Along a Path**, Fig 2.58. If the SIFs have not been computed yet, the Compute SIFs dialog (see Fig 2.38) will be displayed; leave the defaults and select **Finish**.

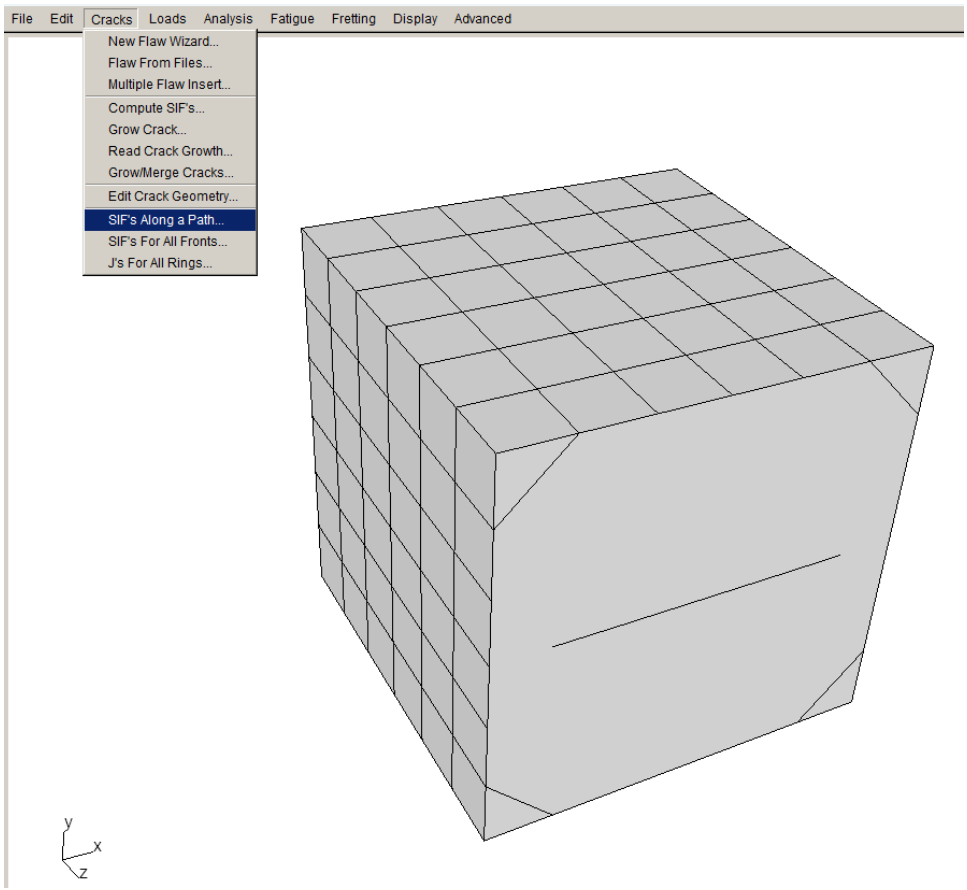


Figure 2.58 SIFs Along a Path menu option

The Stress Intensity Factors (along a path) dialog should appear, Fig 2.59. The crack fronts are displayed on the left along with the path through the fronts; the SIF history along the path is shown in the graph on the right. You can use the tabs above the graph to plot the Mode II and III SIFs as well as the elastic J-integral and T-stress values along the path. You can define a new path, Fig 2.60. You can export the data also; for example, you might need to export the Mode I

SIF history (K vs a) to compute fatigue cycles using a different program. The exported data files are ASCII txt files.

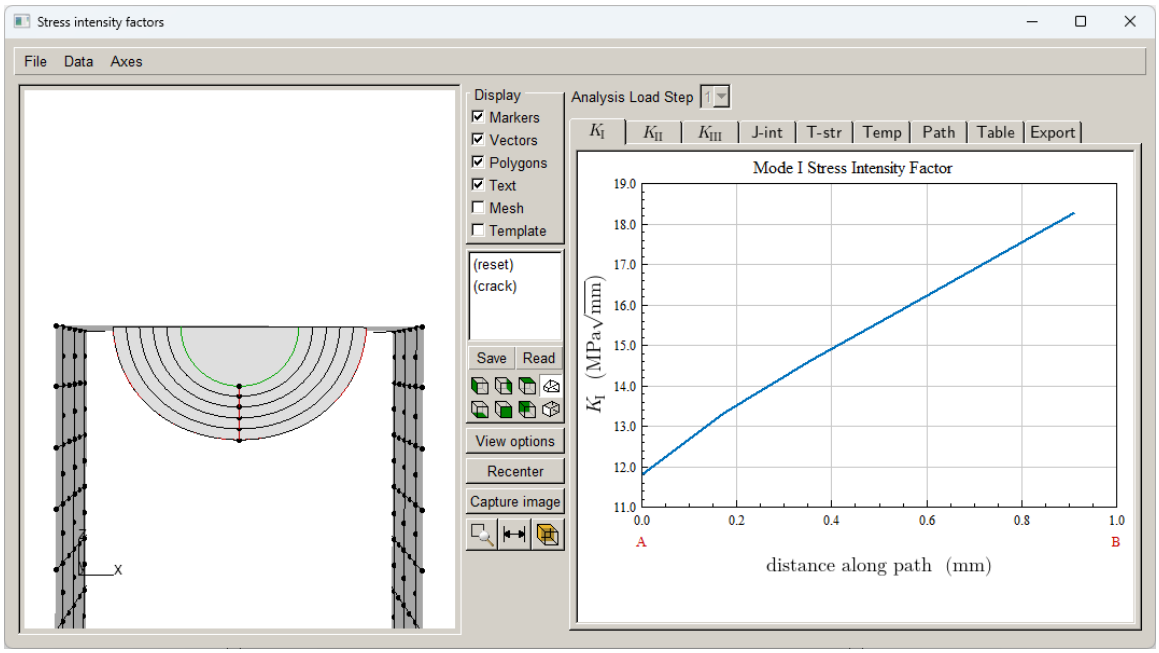


Figure 2.59 SIFs Along a Path dialog – KI plot

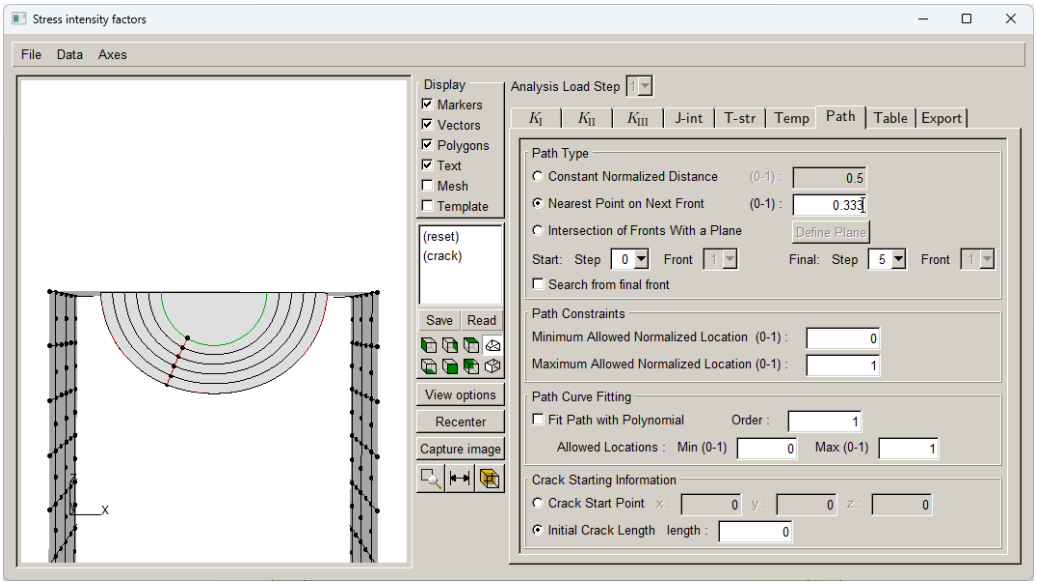


Figure 2.60 SIFs Along a Path dialog – Define Path

Step 9.2: Select SIFs For All Fronts

From the FRANC3D menu, select **Cracks →SIFs For All Fronts**. If the SIFs have not been computed yet, the Compute SIFs dialog will be displayed; leave the defaults and select **Finish**.

The Stress Intensity Factors (for all fronts) dialog should appear, Fig 2.61. The crack fronts are displayed on the left and the SIFs for all crack fronts are displayed on the right. You can use the tabs above the plot to display Mode II and III SIFs as well as the elastic J-integral and T-stress values. If there are multiple crack fronts or multiple load steps, these can be selected using the drop-down lists; this will be seen in subsequent tutorials.

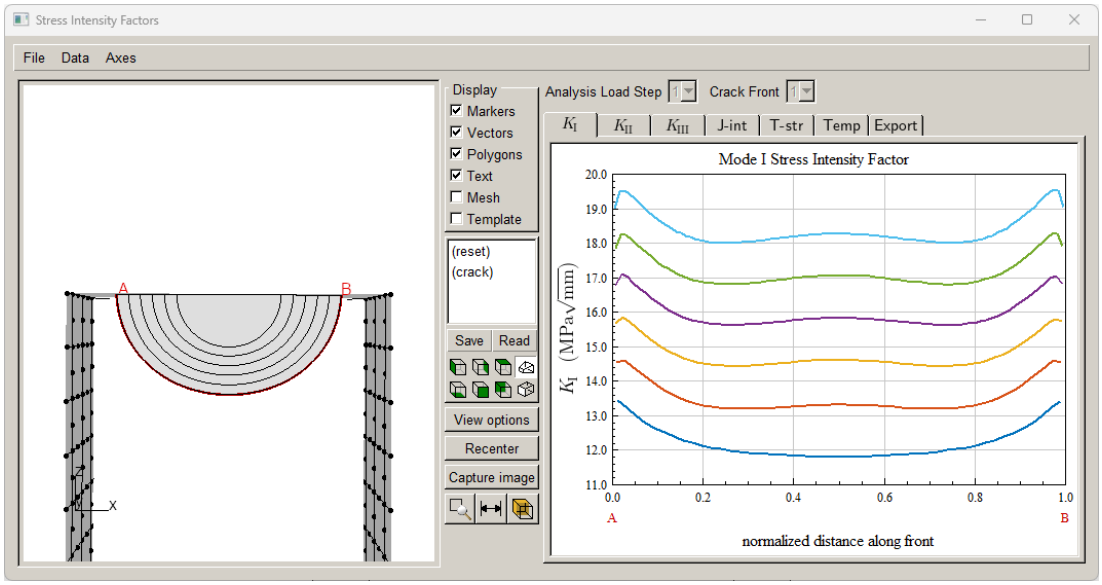


Figure 2.61 SIFs for all Fronts dialog – KI plot

Step 9.3: Select Fatigue Life Predictions

Fatigue life or cycles can be computed next. From the FRANC3D menu, select **Fatigue →Fatigue Life Predictions**, Fig 2.62. If the SIFs have not been computed yet, the Compute SIFs dialog will be displayed; leave the defaults and select **Finish**.

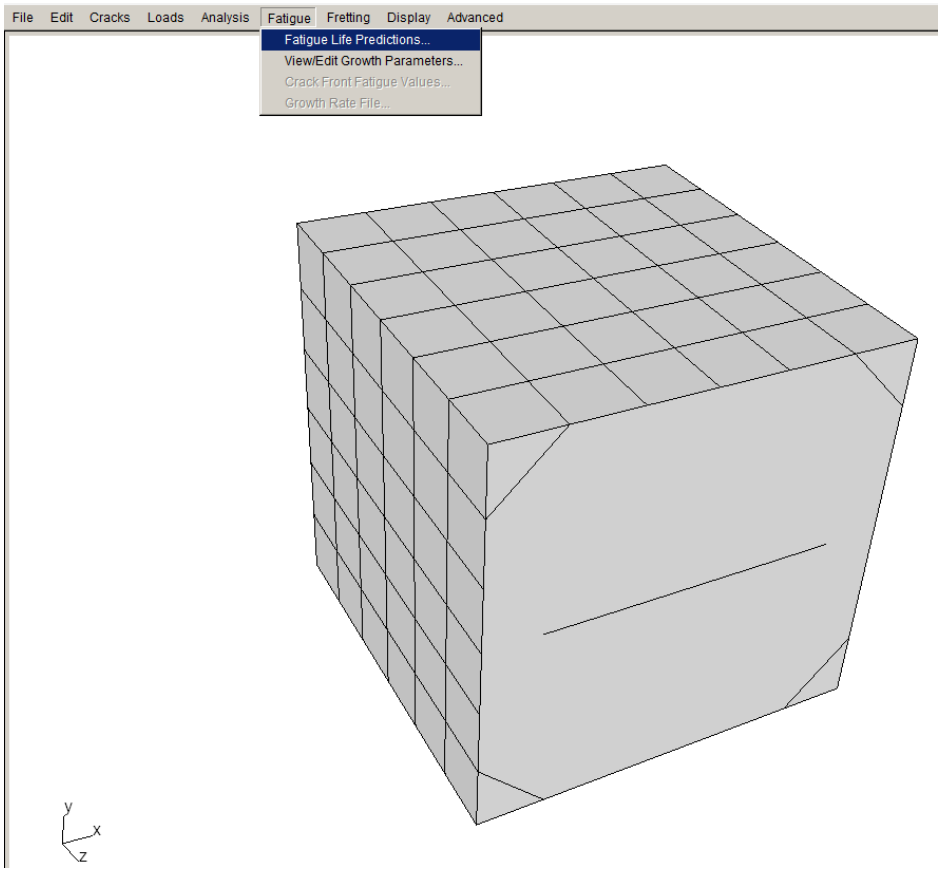


Figure 2.62 Fatigue Life Predictions menu option

The Fatigue Life dialog should appear as in Fig 2.63. The crack fronts are displayed on the left and the window on the right side should be blank (assuming that lifing parameters have not been defined previously). You must **Set** (or **Read**) **Parameters**. Selecting **Set Parameters** displays the dialog shown in Fig 2.64.

The units are MPa and mm; select the **Change** button to display the dialog in Fig 2.65 and set the units to MPa and mm; select **Accept**.

The FEM units can also be set in the main FRANC3D window using the **Edit** menu.

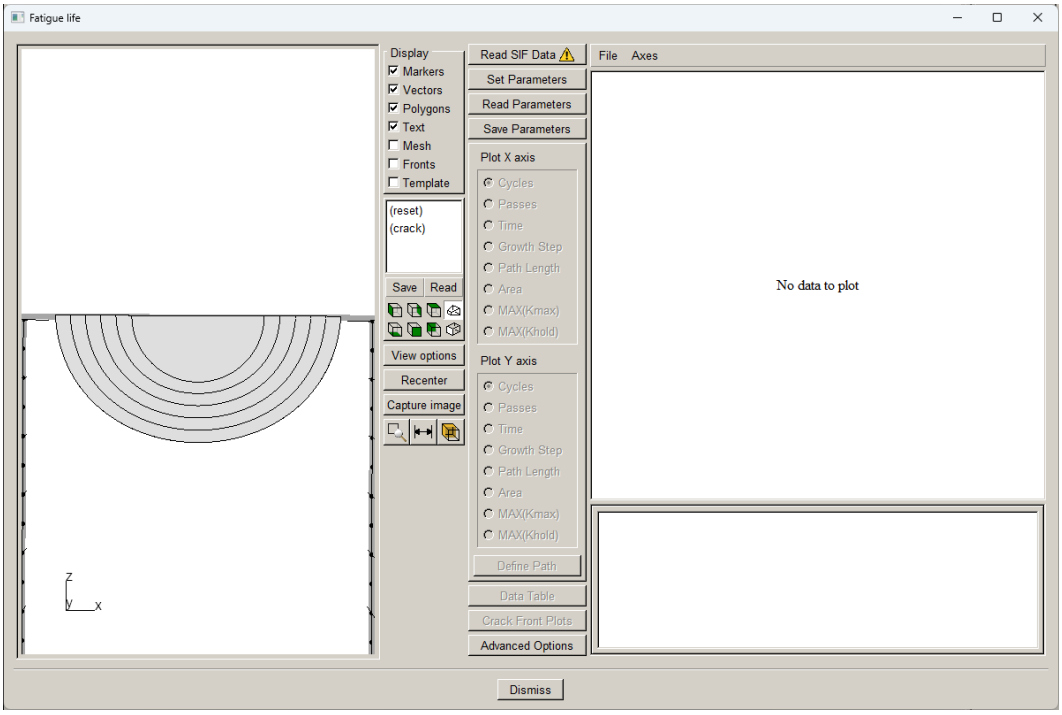


Figure 2.63 Fatigue Life dialog

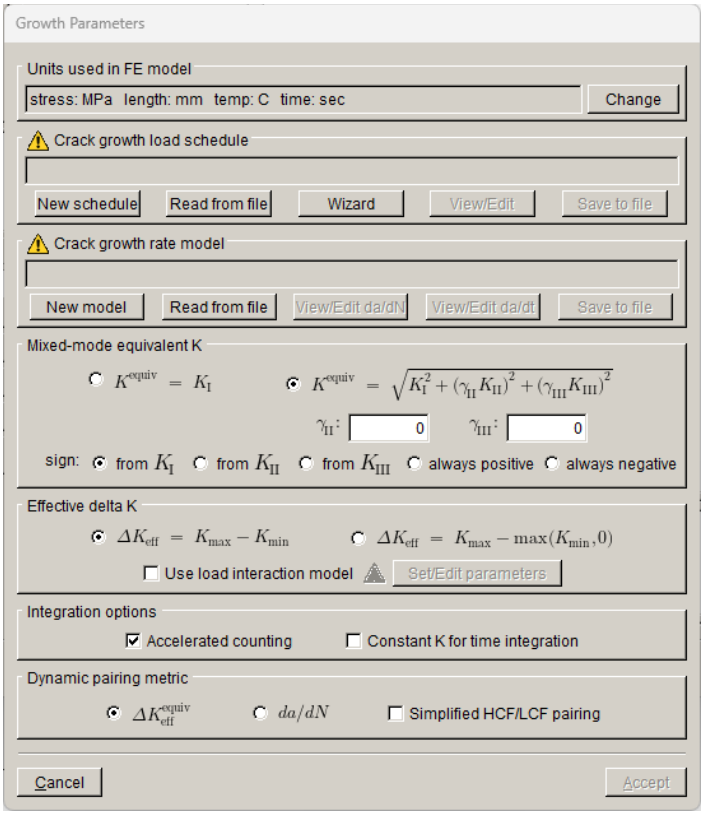


Figure 2.64 Fatigue Life parameters

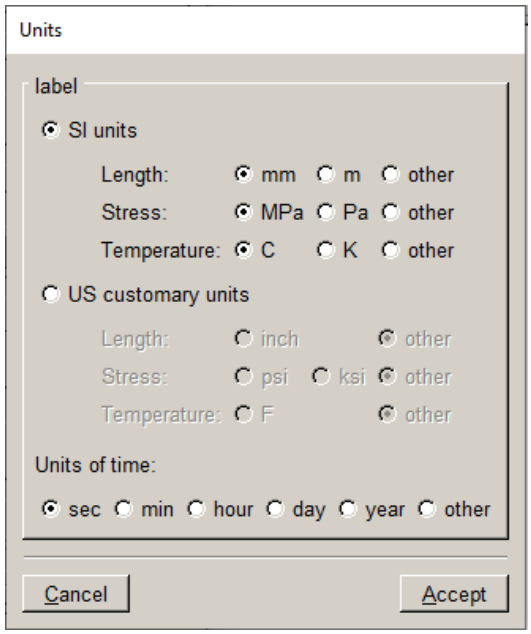


Figure 2.65 FEM units dialog

The crack growth load schedule is defined next. Select the **New Schedule** button (see Fig 2.63) to display the dialog in Fig 2.66. Select the Schedule and then select the **Add** button; see right side image of Fig 2.66. This will display the dialog shown in Fig 2.67; we use a simple cyclic load schedule where the applied load produces Kmax, and Kmin is zero. Select **Accept** to return to the Load Schedule, Fig 2.68.

The stress ratio is set to 0.0. There is only one load case, which represents the Kmax condition. The Repeat count is set to FOREVER. Select **Accept** to finish the load schedule. There is only one event in the schedule so the repeat forever can be set for either the event or the schedule; it is usually best to set Repeat FOREVER for the schedule.

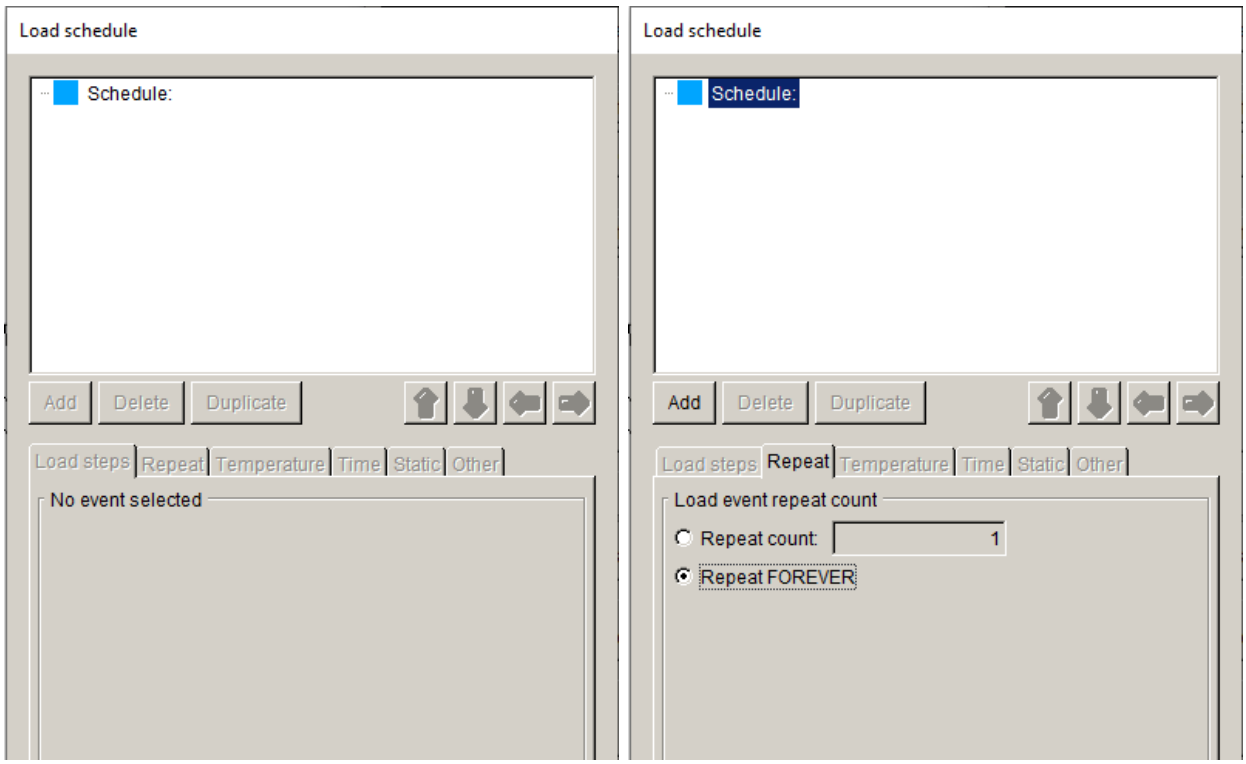


Figure 2.66 New Load Schedule dialog

Figure 2.67 Event type dialog

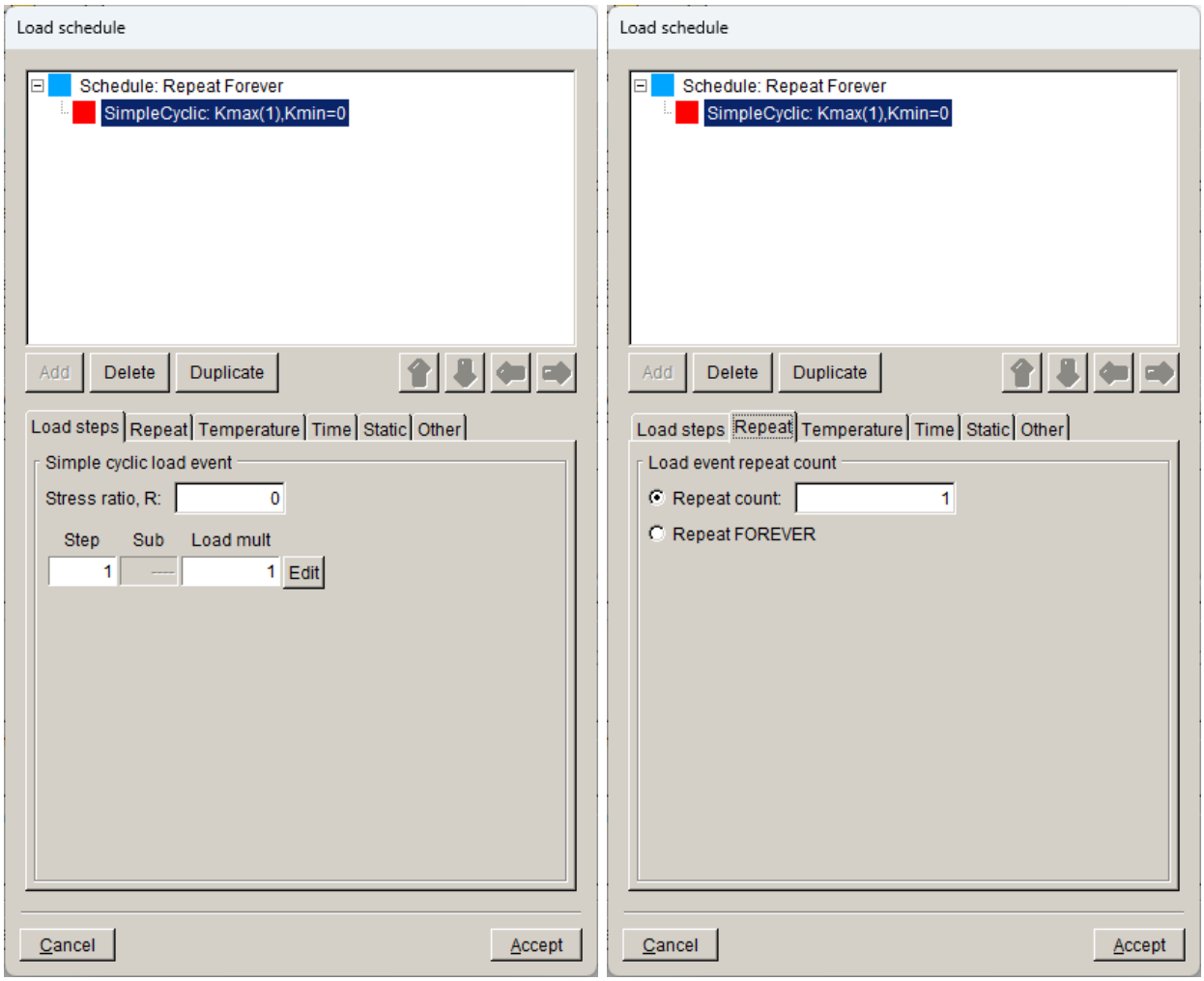


Figure 2.68 Simple cyclic load event repeated once and schedule repeated forever

The crack growth rate model is defined next. Select the **New Model** button (see Fig 2.64) to display the dialog in Fig 2.69. Choose the cyclic loading growth rate model, and select **Next** to display the dialog in Fig 2.70.

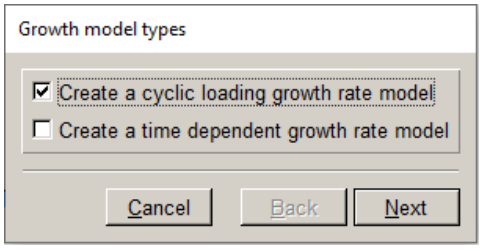


Figure 2.69 Growth model type dialog

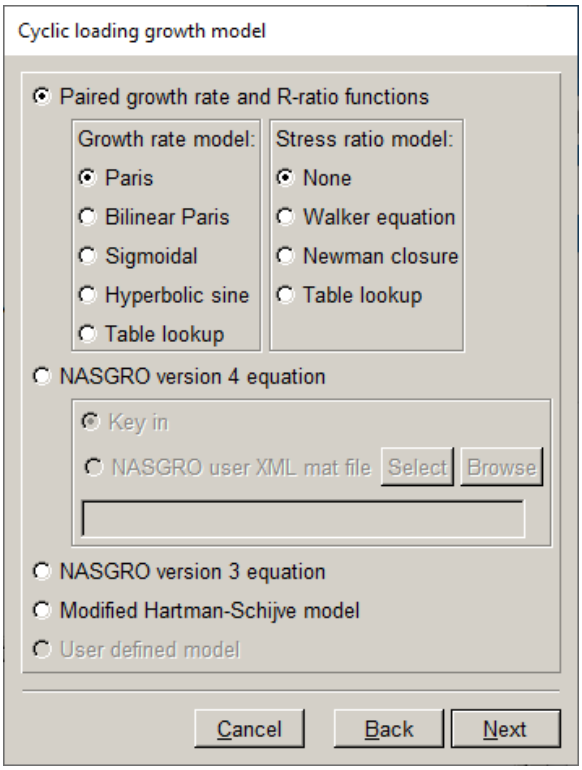


Figure 2.70 Cyclic loading growth model dialog

We use a Paris growth model, which is the default; select **Next** to display the dialog in Fig 2.71. The growth rate model will be temperature independent. Select **Next** to display the Paris growth model dialog, Fig 2.72.

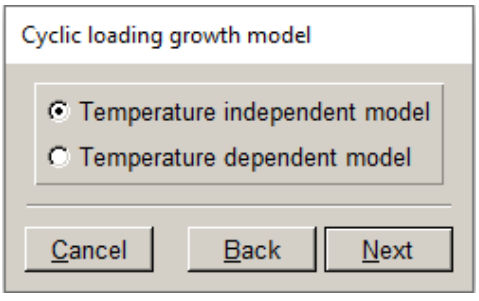


Figure 2.71 Temperature dependent or independent growth rate model dialog

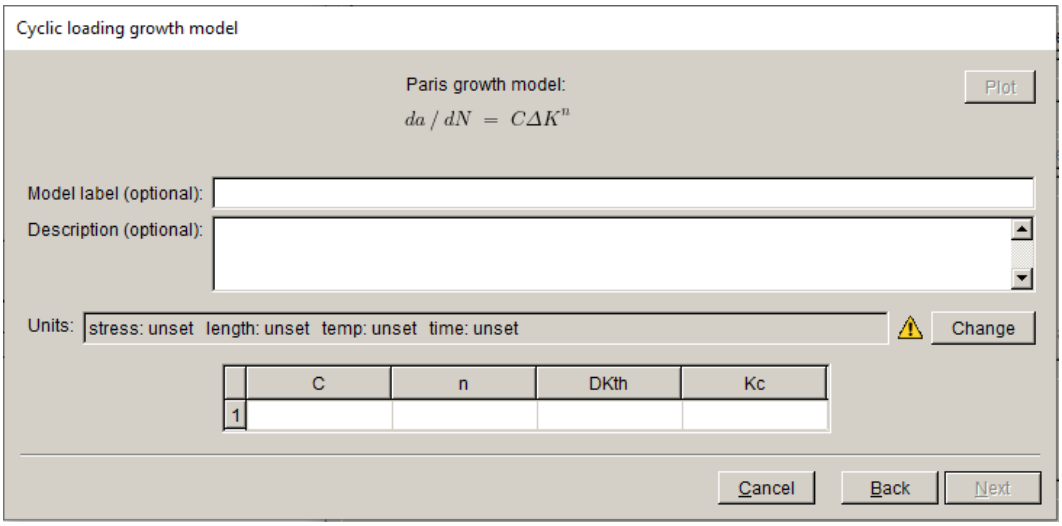


Figure 2.72 Paris growth rate model dialog

First, set the units for the Paris growth model. Select the **Change** button to display the Units dialog, Fig 2.73. Set the units the same as the FEM units that we set earlier, which were MPa and mm. If your growth model units are different from the FEM units, FRANC3D will do the unit conversions automatically. Select **Accept**.

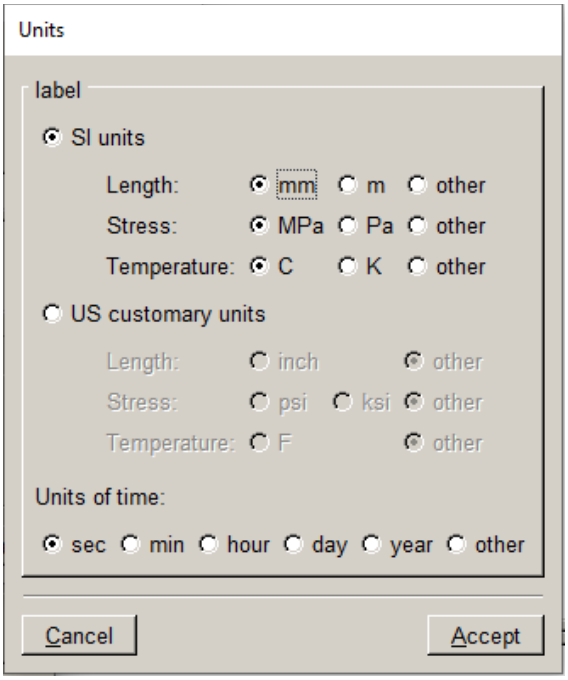


Figure 2.73 Paris growth rate model units

In the Paris growth model dialog, click on the fields for C, n, DKth and Kc and enter the appropriate values, Fig 2.74. Select **Next** to return the main dialog (see Fig 2.64). Other settings and options are left at their default values; select **Accept** (Fig 2.64) to return to the Fatigue Life dialog, Fig 2.75.

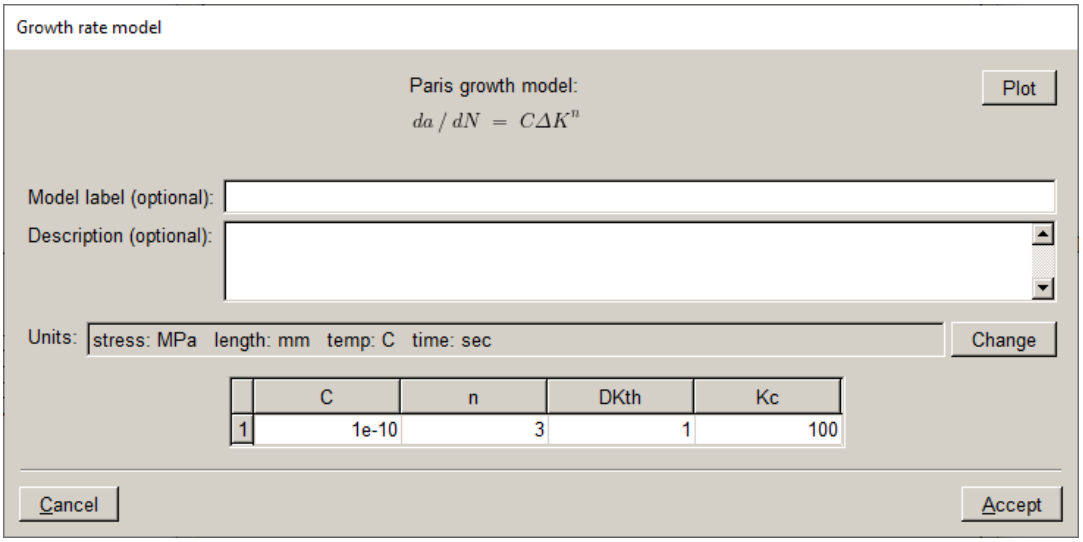


Figure 2.74 Paris growth rate model values

Note that the Paris exponent 'n' is set to 3 whereas we used n=2 for the quasi-static growth. In practice, you can use the "correct" fatigue model and material data during the growth, so that growth is consistent with the fatigue life computations. For a single crack front with relatively uniform K_I values along the front, the quasi-static growth option is acceptable. If there are multiple crack fronts with significantly different growth rates, the "correct" fatigue model should always be used.

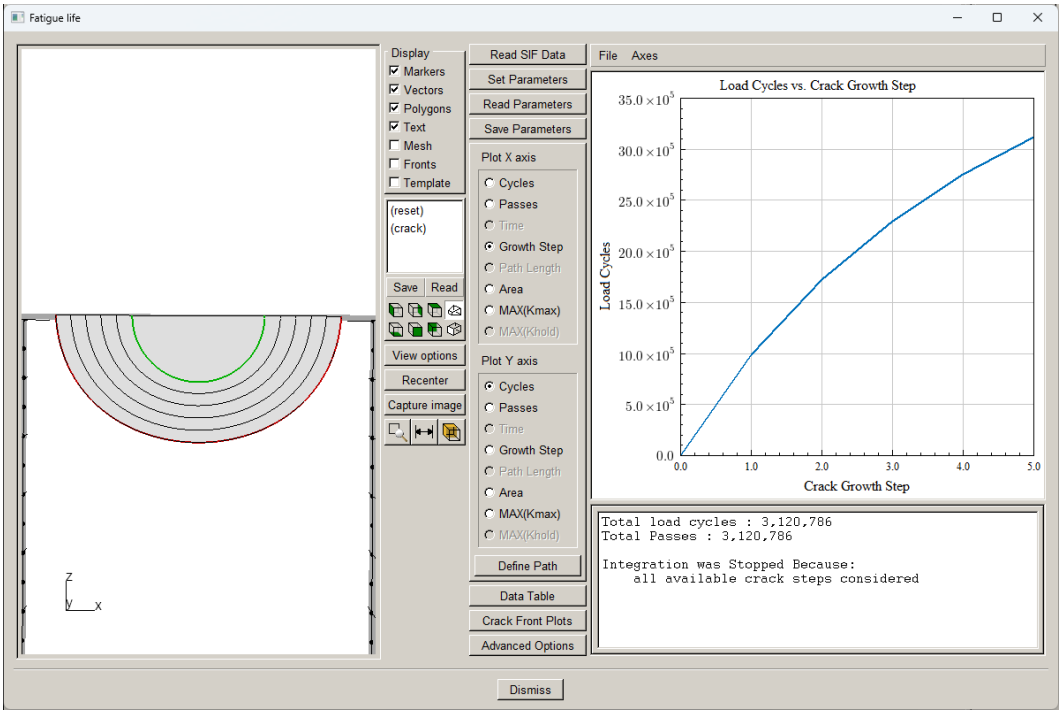


Figure 2.75 Fatigue life dialog showing Cycles vs Crack Growth Step

If we choose to plot cycles based on a path, we must select the Path radio button and then use the **Define** button to create a path, Fig 2.76. The default path is through the crack front midpoints. Once a path is defined, the cycles versus crack path length is displayed, Fig 2.77.

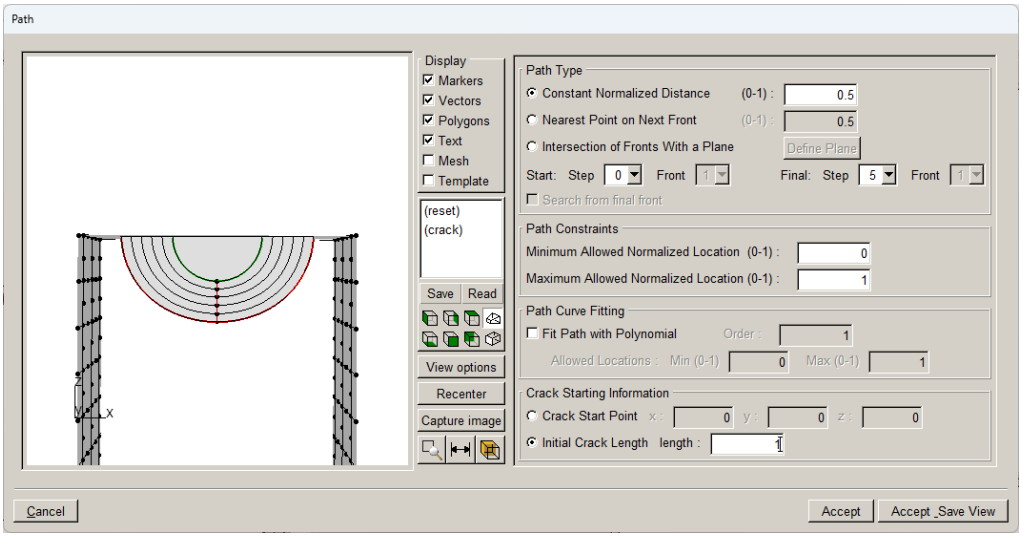


Figure 2.76 Define crack path dialog

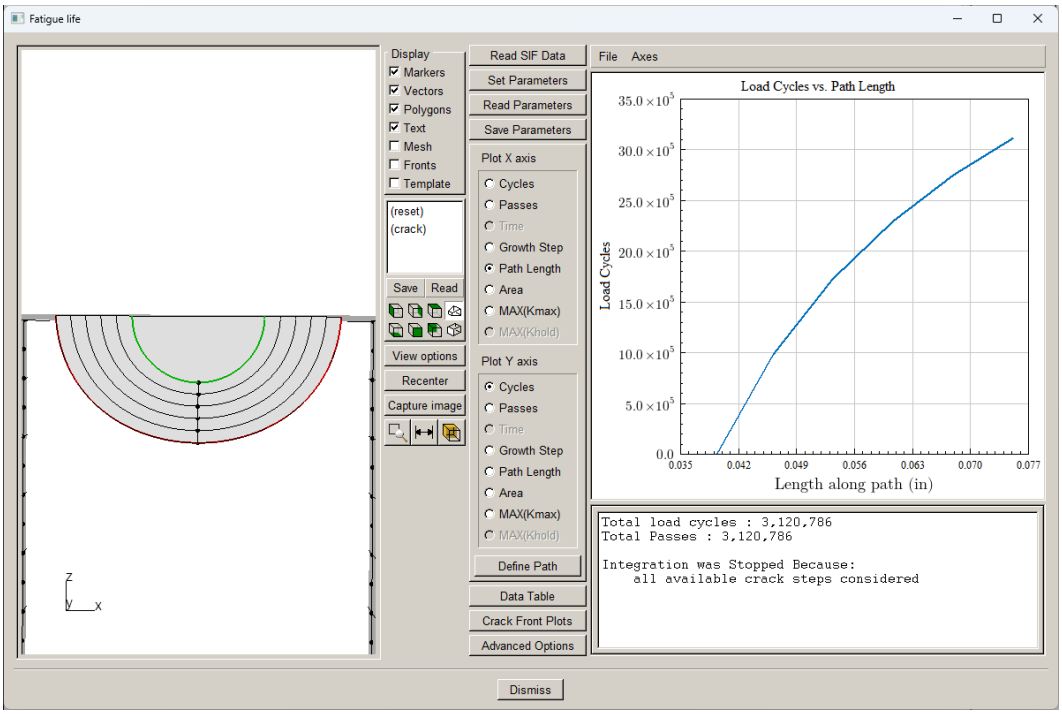


Figure 2.77 Fatigue cycles versus crack length

Appendix A: NASTRAN Keywords

The list of NASTRAN keywords that FRANC3D recognizes is provided here. FRANC3D must process some of the data, but a large amount of the *.bdf* file can be passed through to the *_GLOBAL.bdf* and then to the *_full.bdf*, which is the combination of the cracked/remeshed portion of the model and the *_GLOBAL.bdf* portion.

```
// initialize the dispatch 'dictionary'
// use NASTRAN tags listed above as keys

dispatch.Store(String("ID"), &NastranBdfReader::Ignore) ;
dispatch.Store(String("SOL"), &NastranBdfReader::SolToCend) ;
dispatch.Store(String("TITLE"), &NastranBdfReader::Title) ;
dispatch.Store(String("K2GG"), &NastranBdfReader::Dmig) ;
dispatch.Store(String("P2G"), &NastranBdfReader::Dmig) ;
dispatch.Store(String("SUBCASE"),&NastranBdfReader::Subcase) ;
dispatch.Store(String("SET"),&NastranBdfReader::SetControl) ;

dispatch.Store(String("PARAM"), &NastranBdfReader::Param) ;
dispatch.Store(String("GRID"), &NastranBdfReader::Node) ;
dispatch.Store(String("GRID*"), &NastranBdfReader::Node) ;
dispatch.Store(String("CHEXA"),&NastranBdfReader::Element) ;
dispatch.Store(String("CHEXA*"),&NastranBdfReader::Element) ;
dispatch.Store(String("CPENTA"),&NastranBdfReader::Element) ;
dispatch.Store(String("CPENTA*"),&NastranBdfReader::Element) ;
dispatch.Store(String("CTETRA"),&NastranBdfReader::Element) ;
dispatch.Store(String("CTETRA*"),&NastranBdfReader::Element) ;
dispatch.Store(String("CPYRAM"),&NastranBdfReader::Element) ;
dispatch.Store(String("CPYRAM*"),&NastranBdfReader::Element) ;

dispatch.Store(String("CBAR"),&NastranBdfReader::NativeElement) ;
dispatch.Store(String("CBEAM"),&NastranBdfReader::NativeElement) ;
dispatch.Store(String("CBUSH"),&NastranBdfReader::NativeElement) ;
dispatch.Store(String("CELAS2"),&NastranBdfReader::NativeElement) ;
dispatch.Store(String("CROD"),&NastranBdfReader::NativeElement) ;

dispatch.Store(String("CQUAD8"),&NastranBdfReader::NativeElement) ;
dispatch.Store(String("CTRIA6"),&NastranBdfReader::NativeElement) ;
dispatch.Store(String("CQUAD4"),&NastranBdfReader::NativeElement) ;
dispatch.Store(String("CTRIA3"),&NastranBdfReader::NativeElement) ;
dispatch.Store(String("CQUADR"),&NastranBdfReader::NativeElement) ;
```

```
dispatch.Store(String("CTRIAR"),&NastranBdfReader::NativeElement) ;
dispatch.Store(String("CQUAD8*"),&NastranBdfReader::NativeElement) ;
dispatch.Store(String("CTRIA6*"),&NastranBdfReader::NativeElement) ;
dispatch.Store(String("CQUAD4*"),&NastranBdfReader::NativeElement) ;
dispatch.Store(String("CTRIA3*"),&NastranBdfReader::NativeElement) ;
dispatch.Store(String("CQUADR*"),&NastranBdfReader::NativeElement) ;
dispatch.Store(String("CTRIAR*"),&NastranBdfReader::NativeElement) ;

dispatch.Store(String("RBE1"),&NastranBdfReader::Rbe) ;
dispatch.Store(String("RBE2"),&NastranBdfReader::Rbe) ;
dispatch.Store(String("RBE3"),&NastranBdfReader::Rbe) ;

dispatch.Store(String("SPC"),&NastranBdfReader::Constraint) ;
dispatch.Store(String("SPC*"),&NastranBdfReader::Constraint) ;
dispatch.Store(String("SPCD"),&NastranBdfReader::Constraint) ;
dispatch.Store(String("SPC1"),&NastranBdfReader::Constraint) ;
dispatch.Store(String("FORCE"),&NastranBdfReader::Force) ;
dispatch.Store(String("FORCE*"),&NastranBdfReader::Force) ;
dispatch.Store(String("FORCE1"),&NastranBdfReader::Force) ;
dispatch.Store(String("FORCE2"),&NastranBdfReader::Force) ;
dispatch.Store(String("MOMENT"),&NastranBdfReader::Moment) ;
dispatch.Store(String("MOMENT*"),&NastranBdfReader::Moment) ;
dispatch.Store(String("PLOAD"),&NastranBdfReader::Pressure) ;
dispatch.Store(String("PLOAD*"),&NastranBdfReader::Pressure) ;
dispatch.Store(String("PLOAD1"),&NastranBdfReader::Pressure) ;
dispatch.Store(String("PLOAD1*"),&NastranBdfReader::Pressure) ;
dispatch.Store(String("PLOAD2"),&NastranBdfReader::Pressure) ;
dispatch.Store(String("PLOAD2*"),&NastranBdfReader::Pressure) ;
dispatch.Store(String("PLOAD4"),&NastranBdfReader::Pressure) ;
dispatch.Store(String("PLOAD4*"),&NastranBdfReader::Pressure) ;
dispatch.Store(String("TEMP"),&NastranBdfReader::Temperature) ;
dispatch.Store(String("TEMP*"),&NastranBdfReader::Temperature) ;
dispatch.Store(String("TEMPERATURE"),&NastranBdfReader::Temperature) ;
dispatch.Store(String("TEMPD"),&NastranBdfReader::DefaultTemperature) ;

// parser initially splits at "("
dispatch.Store(String("TEMP(LOAD)"),&NastranBdfReader::Temperature) ;
dispatch.Store(String("TEMP(MATERIAL)"),&NastranBdfReader::Temperature) ;
dispatch.Store(String("TEMP(BOTH)"),&NastranBdfReader::Temperature) ;
dispatch.Store(String("TEMP(INITIAL)"),&NastranBdfReader::Temperature) ;

dispatch.Store(String("RFORCE"),&NastranBdfReader::RForce) ;
dispatch.Store(String("RFORCE*"),&NastranBdfReader::RForce) ;
dispatch.Store(String("GRAV"),&NastranBdfReader::Grav) ;
dispatch.Store(String("GRAV*"),&NastranBdfReader::Grav) ;
```

```
dispatch.Store(String("CORD1C"),&NastranBdfReader::CoordSys) ;
dispatch.Store(String("CORD1R"),&NastranBdfReader::CoordSys) ;
dispatch.Store(String("CORD2C"),&NastranBdfReader::CoordSys) ;
dispatch.Store(String("CORD2R"),&NastranBdfReader::CoordSys) ;
dispatch.Store(String("CORD1C*"),&NastranBdfReader::CoordSys) ;
dispatch.Store(String("CORD1R*"),&NastranBdfReader::CoordSys) ;
dispatch.Store(String("CORD2C*"),&NastranBdfReader::CoordSys) ;
dispatch.Store(String("CORD2R*"),&NastranBdfReader::CoordSys) ;
dispatch.Store(String("CORD2S"),&NastranBdfReader::CoordSys) ;
dispatch.Store(String("CORD2S*"),&NastranBdfReader::CoordSys) ;

dispatch.Store(String("PSOLID"),&NastranBdfReader::PSolid) ;
dispatch.Store(String("PSOLID*"),&NastranBdfReader::PSolid) ;
dispatch.Store(String("PBAR"),&NastranBdfReader::PSolid) ;
dispatch.Store(String("PBUSH"),&NastranBdfReader::PSolid) ;
dispatch.Store(String("PROD"),&NastranBdfReader::PSolid) ;
dispatch.Store(String("PSHELL"),&NastranBdfReader::PSolid) ;
dispatch.Store(String("MAT1"),&NastranBdfReader::Material) ;
dispatch.Store(String("MAT1*"),&NastranBdfReader::Material) ;
dispatch.Store(String("MAT4"),&NastranBdfReader::Material) ;
dispatch.Store(String("MAT4*"),&NastranBdfReader::Material) ;
dispatch.Store(String("MAT5"),&NastranBdfReader::Material) ;
dispatch.Store(String("MAT5*"),&NastranBdfReader::Material) ;
dispatch.Store(String("MAT9"),&NastranBdfReader::Material) ;
dispatch.Store(String("MAT9*"),&NastranBdfReader::Material) ;
dispatch.Store(String("MAT11"),&NastranBdfReader::Material) ;
dispatch.Store(String("MAT11*"),&NastranBdfReader::Material) ;
dispatch.Store(String("MATT1"),&NastranBdfReader::Material) ;
dispatch.Store(String("MATT1*"),&NastranBdfReader::Material) ;
dispatch.Store(String("MATT4"),&NastranBdfReader::Material) ;
dispatch.Store(String("MATT9"),&NastranBdfReader::Material) ;
dispatch.Store(String("MATT9*"),&NastranBdfReader::Material) ;
dispatch.Store(String("MATT11"),&NastranBdfReader::Material) ;
dispatch.Store(String("MATT11*"),&NastranBdfReader::Material) ;
dispatch.Store(String("TABLEM1"),&NastranBdfReader::Material) ;
dispatch.Store(String("TABLEM2"),&NastranBdfReader::Material) ;
dispatch.Store(String("TABLEM3"),&NastranBdfReader::Material) ;
//dispatch.Store(String("TABLEM1*"),&NastranBdfReader::Material) ;
dispatch.Store(String("TABLEM2*"),&NastranBdfReader::Material) ;
//dispatch.Store(String("TABLEM3*"),&NastranBdfReader::Material) ;

dispatch.Store(String("SET"),&NastranBdfReader::NodeSet) ;
dispatch.Store(String("SET1"),&NastranBdfReader::NodeSet) ;
dispatch.Store(String("SET3"),&NastranBdfReader::NodeSet) ;

dispatch.Store(String("LOAD"),&NastranBdfReader::Load) ;
```

```
dispatch.Store(String("SPCADD"),&NastranBdfReader::SpcAdd) ;

dispatch.Store(String("BOLT"),&NastranBdfReader::Bolt) ;
dispatch.Store(String("BOLTFOR"),&NastranBdfReader::BoltForce) ;
dispatch.Store(String("BOLTLD"),&NastranBdfReader::BoltLoadControl) ;

dispatch.Store(String("BCRPARA"),&NastranBdfReader::NXContactParams) ;
dispatch.Store(String("BSURFS"),&NastranBdfReader::NXContactSurfs) ;
dispatch.Store(String("BSURF"),&NastranBdfReader::NXContactSurfs) ;
dispatch.Store(String("BCSET"),&NastranBdfReader::NXContactSets) ;
dispatch.Store(String("BCTSET"),&NastranBdfReader::NXContactSet) ;
dispatch.Store(String("BCTADD"),&NastranBdfReader::NXContactAdd) ;
dispatch.Store(String("BGPARM"),&NastranBdfReader::NXGlueParams) ;
dispatch.Store(String("BGSET"),&NastranBdfReader::NXGlueSets) ;
dispatch.Store(String("BGADD"),&NastranBdfReader::NXContactAdd) ;


dispatch.Store(String("OMIT"),&NastranBdfReader::Omit) ;
dispatch.Store(String("ENDDATA"),&NastranBdfReader::End) ;
```