NASA/TM-20250007388

User Manual for TURBO Displacement File Generator (TDFG)

Kristopher C. Pierson
Glenn Research Center, Cleveland, Ohio

August 2025

NASA STI Program Report Series

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION.
  Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM.
  Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT.
  Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION.
  Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION.
  Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION.
  English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

NASA/TM-20250007388

User Manual for TURBO Displacement File Generator (TDFG)

Kristopher C. Pierson
Glenn Research Center, Cleveland, Ohio

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

August 2025

Acknowledgments

The author acknowledges the Transformational Tools and Technologies (TTT) project of NASA's Transformative Aeronautics Concepts Program (TACP) for funding this research, as well as Milind Bakhle and Gregory Heinlein for sharing their knowledge of TURBO, which greatly contributed to the creation of TURBO Displacement File Generator.

This work was sponsored by the
Transformative Aeronautics Concepts Program.

Level of Review: This material has been technically reviewed by technical management.

This report is available in electronic form at https://www.sti.nasa.gov/ and https://ntrs.nasa.gov/

NASA STI Program/Mail Stop 050
NASA Langley Research Center
Hampton, VA 23681-2199

Contents

User Manual for TURBO Displacement File Generator (TDFG)

Kristopher C. Pierson
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Summary

The TURBO Displacement File Generator (TDFG) is a Python-based graphical user interface designed to efficiently generate grid displacement files for the computational fluid dynamics (CFD) program TURBO. TDFG is intended to be used in conjunction with the Geometry Alignment and Mode Mapping Application (GAMMA). First, GAMMA maps structural mode shapes from finite element analysis onto CFD surface meshes. Subsequently, TDFG generates displacement fields that extend these surface displacements throughout the entire CFD computational domain. The resulting grid displacement files produced by TDFG enable one-way or two-way fluid-structure interaction analyses using TURBO. This user manual provides comprehensive instructions on the installation and effective use of TDFG.

1.0 Introduction

The TURBO Displacement File Generator (TDFG) is a Python-based tool that is used for generating displacement files to run a turbomachinery aeromechanics analysis in TURBO (Ref. 1). The program is configured to work with files output from the Geometry Alignment and Mode Mapping Application (GAMMA) (Refs. 2 and 3), which is used to map mode shapes from the finite element analysis solution to the blade surface in the computational fluid dynamics (CFD) domain.

The instructions in this manual assume that the user's system satisfies the following two requirements:

- The user's system is a Linux system.
- The user can install Miniconda to setup the Python environment.

Section 2.0 contains information for the setup of TDFG, and Section 3.0 contains instructions on its usage.

2.0 TURBO Displacement File Generator Setup

2.1 System Compatibility

Instructions in this document are for users with a Linux workstation. Though it is possible to run TDFG on a Windows or Apple operating system, it has not been tested and is not recommended. This is mainly because GAMMA must operate in a Linux environment, and the files to start the process in TDFG must be output from GAMMA. It is recommended to run both applications in the same environment.

2.2 Cloning TDFG From Git

TDFG is available through its project page (Ref. 4) on the Glenn Research Center (GRC) GitLab. First, the user must make certain that they have access to the GRC GitLab through the NASA Access

Management System (NAMS). At the time of this writing, the NAMS request is called 'GRC GitLab Enterprise Edition' with ID 268639. A URL for this request is included here:

- https://idmax.nasa.gov/nams/asset/268639/

The TDFG GitLab project is open to anyone with access to GRC GitLab. The next step in the process is to make certain that Git is installed on the local system. If Git is not on the system, the following command will install it on a Debian-based system such as Ubuntu.

- `sudo apt install git`

To install Git onto a Red Hat Enterprise Linux (RHEL) based system such as RHEL or Oracle Linux, one of the following two commands will work. The `dnf` command is often used in newer systems.

- `sudo yum install git`
- `sudo dnf install git`

Now, create a folder in which the TDFG program files will reside and navigate to that folder. Then, enter the following command to clone the Git repository:

- `git clone https://gitlab.grc.nasa.gov/kcpierso/turbo-displacement-file-generator`

2.3 Python Environment

Instructions for setting up the Python environment using a utility called pyenv (Ref. 5) are included in the user manual for GAMMA (Ref. 2 and 3). The pyenv installation method works well when installing on a Debian Linux system such as Ubuntu. So, users of Ubuntu or other types of Debian Linux should follow the instructions in the GAMMA User Manual in Reference 3.

However, it has recently been observed that an environment created using Miniforge (Ref. 6) is more robust when installing on a RHEL-based system. Miniforge is a community-driven version of the Conda package manager. Miniforge is a convenient solution because it does not require a license for use, in contrast to the version of Conda provided by Anaconda Inc., which requires a license to use the Anaconda repository. The setup using the Miniforge version of Conda is also simpler compared to the pyenv setup and allows the Python environment to be reproduced exactly.

To install Conda, it is recommended that the user visit the Conda GitHub page:

- https://github.com/conda-forge/miniforge

Then, scroll down to the section labeled 'Unix-like platforms (macOS, Linux, & WSL)' and follow the installation instructions. If additional help is needed, artificial intelligence (AI) systems such as ChatGPT, Gemini, or Claude can provide Miniforge installation instructions accurately, as Miniforge is a very commonly used tool in software development.

After Miniforge is installed, the Python environment for TDFG can be created using the ModeEnv.yml file. This file is included with the TDFG program files in the directory labeled 'Installation'. To create the environment, run the following command from inside the 'Installation' directory:

- `conda env create -f ModeEnv.yml`

After the environment setup is complete, activate the newly created environment, named ‘ModeEnv’, by using the following command:

- `conda activate ModeEnv`

3.0 Turbo Displacement File Generator Usage

3.1 Starting TDFG

At this point, the program files have been downloaded and the environment is set such that the user can start TDFG. To start TDFG, ensure that the Conda environment is active (refer to the activation command in Section 2.3), then run the command shown below, replacing `/path/to/tdfg/directory/` with the actual path to the TDFG program files directory:

- `python /path/to/tdfg/directory/tdfg.py`

3.2 Surface Selection Interface

Upon startup, the first tab of TDFG (the surface selection interface) is displayed as shown in Figure 1. Initially, this interface has only three active buttons:

- Browse
- Load Blocks
- Load Data (discussed in Section 3.3)

The ‘Browse’ and ‘Load Blocks’ buttons allow the user to start a new process with their chosen geometry. The ‘Load Data’ button allows the user to load previously saved data.

3.2.1 Loading Turbo Block Files

Before loading the TURBO grid files, please note the following:

- The ‘Load Blocks’ action can only be performed once, so the grid block files must all be in the same directory.
- The grid block files must retain the naming convention expected by TURBO. If the file names have been changed, TDFG cannot recognize them as TURBO grid block files.

To load the TURBO grid block files, first select the ‘Browse’ button, then use the file browser to select the directory in which the files are stored. After the directory is chosen, its path is displayed on the line labeled ‘Block Directory’. Next, click the ‘Load Blocks’ button. The list area labeled ‘List of Blocks’ will then be filled with the block file names from the directory that was selected using the ‘Browse’ button.

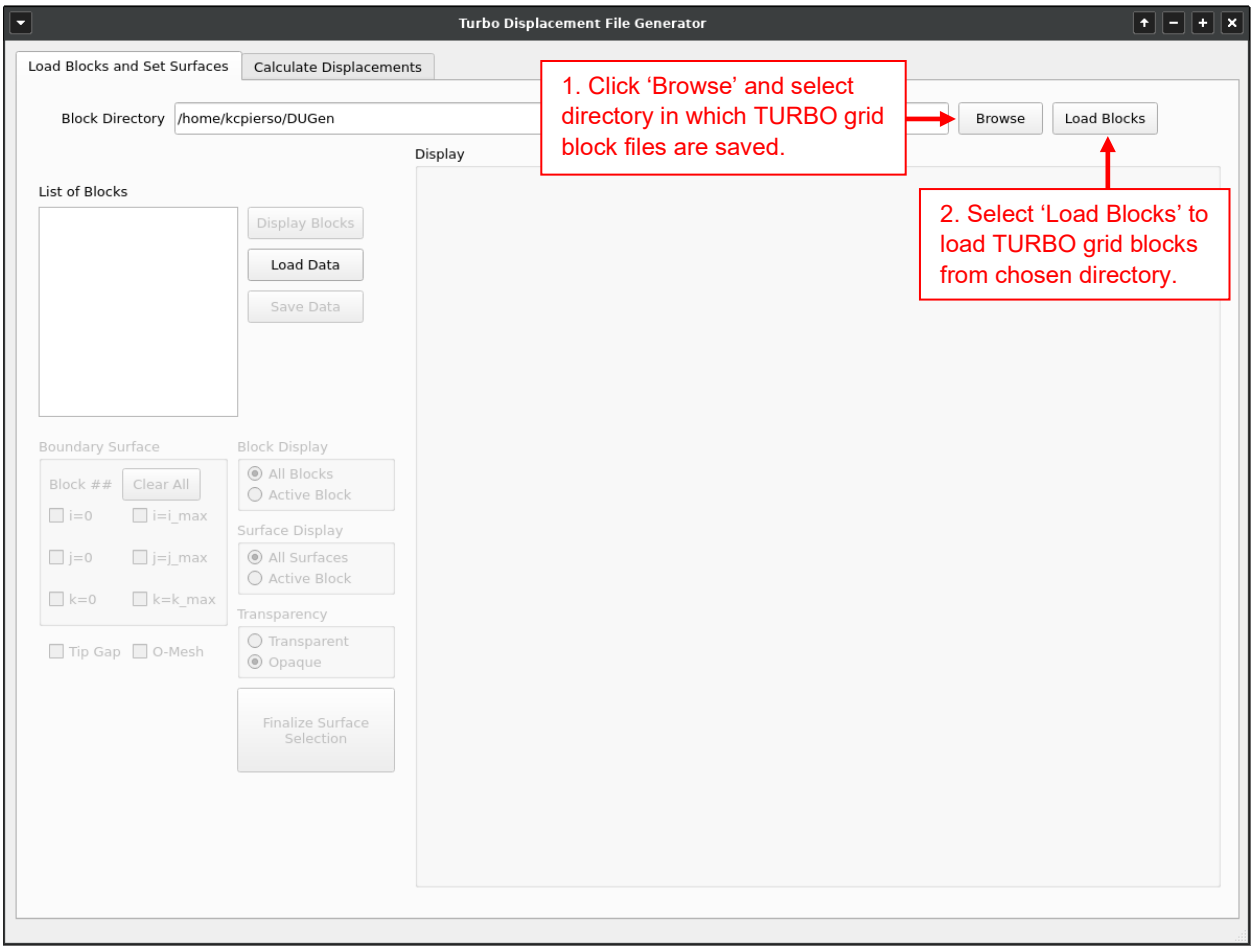

Figure 1.—TDFG interface upon startup.

3.2.2 Displaying Grid Blocks

To display the grid blocks, simply click the 'Display Blocks' button, as shown in Figure 2. If the default block display options have not been changed, every block is displayed, with the active block colored orange in the display area. To change the active block, select a different one from the list using left click on the mouse.

The graphics display window has the following controls using the mouse:

- Rotate: Hold left button and move the mouse.
- Pan: Hold middle button and move the mouse.
- Zoom: Use mouse scroll wheel or hold right button and move the mouse up and down.

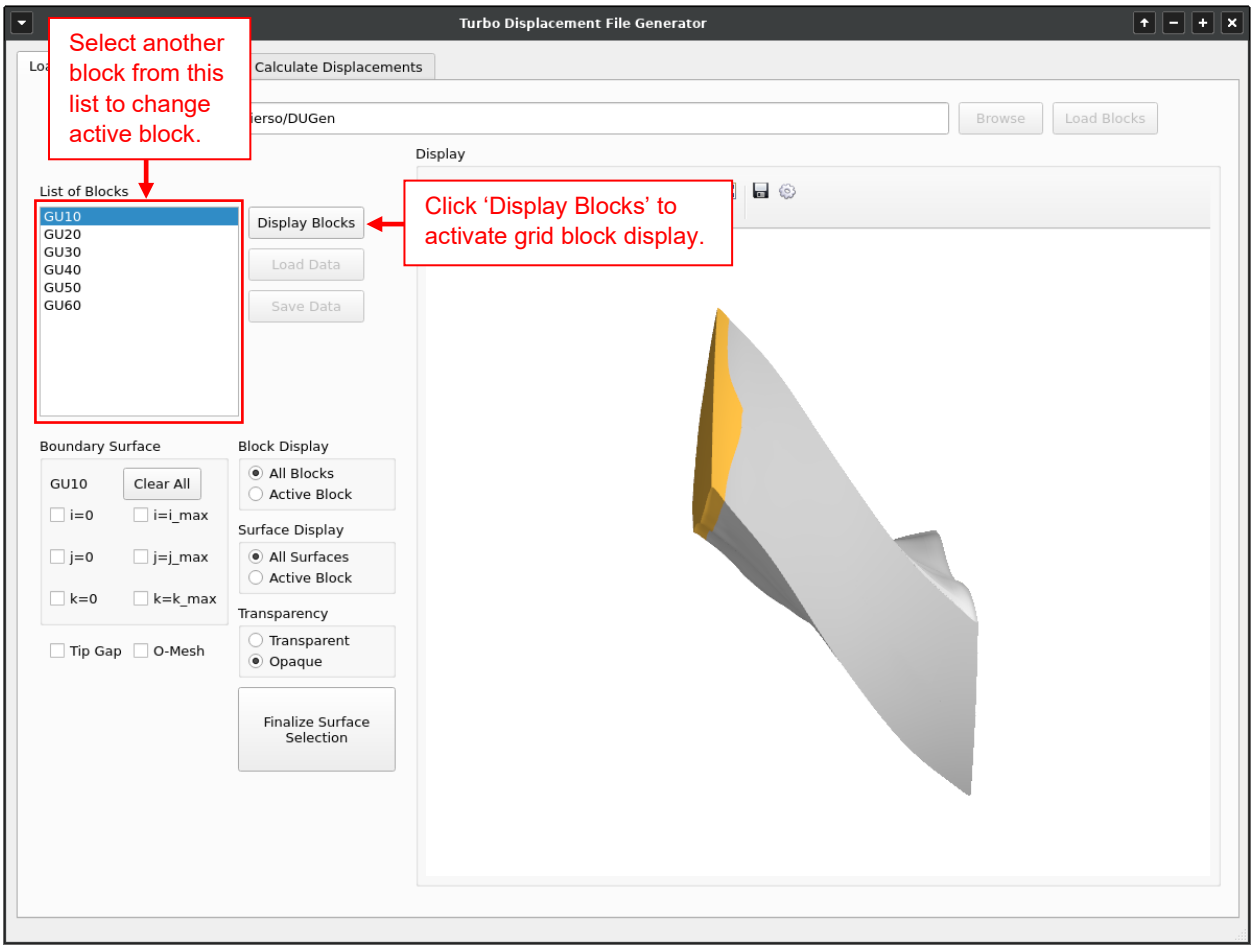

Figure 2.—To display blocks, select button as shown in this figure. The active block is highlighted on the list and colored entirely orange in display.

3.2.3 Block Display Settings

TDFG has three display options, shown in Figure 3, that have been developed to make the surface selection process more user-friendly. The options in the box labeled 'Block Display' allow the user to choose between showing all loaded blocks or only the active block. Showing all loaded blocks allows the user to see surfaces that would otherwise be hidden. Typically, the user will not want to select any interior surfaces; this option simply allows the user to be sure about their selection.

The 'Surface Display' options allow the user to choose between showing selected surfaces from the active block only or showing selected surfaces from all blocks. Retaining selected surfaces from all blocks (Surface Display: All Surfaces) while showing only the active block (Block Display: Active Block) is a good strategy, as this display can make it easier for the user to determine which surfaces are the outer surfaces.

The 'Transparency' options may not work on all systems. Notably, if using TDFG through virtual network computing (VNC), it is likely that the graphics will not render properly. This display option is not needed, but it can be useful to make sure that no internal surfaces have been selected.

The new user should practice using the various domain display options and develop a visualization strategy that works well for them.

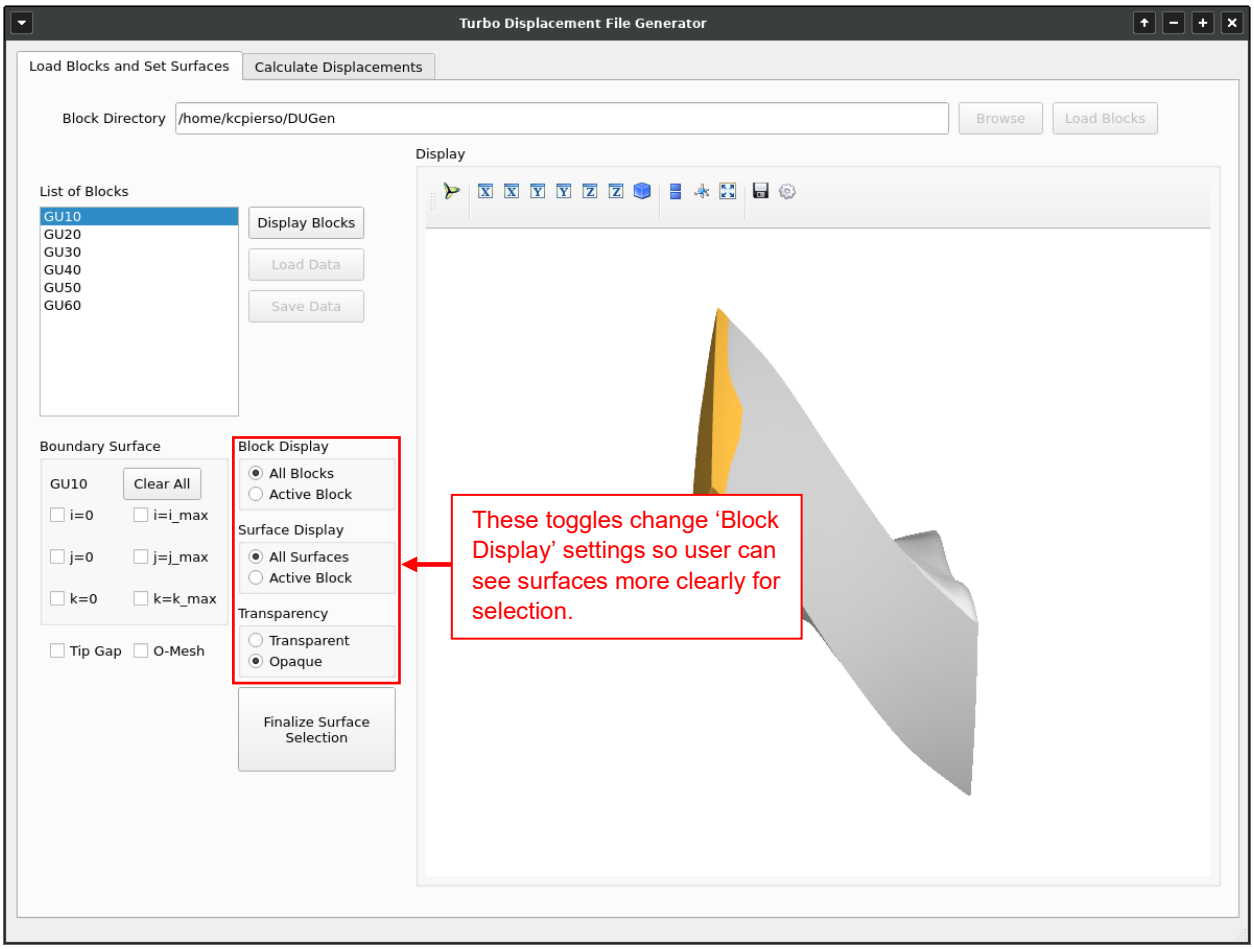

Figure 3.—TDFG has three toggleable block display options, shown in red box.

3.2.4 Selecting Exterior Surfaces

Besides loading the data, the main purpose of the functions on the 'Load Blocks and Set Surfaces' tab is to set the exterior boundary surfaces. The 'Boundary Surface' selection options are shown in Figure 4. The best recommended practice is to omit the hub and casing from the boundary surface selection. Though these can be included, the decay of motion that is generated by omitting the hub and casing boundaries is preferred. Also, the user should not set any surfaces from the O-mesh or the tip gap; these have separate selections created specifically for those blocks.

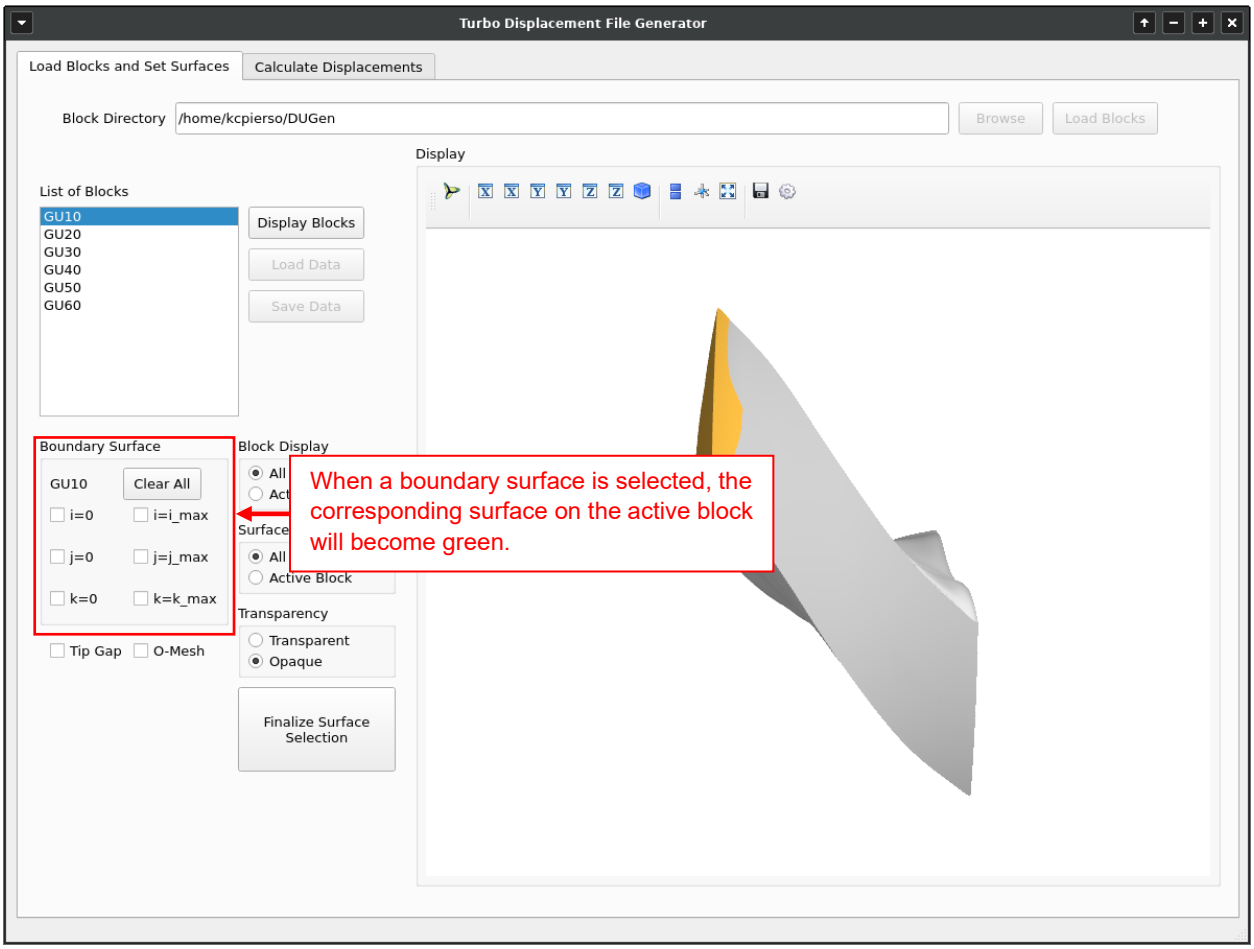

Figure 4.—Boundary surface selection box is used to select external boundary surfaces.

Figure 5 shows an example of surface selection. In this example, all but one of the exterior surfaces have been selected. The figure also shows the usefulness of displaying only the active block while still displaying all selected surfaces. This makes it easier to avoid accidentally selecting internal surfaces. Also, seeing the already selected surfaces should make it easier to determine which surface on the active block is actually the exterior surface.

The checkboxes in the ‘Boundary Surface’ box enable the user to select surfaces corresponding to the i-, j-, or k-indices of the grid. Since boundaries only occur on the maximum or minimum i-, j-, or k-planes, these are the available options for selection. Simply click the checkbox to mark it as a boundary surface, and it will become green in the graphics display.

The proper surface selections for the example case that is provided in the Example folder from the TDFG Gitlab are included in Table 1.

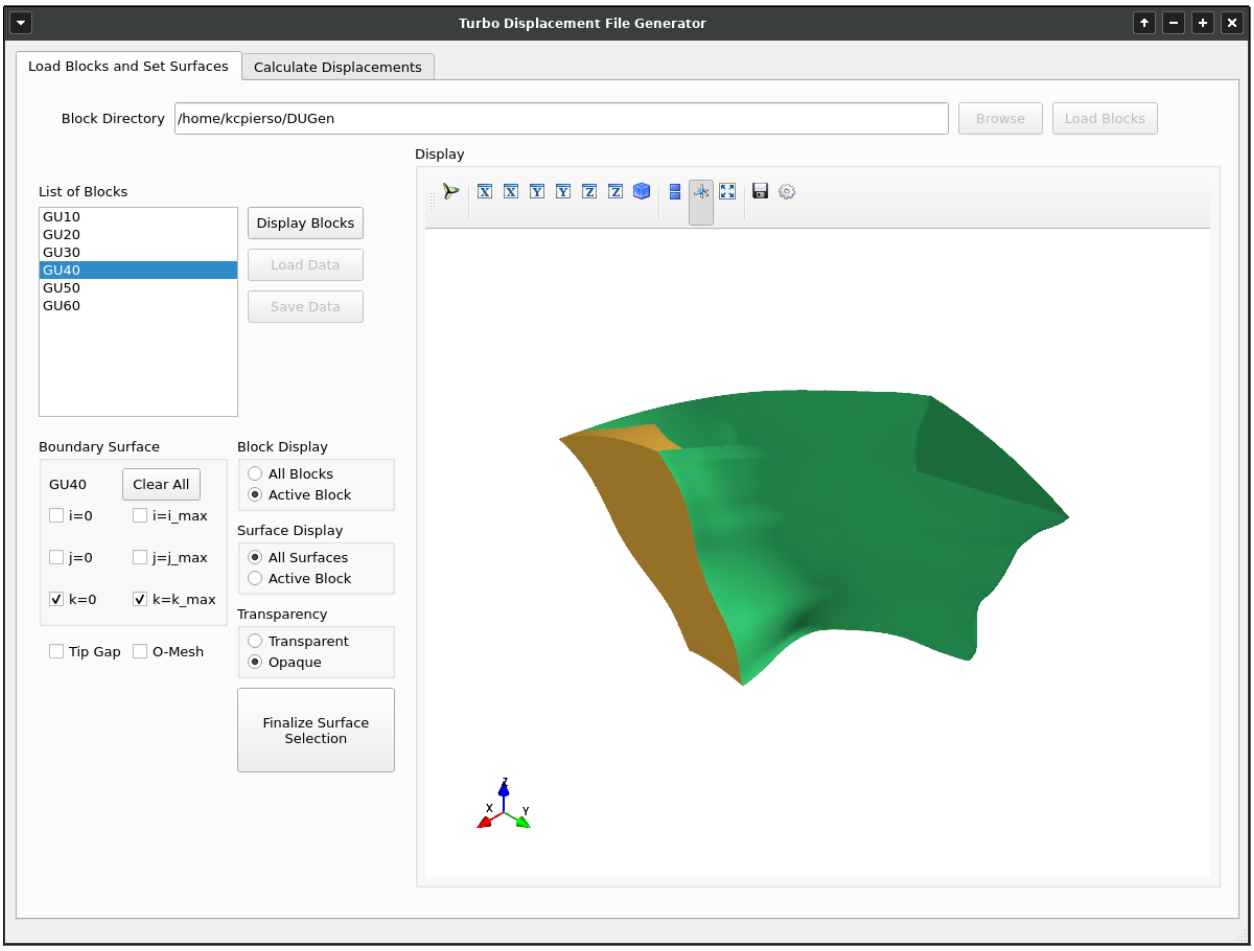

Figure 5.—Boundary surface selection example using recommended viewing settings: display only active block and display all selected surfaces.

TABLE 1.—SURFACE SELECTIONS FOR TUTORIAL CASE IN TDFG GITLAB EXAMPLE FOLDER

Grid block	Surface selection
GU10	i=0, k=0, k=k_max
GU20	k=k_max
GU30	k=0
GU40	i=i_max, k=0, k=k_max

3.2.5 Selecting Tip Gap and O-Mesh

The selection of the tip gap and the O-mesh blocks is required for the displacement decay algorithm to work. These blocks are treated as special zones during the creation of the displacement field. The algorithm requires there to be only one tip gap block and one O-mesh block. To mark the appropriate block for selection, simply set it as the active block and click the appropriate checkbox. Figure 6 indicates where these selection options are in the interface. Please note that no surface selection can be made for the tip gap and O-mesh blocks. The algorithm assumes that the blade surface is in the first k-index and

that the tip gap *j*-indices move from the tip of the blade to the casing. In the tip gap, the blade displacement motion linearly decays from the blade tip to the casing. The proper selection of O-mesh and tip gap for the example case are included in Table 2.

TABLE 2.—O-MESH AND TIP GAP SELECTION FOR CASE IN EXAMPLE FOLDER

Grid block	Selection
GU50	O-mesh
GU60	Tip gap

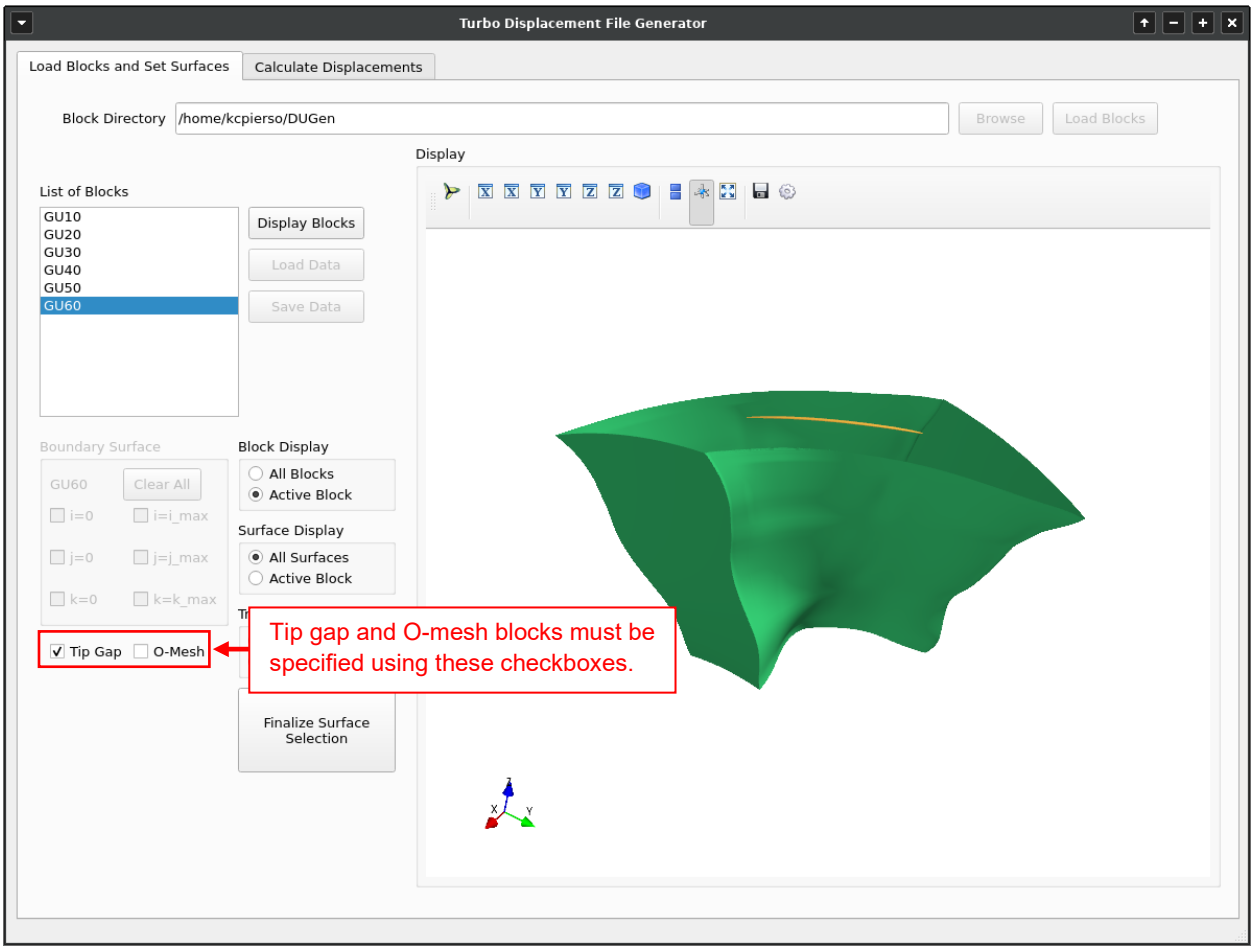

Figure 6.—Tip Gap and O-Mesh checkboxes are located directly below surface selection.

3.2.6 Finalizing Surface Selection

After completing the task of marking the surfaces and blocks, the next step is to finalize the selection. Finalizing the surface selection must be done before the remaining functions of TDFG can be accessed. After finalization, changes cannot be made to the surface and block selection. The user can still view previous selections by moving through the list of blocks, but the selection functions are disabled.

To finalize the selection, click the ‘Finalize Surface Selection’ button, which is indicated in Figure 7. After selection finalization, it is recommended to immediately save the case to avoid going through the surface selection process again. Simply click the ‘Save Data’ button that is now enabled, and a file dialog will appear that allows the user to save the case file. The case file uses ‘mod’ as the file extension.

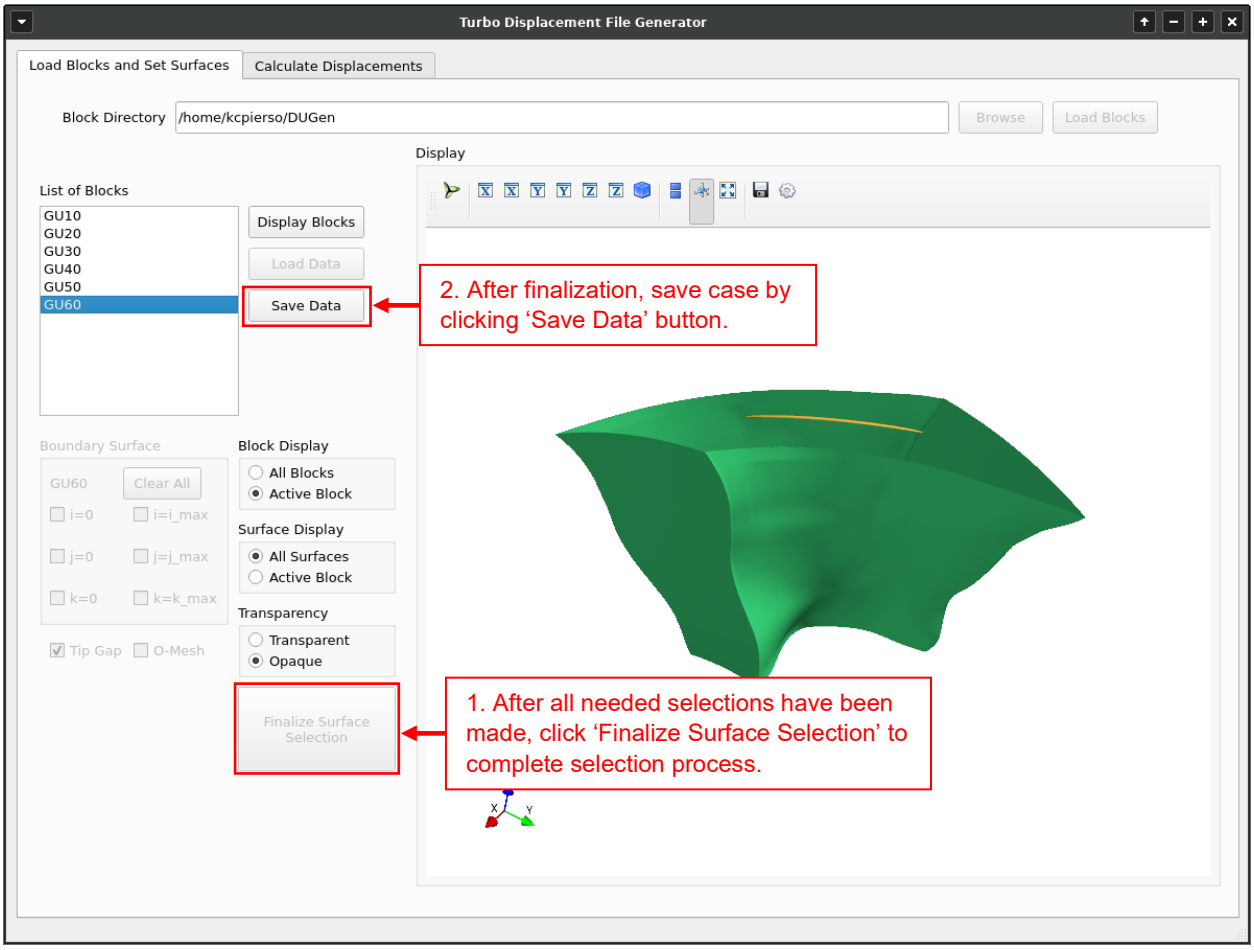

Figure 7.—Finalize surface selection after appropriately marking all surface and blocks.

3.3 Loading Case File

The ability to save cases, described in Section 3.2.6, can help the user avoid repeating the somewhat tedious block and surface selection process. The saved data can only be loaded as the first step after launching the graphical user interface (GUI). If the user loads blocks after launching the GUI, the ‘Load Data’ option becomes disabled.

To load a saved case, click on the ‘Load Data’ button, shown in Figure 8. A file browse dialog then appears that can be used to select the case file. After the file is selected using the browse dialog, it is automatically loaded.

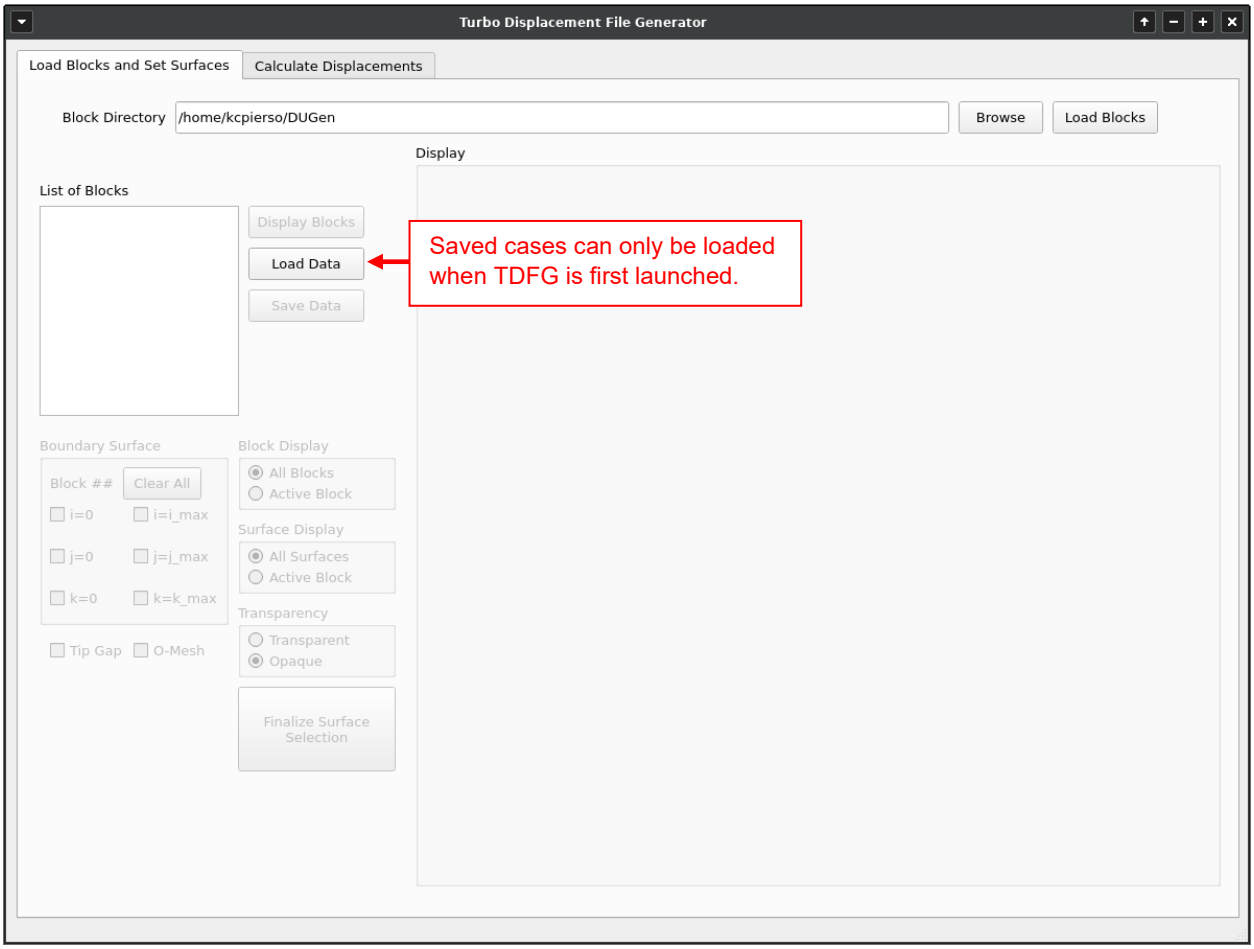

Figure 8.—‘Load Data’ button is active only upon a fresh launch of the GUI.

3.4 Generating Grid Displacement Fields

To generate grid displacement fields, the user must have blade surface displacements already defined using GAMMA (Ref. 2). For real mode shapes, GAMMA outputs a single file; for complex mode shapes, it outputs two files. Please refer to the GAMMA manual for more information on the file output from GAMMA.

3.4.1 Determining Node Distance

The ‘Determine Node Distance’ button is shown in Figure 9. Determination of the nodal distance is a simple step for the user but a computationally intensive one for TDFG. In this step, the distance of interior nodes from the blade surface and the boundary are determined. This process must loop through every node in the domain, so the computing resources needed to do this increase with mesh size. The exact length of time this step takes will therefore vary depending on the mesh, but the user should expect this step to take several minutes. During this process, information on the progress of the operation prints to the terminal from which TDFG was launched. After this step, it is highly recommended that the user save the case so that this lengthy process does not need to be repeated.

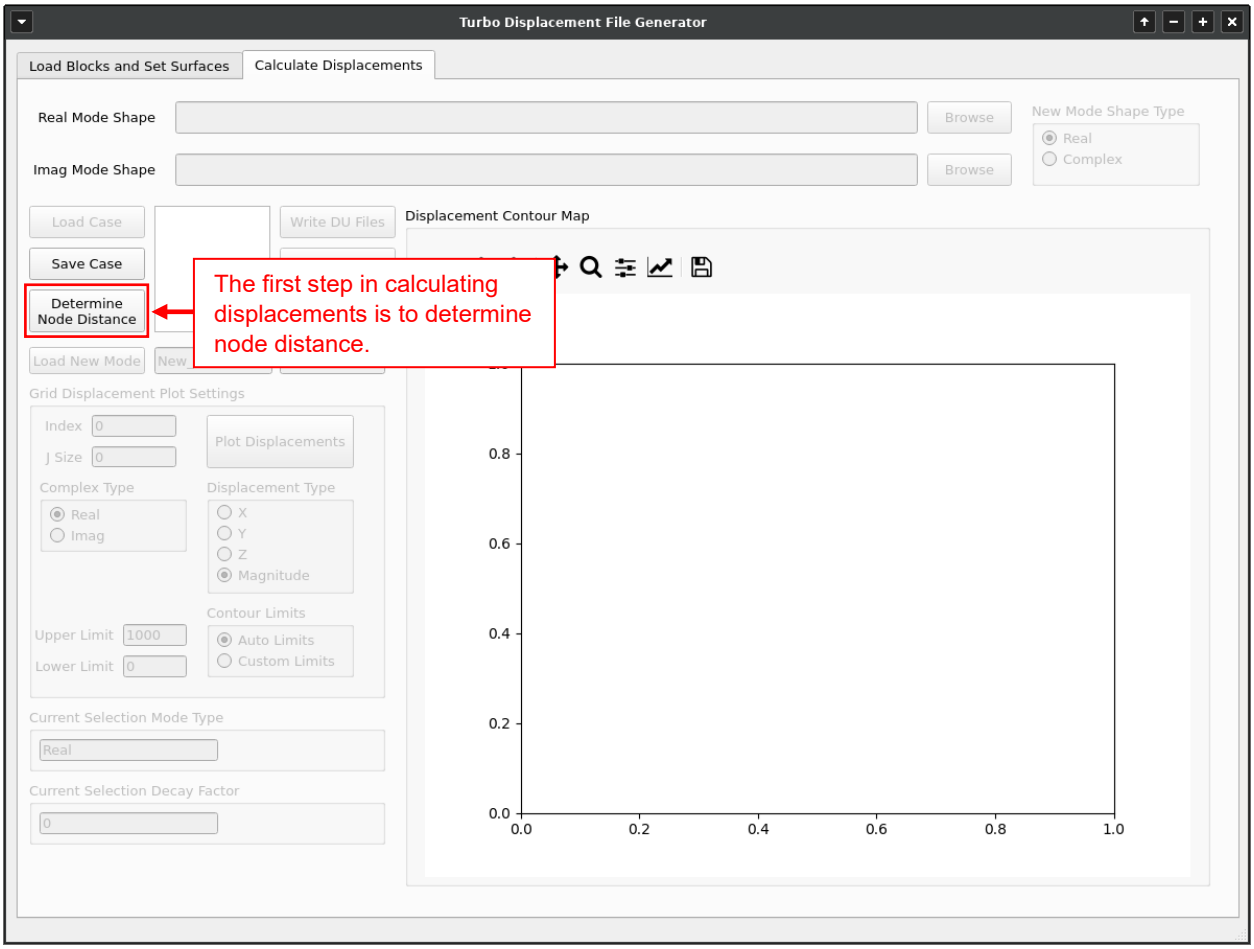

Figure 9.—Before calculating displacement fields, node distance must be determined. This process is computationally intensive and may take several minutes or more.

3.4.2 Loading Blade Surface Modal Deflections

To ensure compatibility with TDFG, the user should use GAMMA to generate surface deflection profiles. Figure 10 illustrates the steps for loading a modal deflection file. The first step for loading the blade surface displacements for the mode shape is to choose the type of mode shape, real or complex. Next, the user should load the real mode shape file. If the mode shape is complex and the GAMMA naming convention has not been altered, TDFG will automatically load the imaginary modal displacement file. If the names have been changed, then the user will likely need to use the 'Browse' button directly below the one highlighted in Figure 10 and select the imaginary file.

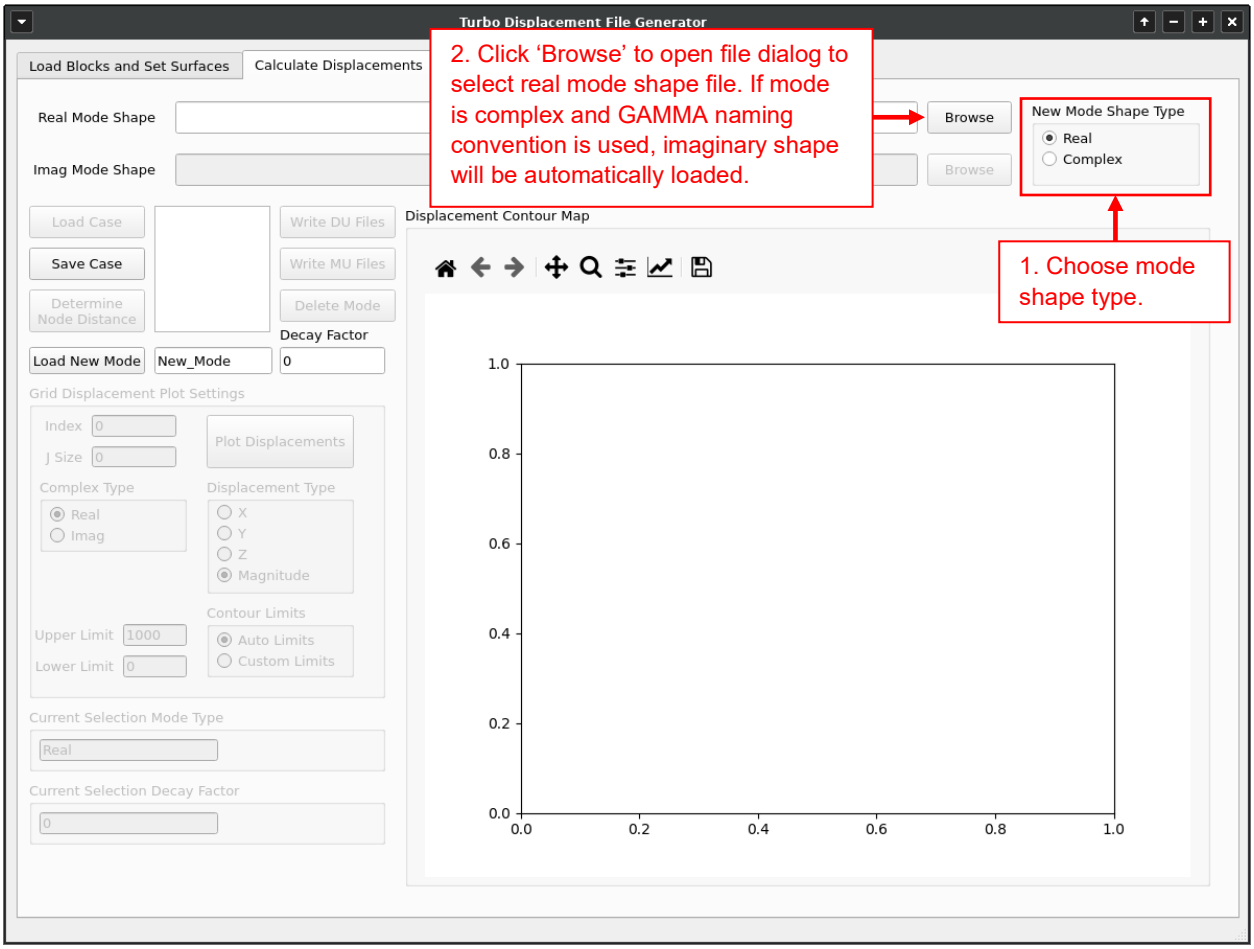

Figure 10.—TDFG process to load real or complex mode shape created by GAMMA.

Before loading the mode shape, the user should first set the decay factor and the name for the mode; these input fields are shown in Figure 11. The decay factor is set to zero by default, which results in a linear decay from the edge of the O-mesh to the boundary surfaces selected in Section 3.2.4. No decay occurs within the O-mesh in order to avoid issues with the wall model used in turbulence modeling. This is needed because the cell-centered distance from the wall does not update during computations, so it should not deviate from its original value for cells in which the wall model is active (near wall cells). Displacement decay near the blade boundary could squeeze or expand near wall cells, changing the distance of their cell centers and violating the original distance calculated when TURBO begins the simulation.

Setting the decay factor to something other than zero results in a bias of the decay toward either the blade or the boundary. Section 3.4.3 explains how the decay factor is used to determine the displacements in the domain.

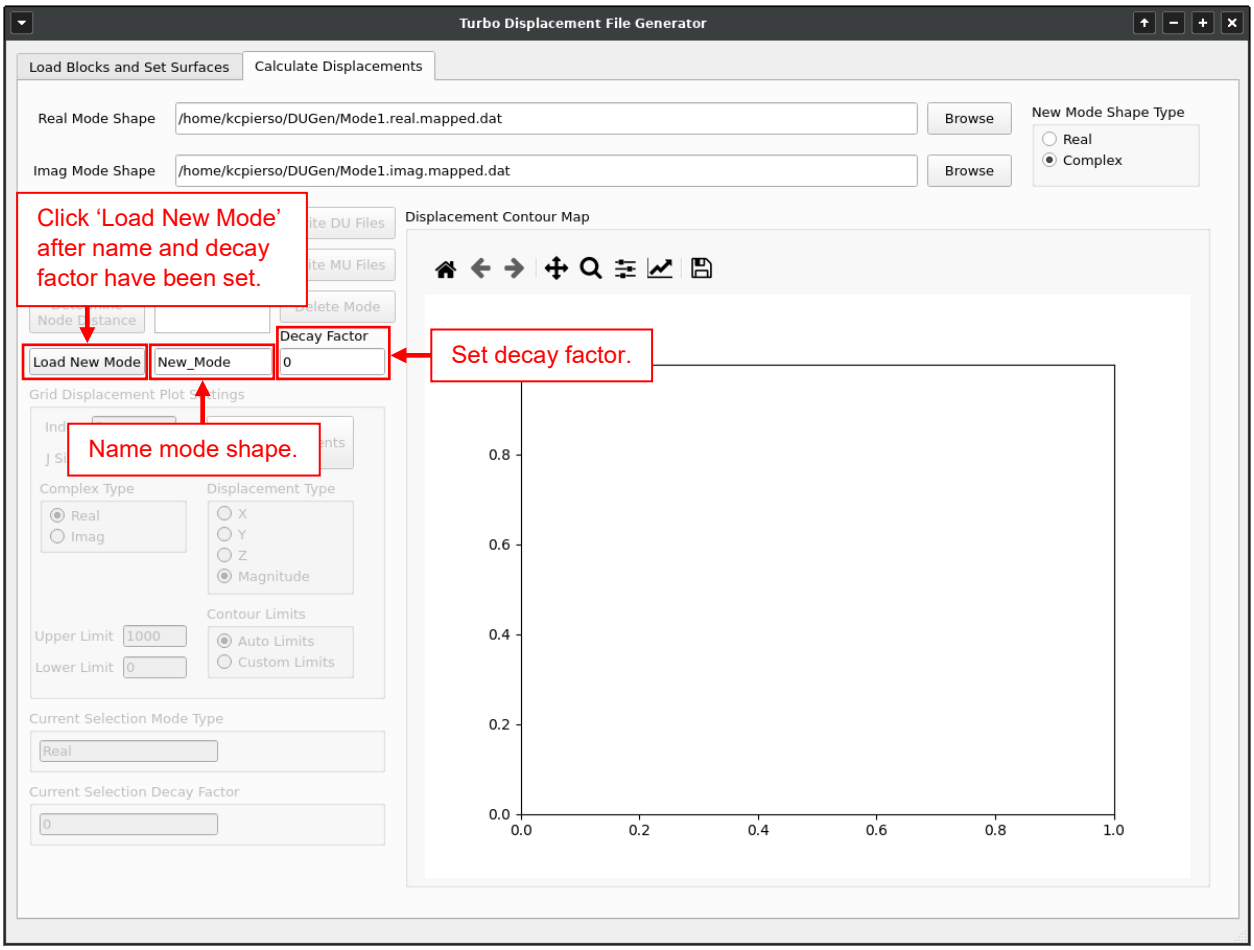

Figure 11.—Buttons and inputs for creating new displacement field for particular modal deflection file.

3.4.3 Decay Factor

As the surface of the blade deforms, it is also necessary to deform the other nodes within the domain. The decay factor controls the rate at which the blade motion decays. An algorithm developed by Akiva Wernick (Ref. 7) is used to determine the motion within the CFD domain. For each node in the domain, a location parameter, P, is defined using the distance to the closest O-mesh node, d_{bld}, and boundary node, d_{bnd}:

$$P = \frac{d_{bld}}{d_{bld} + d_{bnd}} \tag{1}$$

The displacement scale factor, D_f, is determined at each mesh point using the decay factor, λ, and the P value in Equation (2) or Equation (3). If $\lambda = 0$, D_f is simply equal to P as shown in Equation (2). This results in a linear decay from the outer surface of the O-mesh to the domain boundary selected in Section 3.2.4.

$$\text{if } \lambda = 0, \text{then } D_f(P) = P \tag{2}$$

For all other values of λ, Equation (3) is used to determine D_f. Left of the arrow symbol in Equation (3), P_{bnd} is P at the passage boundary and P_{bld} is P at the O-mesh surface. In Equation (3), please note that $P = 0$ on the O-mesh surface and $P = 1$ on the passage boundary. Given the known value of P at the O-mesh surface and passage boundary, Equation (3) can be simplified to the form on the right-hand side of the arrow symbol.

$$\text{if } \lambda \neq 0, \text{then } D_f(P) = \frac{e^{-\lambda P} - e^{-\lambda P_{bnd}}}{e^{-\lambda P_{bld}} - e^{-\lambda P_{bnd}}} \rightarrow D_f(P) = \frac{e^{-\lambda P} - e^{-\lambda}}{1 - e^{-\lambda}} \tag{3}$$

Figure 12 shows a plot of D_f computed with various values of λ for P ranging from 0 to 1, which is the full range of P that will exist in any given domain. The figure shows that a positive decay factor causes D_f to fall toward zero more quickly, whereas a negative decay factor will delay the decrease of D_f. As of the writing of this manual, linear decay is the recommended setting, which can be used by retaining the default value of 0 for the decay factor (λ) in the TDFG interface (see Figure 11). Decay factors other than 0 may be an interesting research topic.

The displacement scale factor works with the displacements on the surface of the O-mesh to determine the motion of individual nodes. For example, the x-displacement of a particular node, Δx_{ijk}, is determined first by calculating D_f for that point and then by assigning a Δx_{bld}, which is the x-displacement of the nearest node on O-mesh surface.

$$\Delta x_{ijk} = D_f(P)\Delta x_{bld} \tag{4}$$

The mesh deformation is also determined for Δy_{ijk} and Δz_{ijk} in the same way as Equation (4).

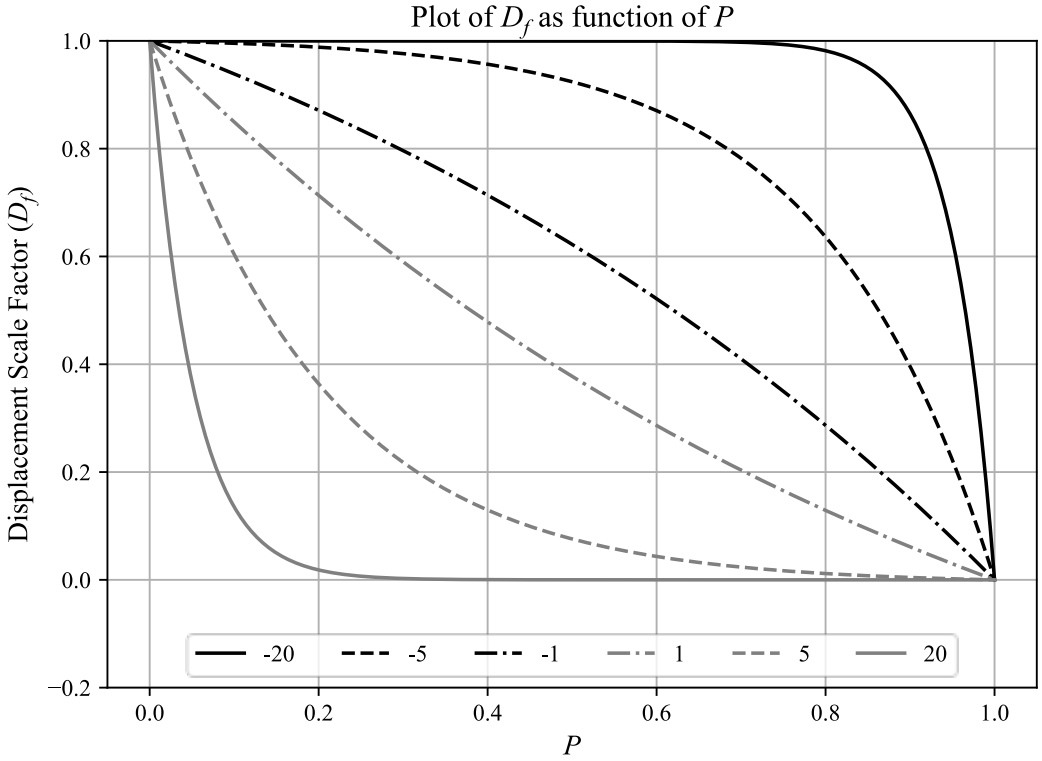

Figure 12.—Plot of displacement scale factor as function of P.

3.4.4 Examining Displacement Fields

After loading the mode, the user can examine the displacement field using the plotting options indicated in Figure 13. The user has the option to display the displacement magnitude or one of the X, Y, or Z components. The plot tool shows the displacement field as a 2D contour of a particular j-index. The user sets the j-index in the box next to the label ‘Index’. The limits of the contour plot can either be specified by the user or automatically set. If automatically set, the limits are based on the range for the entire domain rather than the range for only the current j-index. The decay factor used to generate the displacement field and the type of mode (real or complex) is also displayed as indicated in Figure 13.

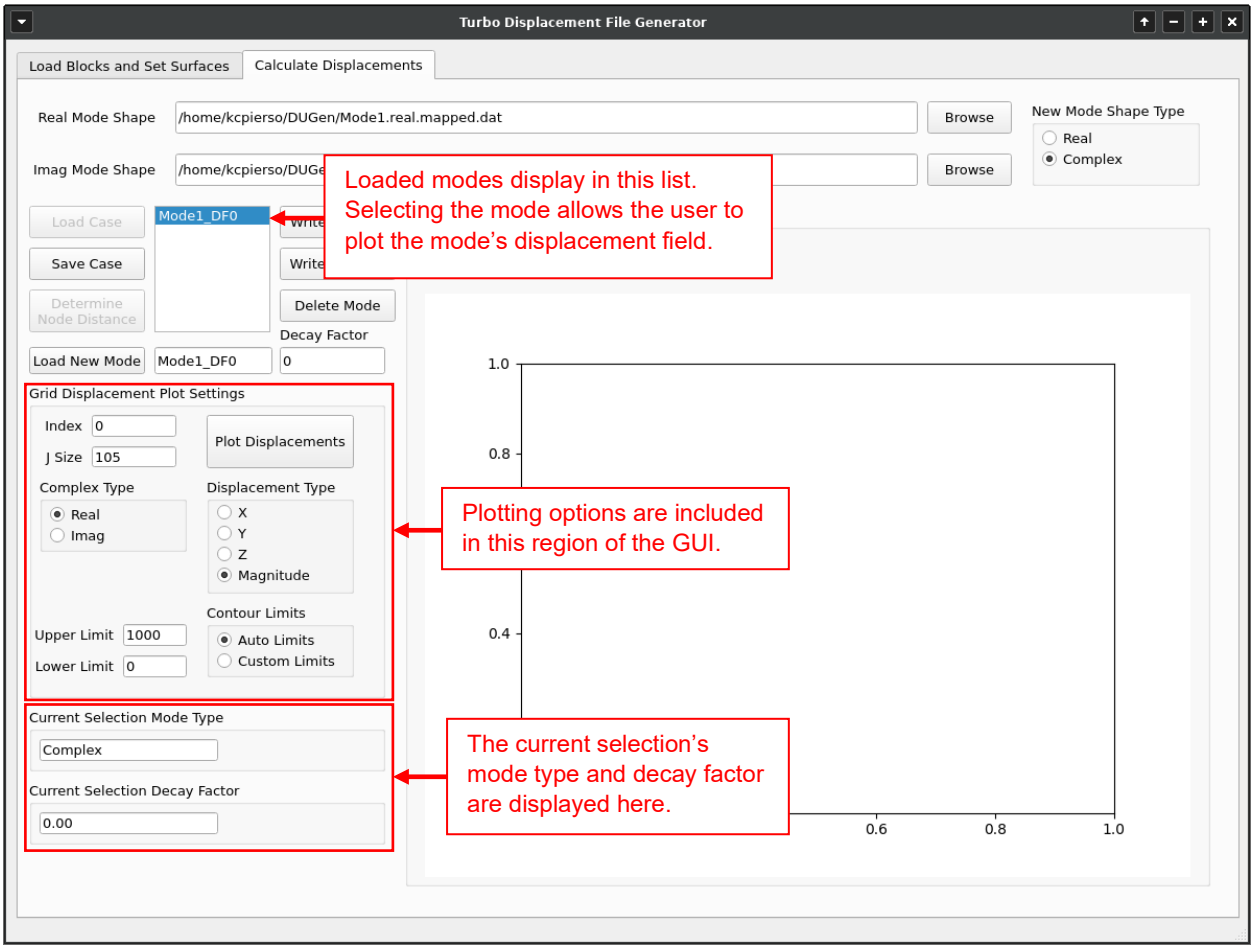

Figure 13.—Loaded mode selection and plotting options.

An example of displaying the displacement field at a particular j-index is shown in Figure 14. The number of j-indices in the domain is indicated just below the location of the input of j-index for plotting. Note that the number of j-indices in the tip gap is much less than in the rest of the domain. The plotting utility accounts for the differences in the j-indices in the tip gap and automatically adds the tip-gap region in areas where it lies.

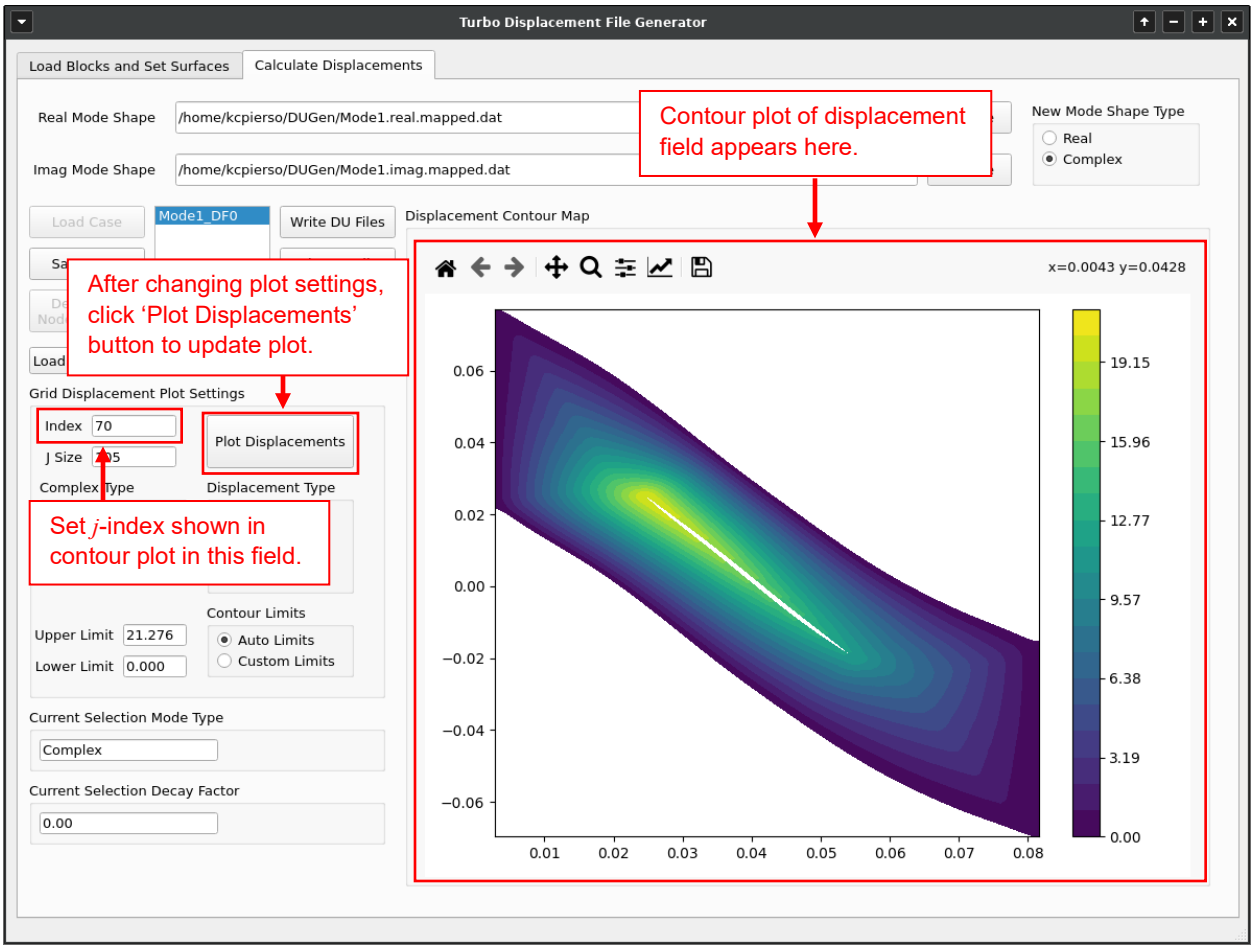

Figure 14.—Example showing contour plot of displacement field at j-index of 70.

3.4.5 Outputting Displacement Field

TURBO can read two different types of files, DU and MU, as the definition of the displacement field. The DU file type is the traditional form that is defined in Cartesian coordinates. The MU file type was developed for use with two-way modal coupling. For MU files, the deflection is defined in terms of radial and tangential components instead of y and z coordinates (Note: x-displacement is defined identically between the two formats). The relationship between the two Cartesian deflections (dy, dz) and radial deflections (dr) or tangential deflections (dt) is defined in Equations (5) and (6), respectively. The value of θ is determined by the grid node's θ value. Please note that TURBO grid files are defined in a cylindrical coordinate system, so the θ value at the grid points is readily available.

$$dr = dz \cdot \cos(\theta) + dy \cdot \sin(\theta) \tag{5}$$

$$dt = -dz \cdot \sin(\theta) + dy \cdot \cos(\theta) \tag{6}$$

For the MU files, grid displacement is defined in terms of dr, dt, and dx, which allows all rotor passages to be represented using data from just a single passage of grid displacement files, even when simulating the full wheel. With the MU files, TURBO also offers the capability to set a nodal diameter

pattern. The MU files are the only type of grid displacement file that can be used for two-way modal coupling. With all of these advantages, it is recommended that the user generally work with MU files. However, please note that many more simulations have been conducted using DU files as compared to MU files, so if the user has a problem with the MU files, it is recommended that they revert to DU files.

To output the grid displacement files, simply click the button corresponding to the desired file type, as shown in Figure 15. The files are then output to a new directory that is named using the mode name. This new directory is created in either the directory from which the blocks were loaded or the directory from which the saved case was loaded. The path of the output file prints to the terminal that was used to launch TDFG.

The next step is to use these files in a TURBO aeromechanics analysis.

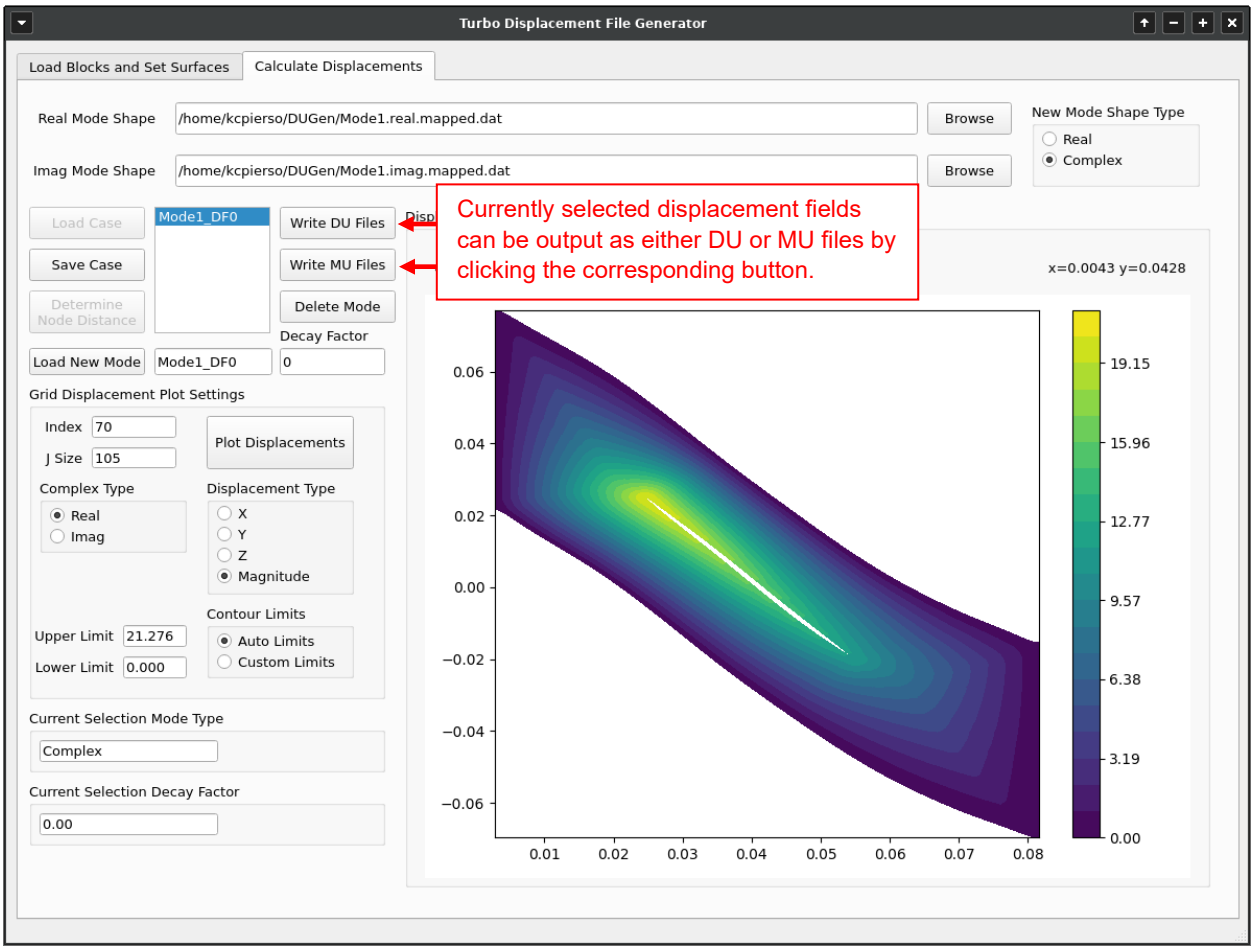

Figure 15.—Click the appropriate button to write either DU or MU files.

References

1. NASA Technology Transfer Program, “TURBO-AE: An Aeroelastic And Multi-stage Aerodynamic Analysis Code Based On Unsteady Three-dimensional Navier-Stokes Equations,” 2024. [Online]. Available: https://software.nasa.gov/software/LEW-17514-1. [Accessed August 14, 2025].
2. K. Pierson, “GAMMA Project on NASA GRC GitLab,” July 2024. [Online]. Available: https://gitlab.grc.nasa.gov/kcpierso/gamma. [Accessed August 14, 2025].
3. K. C. Pierson, “User Manual for Geometry Alignment and Mode Mapping Application (GAMMA),” NASA Technical Reports Server, 2015. https://ntrs.nasa.gov. [Accessed August 14, 2025].
4. K. Pierson, “TDFG Project on NASA GRC GitLab,” July 2024. [Online]. Available: https://gitlab.grc.nasa.gov/kcpierso/turbo-displacement-file-generator. [Accessed August 14, 2025].
5. pyenv Developers, “pyenv Github,” [Online]. Available: https://github.com/pyenv/pyenv. [Accessed August 14, 2025].
6. Miniforge Developers, “Miniforge Github,” [Online]. Available: https://github.com/conda-forge/miniforge. [Accessed August 14, 2025].
7. A. R. Wernick and M. A. Bakhle, “Tail-Cone Thruster Fan Rotor Flutter Analysis,” NASA Technical Reports Server, 2024. https://ntrs.nasa.gov. [Accessed August 14, 2025].