Janus: A Post-Processor for VecTor Analysis Software

by

Ivan N. Chak

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of Civil Engineering
University of Toronto

Abstract

Janus: A Post-Processor for VecTor Analysis Software

Ivan N. Chak

Master of Applied Science

Graduate Department of Civil Engineering

University of Toronto

2013

VecTor is a suite of computer programs developed for the nonlinear finite element analysis of reinforced concrete structures. Due to the substantial nature of output data produced by the programs, accessing pertinent analysis information is not easily accomplished. A graphics-based post-processor would greatly improve the overall utility of the VecTor programs by allowing the multitude of information to be visually displayed and manipulated for the purposes of data synthesis and rapid verification of results.

The intent of this manual is to demonstrate a post-processor program which reads and displays the results of VecTor-based analyses in a robust and straightforward manner. The proposed post-processor program, named Janus, will provide the user with the capability to display both local and global response characteristics. Janus will allow the user to comprehensively recall and manipulate structural analysis results on a model-wide basis as well as display pertinent information for individually specified elements of interest.

Acknowledgements

I gratefully acknowledge and thank Professor Frank Vecchio for his guidance and supervision throughout the duration of this research. His expertise and direction have deeply enriched my academic experiences, and I am grateful for his continued patience and encouragement for me to develop my potential - both as a scholar and as an engineering professional. Without his belief in my abilities, I could not have reached my goals. Additional thanks to Professor Evan Bentz for his invaluable guidance and help in reviewing my work.

I would also like to thank my friends and colleagues Akira Jodai, Vahid Sadeghian, Fady ElMohandes, Trevor Hrynyk, Serhan Güner, David Carnovale and Heather Trommels for their contributions and advice throughout the course of this project.

I would also like to express my gratitude for the financial support provided by: the Natural Sciences and Engineering Research Council of Canada (NSERC), Marshall Macklin Monaghan Group Limited (MMM Group), the University of Toronto, and Professor Frank Vecchio.

Last but not least, I wholeheartedly give thanks to my mother, father, and brother for their loving support in all of my endeavours.

Contents

List of Tables

List of Figures

Notation

ρ	Reinforcement ratio
ρ_t	Transverse reinforcement ratio
ρ_z	Out-of-plane reinforcement ratio
ε_0	Concrete compressive strain, corresponding to f'_c
ε_{sh}	Reinforcement strain hardening strain
ε_u	Reinforcement ultimate strain
A_s	Cross-sectional area of reinforcing bar
d_b	Diameter of reinforcing bar
D_c	Thickness of concrete layer
E_c	Initial tangent stiffness of concrete
E_s	Initial tangent stiffness of reinforcement
f'_c	Concrete cylinder uniaxial compressive strength
f'_t	Uniaxial cracking strength of concrete
f_u	Ultimate strength of reinforcement
f_y	Yield strength of reinforcement
N_x	Number of concrete layers
W_c	Width of concrete layer
Y_s	Centroidal depth of reinforcing bar (from top of section)

Chapter 1

Introduction

1.1 Background

In the contemporary practice of structural engineering, engineers and design professionals are perpetually challenged in their abilities to accurately and efficiently design a structure. In order to satisfy performance-based design standards and safely accommodate extreme loading scenarios, advanced computational tools such as finite element analysis software have been developed to aid designers in conducting analyses beyond conventionally available techniques. To that end, the VecTor software suite has been developed at the University of Toronto for the purpose of modelling the nonlinear behaviour of different types of reinforced concrete (RC) structures. The VecTor software suite currently consists of five programs, each with specialized analysis capabilities for particular types of structural elements: VecTor2 for 2D membrane structures, VecTor3 for 3D solid structures, VecTor4 for plates and shells, VecTor5 for plane frames, and lastly VecTor6 for axisymmetric solids. The veracity of the VecTor programs has been confirmed through an assortment of experimental test programs conducted at the University of Toronto and elsewhere, as well as rigorous analyses of real-world structures such as frames, slabs, shear walls, silos, and even nuclear containment structures.

1.2 Post-Processor Rationale

As advanced nonlinear finite element analysis (NLFEA) software, the VecTor programs generate a considerable volume of data as a result of their analyses. The analysis output data - comprehensively describing the complete structural state and/or position of each node and element included in the model - are discretized into separate files for each load or time stage increment defined by the user. For user verification purposes, the ensuing output data also includes the structural model, analysis and material parameters, and applied loads as they have been interpreted by the VecTor program. While this information is ideally formatted and categorized for efficient and systematic data handling within the digital computing environment, this information layout is conversely problematic for human comprehension. Although standard text editors are capable of displaying the raw numerical arrays of output data, significant automation and customization would be required to present it in a versatile arrangement that is insightful or analytically valuable to the common user. As VecTor finite elements models increase in complexity beyond a nontrivial number of analysis points and/or load stages, ad hoc access of unprocessed output files through conventional text editor programs quickly becomes an ungainly and ineffectual task.

In order to address the difficulties associated with interpreting numerical data from VecTor analyses, a graphical and user-friendly post-processor program is required. The post-processor would help design engineers to assess, confirm and synthesize the resulting information in a useful and convenient manner. It would serve a role as a supporting and terminal platform in the VecTor finite element analysis procedure, providing a visually intuitive and interactive tool for designers to verify their findings as well as meaningfully present them to others.

1.3 Manual Objectives

The purpose of this manual is to present a custom post-processor program that supplements the overall utility of the existing VecTor software suite, providing an effective means to view and interpret analysis output results. The post-processor, designated as Janus, is responsible for presenting the entire finite element model on a macroscopic scale as well as intermediate section and/or layer views where applicable.

Prior to the creation of this document, the software framework for Janus had been partially developed by Dr. Hossein Mostafaei at the University of Toronto in 2008. Due to the fact that there is a diverse array of VecTor programs - each responsible for analyzing a distinct form of RC structure - significant further work was required in order to expand Janus' original data acquisition and display functionality to accommodate all contemporary VecTor model types. Hence, one of the primary objectives of this manual is to exemplify Janus as a post-processor that can universally support all model types currently supported by the VecTor analysis programs, possessing the capability to display relevant structural input features such as nodal restraints, specified material types, load cases, and more.

In a similar vein, the crux of design and functionality requirements in Janus involve being able to display all relevant data upon request by the user, as well as provide pertinent navigation and interface cues for general ease of use. Additionally, the results need to be shown in an intuitive way such that even an unfamiliar viewer would be able to readily interpret the results without significant effort or prior training. Thus, a second major objective of this manual is to demonstrate the ability of Janus to present VecTor program analysis results in a user-friendly manner. Subsequently, VecTor program-specific model examples will be systematically presented in a user-oriented context for practical illustration purposes.

1.4 Organization

Chapter 2 of this manual is designated for providing background information concerning the VecTor analysis software. For the given scope of the manual, discussion pertaining to VecTor is limited to a general qualitative overview of each program, their respective analysis capabilities, as well as a description of the general system of output files used and interpreted by Janus. In Chapter 3, the Janus post-processor program and its facilities are presented in technical detail, describing properties such as: development background and program organization, user interface, and its general functionality as a post-processor program. Chapters 4 through 8 include specific details on each VecTor model type in terms of its interpretation by Janus, as well as any functions which have been customized to a particular VecTor model type. Additionally, Chapter 9 is designated for providing VecTor program-specific demonstrations on how Janus displays each type of modelled structure, as well as discussing available features in the context of the model type being displayed. Lastly, conclusions concerning the post-processor program Janus and recommendations for future work are presented for consideration in Chapter 10.

Chapter 2

VecTor Software Suite

2.1 Introduction

The VecTor software suite is a series of nonlinear finite element analysis programs developed at the University of Toronto. In its modern form, VecTor is represented through five distinct programs, each with a unique range of analysis capabilities. VecTor2 is used for the analysis of 2D membrane structures, VecTor3 is used for 3D solid structures, VecTor4 analyzes 3D shell elements, VecTor5 provides analysis capabilities for plane frames, and finally VecTor6 is employed for the analysis of axisymmetric solids.

The VecTor programs utilize the theoretical bases of the Modified Compression Field Theory (MCFT) (Vecchio and Collins, 1986) and Disturbed Stress Field Model (DSFM) (Vecchio, 2000), and integrate a broad spectrum of nonlinear material models in their analysis procedures. The VecTor programs represent cracked concrete as an orthotropic material with smeared rotating cracks, using an incremental total load, iterative secant stiffness algorithm to achieve a stable and efficient solution procedure (Wong, 2002). Since its inception, numerous improvements to the constitutive modelling relationships employed in VecTor have also allowed for several notable second-order effects to be accommodated in the predicted response of RC elements, such as: tension stiffening, tension softening, compression softening, etc.

By combining such comprehensive material behaviour models with advanced finite element analysis techniques, the VecTor programs are capable of providing superior structural behaviour approximations relative to typical linear-elastic analyses. Confidence in the results of VecTor software can be assured from good corroboration with results obtained with multiple experimental test programs successfully completed at the University of Toronto and elsewhere, as well as actual responses of numerous real-world structures.

Among the VecTor software programs, each program is designated by their distinct structural analysis capabilities - providing specialized and efficient means for approximating the response of distinct types of RC structural elements. Altogether, the software suite is capable of modelling a broad range of structural components that may be readily encountered in the realm of structural engineering. The following sections provide a general description of each of the VecTor programs as they pertain to Janus as a supporting post-processor program.

2.2 VecTor2

VecTor2 is specified for the nonlinear finite element analysis of 2D membrane structures - utilizing fine modelling meshes consisting of low-power elements with two degrees of freedom per node. This approach is deemed appropriate for the analysis of RC structures, since the relative high element density allows for adequate reinforcement detailing as well as satisfactory presentation of localized crack patterns (Wong, 2002).

The element library for VecTor2 models consists of the following element types: three-noded triangular elements, four-noded rectangular elements, four-noded quadrilateral elements, two-noded truss-bar elements for representing discrete reinforcement, and lastly, two-noded link elements and four-noded contact elements for representing slip and contact interactions. See Table 2.1 below for a graphical representation of the element types available in VecTor2.

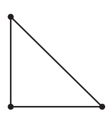
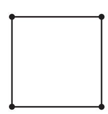
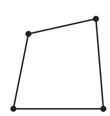
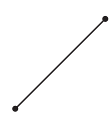
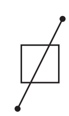
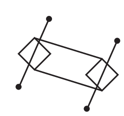
					
Triangular	Rectangular	Quadrilateral	Truss	Link	Contact

Table 2.1: VecTor2 Element Library

2.3 VecTor3

The analytical capabilities of VecTor3 are developed in an analogous manner to VecTor2, although extended for approximating the behaviour of solid finite elements in three-dimensional space. The intent of VecTor3 is to function at a similar capacity to VecTor2 but also be able to accurately capture all forms of out-of-plane behaviour which had been inherently neglected due to the 2D nature of VecTor2 analyses.

The element library for models developed in VecTor3 is similar in capacity to VecTor2, including: six-noded wedge elements, eight-noded regular and isoparametric hexahedral elements, two-noded truss-bar elements, and two-noded link elements. A conceptual representation of the available VecTor3 element types may be observed in Table 2.2.

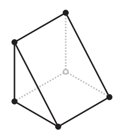
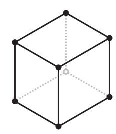
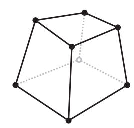
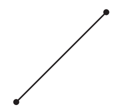
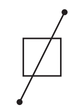
				
Wedge	Regular Hexahedral	Isoparametric Hexahedral	Truss	Link

Table 2.2: VecTor3 Element Library

2.4 VecTor4

VecTor4 is used for the analysis of 3D shell elements, capable of accounting for nodal displacements as well as in-plane rotations. In contrast to the other VecTor programs discussed thus far, shell elements are relatively high-power elements - utilizing nine-noded layered heterosis elements with 42 degrees of freedom (Hyrnyk, 2013). A typical shell element consists of eight perimeter nodes, each with three displacement and two rotational degrees of freedom, and a ninth central node possessing two rotational degrees of freedom.

In order to provide analysis capabilities of curved surfaces such as shells, VecTor4 features the use of numerical integration and quadratic shape functions, requiring a unique array of Gauss Points for each shell element. Accommodating in-plane rotations allows for the shell elements to demonstrate displacements due to bending in addition to typical axial and shear deformations. As well, users assign distinct material properties to each shell element, specifying a finite number of concrete and reinforcement layers to stratify through the out-of-plane thickness of the element. In doing so, discrete sectional response characteristics of each element are obtained for each load stage - providing the means for sectional analysis capabilities in addition to the overall global structural response.

The element library for VecTor4 consists of nine-noded shell elements and two-noded truss-bar elements, as illustrated in Table 2.3.

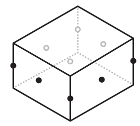
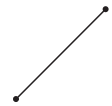
	
Shell	Truss

Table 2.3: VecTor4 Element Library

2.5 VecTor5

VecTor5 is designated for the purpose of analyzing plane frames. Finite element models in VecTor5 consist entirely of two-noded member elements, with the capability to account for nodal displacements as well as in-plane rotations. Akin to VecTor4, users specify a number of concrete and reinforcement layers as part of the assignment of member material properties. In addition to the global structural response data, users are able to customize the resulting VecTor5 analysis data files by optionally requesting the sectional stress and strain conditions of specific member elements to be included as part of the analysis output file. (Güner, 2008).

The element library for VecTor5 consists solely of two-noded member elements, as presented in Table 2.4.

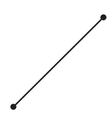

Member

Table 2.4: VecTor5 Element Library

2.6 VecTor6

In a similar fashion to VecTor2, VecTor6 is designated as a 2D finite element analysis program. However, VecTor6 differs in its ability to treat model elements in an axisymmetric fashion, extruding them about a common axis of rotation as annular shapes. Although VecTor6 models are visualized as simplified 2D structural models, VecTor6 is able to accommodate both in-plane and out-of-plane forms of discrete reinforcement.

The VecTor6 element library consists of the following element types: three-noded axisymmetric triangular elements, four-noded axisymmetric quadrilateral elements, two-noded truss elements for representing radial and longitudinal discrete reinforcement, and one-noded ring bar elements for representing circumferential discrete reinforcement. See Table 2.5 below for a graphical representation of element types available in VecTor6.

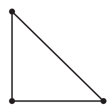
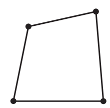
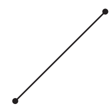
			
Triangular	Quadrilateral	Truss	Ring Bar

Table 2.5: VecTor6 Element Library

2.7 VecTor File Format

2.7.1 Overview

In order to successfully initialize the finite element analysis procedure, VecTor programs require an extensive series of input parameters which provide a complete description of the structural model, including: node and element definitions, material properties, load cases, analysis parameters, load case data, etc. Within the VecTor program nomenclature, each structural model and corresponding set of analysis specifications is denoted as a "job". As such, general job parameters such as load case specifications, structure and load filenames, model description and analysis options are consolidated within a central input file known as the job file. In a similar manner, a separate input file known as the structure file specifically contains data relating to the structural model itself - nodal coordinates, element declarations, and material properties. Users may assign up to five distinct types of load cases to be superimposed upon the structural model. While general load characteristics of each load case are stored within the job file, the specific input data for each load case are organized into corresponding separate load files.

For versatility and ease of user input, many fields of user input data within the structure file and load file(s) may be entered in an incremental format. Essential modelling instructions, such as establishing node coordinates, element geometry, material assignment and load declarations - which may otherwise be cumbersome and inefficient to manually enter on an individual basis - may be completed by specifying incremental node and/or element values in different orthogonal directions. For structural models with regular geometry and/or loading configurations, the amount of manual data entry required to establish a new model using the VecTor programs can be significantly mitigated through use of the incremental input format.

The pre-processor program FormWorks, originally developed by Wong in 2002 and improved by Sadeghian in 2012, provides a user-friendly graphic interface for the creation and development of structural models for analysis using the VecTor software suite. FormWorks serves as a superior alternative to using standard text editors for specifying VecTor input file parameters - allowing users to actively create, review and modify entries to the job, structure, and load files in real time.

At the onset of each analysis execution, the VecTor programs produce a matching set of new output data files which reiterates the provided structure and load input parameters in an expanded form, effectively allowing users to verify that all of the input variables have been correctly chosen and interpreted as intended. The expanded structure and load files explicitly list all applicable values per node and/or element as they have been declared by the user. This feature of the expanded structure and load files is particularly useful in verifying that all incremental node and/or element assignments have successfully encapsulated the desired range - especially if a graphical pre-processor program like FormWorks is unavailable to visually display the user-intended inputs.

During the analysis procedure, successful convergence of each user-prescribed load stage produces a new and distinct analysis output file containing the associated analysis output data, sequentially named as per the defined name convention denoted in the job file. Like the expanded structure and load file(s), VecTor analysis output files state all resulting node and/or element-related variables in an explicit fashion. As mentioned in Section 1.2, the sheer volume of unrefined data organized per load stage makes numerical interpretation and synthesis an ungainly task for users. The fact that output data is separated into distinct files poses a particular challenge for analysis purposes, as structural analysts are often interested in observing the magnitude and relative change in a specific structural response parameter over a sequential range of load stages.

Janus utilizes the expanded structure files, expanded load files as well as the expanded analysis data files in order to graphically represent the results of the VecTor software analysis. Since the expanded structure and load files are created as a direct result of the VecTor analysis procedure, Janus also utilizes them as a basis of displaying the structural model and load cases - facilitating a graphical means for users to directly verify that their model specifications are accurately defined. For a conceptual overview of file types associated with VecTor, see Figure 2.1. The following sections provide a general description of the files used and generated by VecTor programs as they relate to Janus.

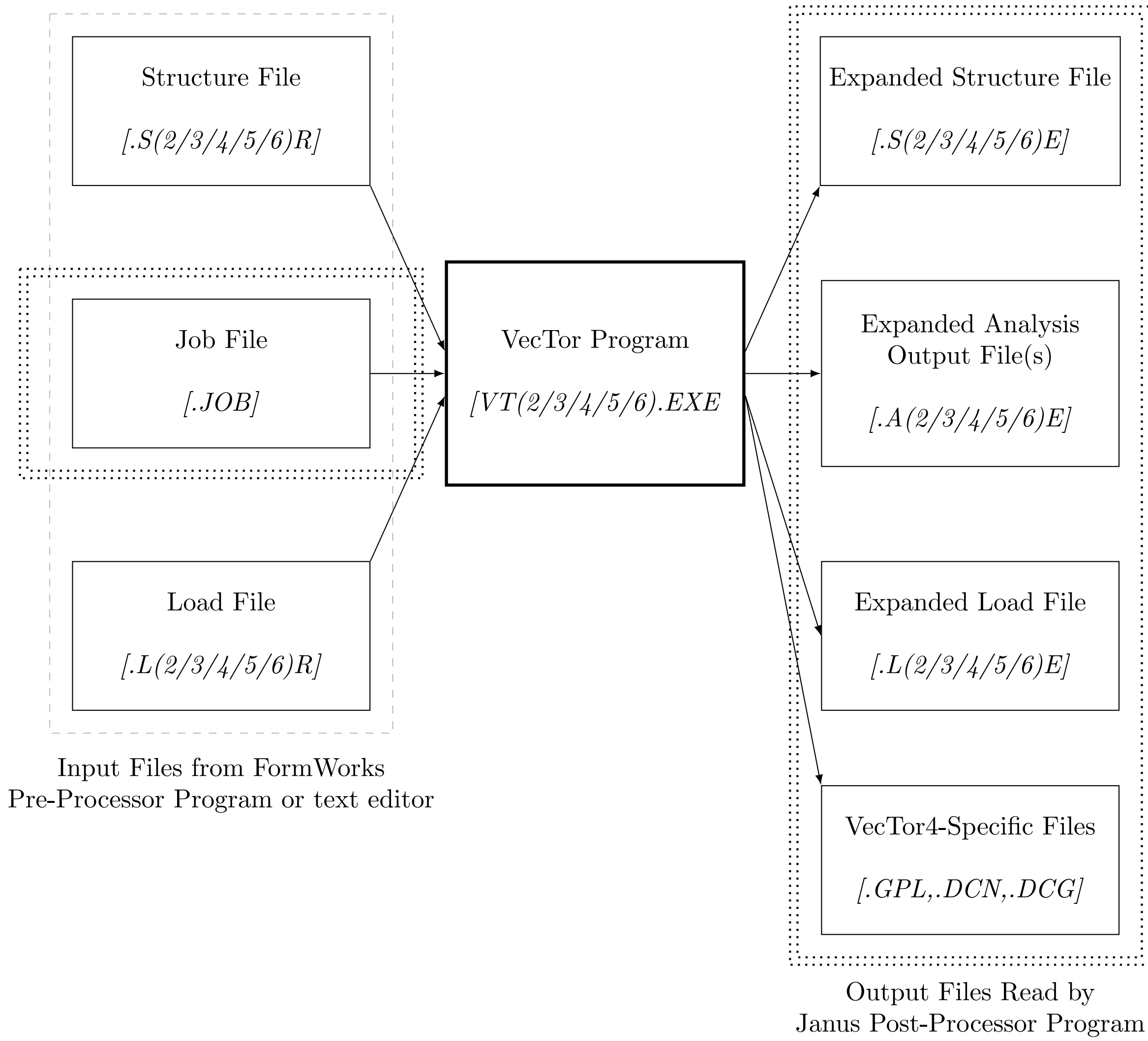


Figure 2.1: Conceptual Organization of VecTor Program File Types

2.7.2 Job File

The organization of the job file format is largely consistent across all VecTor programs. Job files may be specifically tailored for input into a particular VecTor program by listing additional options for analysis modes and/or parameters. Job files are universally distinguished by the file extension ".JOB".

In terms of structure-specific data within the job file, the user designates the name of the structure file, as well as the VecTor program associated with the type of structure to be analyzed - "2" for VecTor2, "3" for VecTor3, and so on.

The user also specifies several load case related parameters within the job file, including: number of load stages, starting load stage number, and expanded analysis data file naming convention. Load stage specifications for up to 5 distinct load cases may be activated and provided within the job file. The load case specifications provide applicable entries for the following values: load file name, initial and final load factor values, load increment values, loading type, number of repetitions, and cyclic increment factor.

Following the load case data, the job file lists various options for analysis parameters and material behaviour models. In order to customize analysis settings, users may provide numerical values corresponding to applicable options denoted in the VecTor software literature (Wong et al., 2012).

2.7.3 Expanded Load File

For each active load file listed in the job file and read by the VecTor program, a matching expanded load file is produced. The expanded load files have program-specific file extensions corresponding to the software which produced them. For example, VecTor6 is associated with ".L6E" file extensions, VecTor2 with ".L2E", etc. As previously described, each expanded load file explicitly reiterates the input load data values from the input load file as it had been interpreted by the VecTor program. The expanded load file lists all relevant values for each load assignment, such as load magnitude and direction of application. As the expanded load files are an integral component of model verification, Janus reads the expanded load file as part of its post-processing capabilities. By viewing graphically displayed loads acting upon a structural model, users can readily verify whether or not the assigned load values are applied to the intended locations with the correct magnitude and sense.

2.7.4 Expanded Structure File

Upon initiation of a VecTor analysis procedure, a new expanded structure file is created. In a likewise manner to the expanded load file format, expanded structure files have specific file extensions denoting the VecTor program source. For example, VecTor4 would generate expanded structure files with the extension ".S4E", while VecTor5 expanded structure files have the file extension ".S5E", and so on. Each generated expanded structure file contains a complete description of the finite element model, customized to the element library of the respective VecTor program source. A general overview of the expanded structure file contents is provided in Table 2.6.

2.7.5 Expanded Analysis Data File

Each successfully converged analysis of a load stage produces an expanded analysis data file. Expanded analysis data files have file extensions ".A2E" to ".A6E" to represent VecTor2 to VecTor6, respectively. The expanded analysis data file represents the complete structural response of the structure subjected to

Structural Parameters	• User-specified structure title and file name • Number of declared reinforced concrete, steel, and/or bond material types • Total quantities of utilized element types • Total quantities of restrained degrees of freedom or restrained nodes
Material Specifications	• Reinforced concrete material properties • Steel material properties (if applicable) • Bond material properties (if applicable)
Element Information	• Nodes per element • Material type per element
Nodal Coordinates	• Global coordinates per node
Support Restraints	• Restrained nodes • Restrained degree(s) of freedom per node

Table 2.6: Expanded Structure File Contents

a particular load increment. A general overview of the expanded analysis output file contents is provided in Table 2.7.

2.7.6 VecTor4-Specific Files

As previously described in Section 2.4, VecTor4 utilizes degenerated heterosis finite elements and quadratic shape functions in its analysis procedures, requiring a need for numerical integration and unique arrays of Gauss Points to be assigned to each modelled shell element (Hyrnyk, 2013). The generation of Gauss Points is completed through computations within the VecTor4 analysis environment, and the list of global coordinate positions of elemental Gauss Points are exported in a separate external file. For modelling verification purposes, it may be of interest to the user to inspect and confirm the positions of the Gauss Points determined by VecTor4 software formulations.

In order to allow for VecTor4 element nodes to independently displace in all orthogonal directions as well as exhibit in-plane rotations, unique local coordinate systems must be adopted for each Gauss Point and node per element. Similar to the internal processing of Gauss Points, VecTor4 establishes local Gauss Point and nodal coordinate systems at the time of program execution. The local Gauss Point and nodal coordinate systems are denoted as the local coordinate system and nodal coordinate systems, respectively (Hyrnyk, 2013). VecTor4 also utilizes a fourth coordinate system known as the curvilinear coordinate system for analysis purposes, but it is not required for any post-processing purposes in Janus. Although

General Job, Load and Structure Parameters	• User-specified job title and file name • User-specified structure title and file name • User-specified analysis data file name and current load stage • Number of iterations, convergence factor and specified averaging factor
Loading Condition	• User-specified load case file name(s) • Load case load factor(s) for current load stage
Analysis Parameters	• User-specified material model parameters and analysis settings
Secant Moduli Convergence Factors	• Secant moduli convergence factors in principal directions
Vital Signs	• Applicable material parameters used for general structural diagnostics purposes
Crack Conditions	• Applicable crack data for RC elements
Nodal Displacements and Reactions	• Applicable nodal displacements, rotations and/or reaction values
Element Stresses and Strains	• Applicable element stresses and strain values

Table 2.7: Expanded Analysis Output File Contents

direct representation of the nodal and local coordinate systems may not be of direct utility to the user, inclusion of such coordinate systems is key for the post-processing functionality of displaying VecTor4 models in Janus. Several element-specific features, such as in-plane reinforcement layer orientation and RC layer thicknesses, are defined relative to the local or nodal coordinate systems. In order to provide the greatest utility in visualization to the user, Janus reads the external local coordinate system files as they are provided as part of the VecTor4 analysis output. See Figure 2.2 below for a conceptual representation of the a) global (x, y, z), b) nodal (v_1, v_2, v_3, θ_1, θ_2) and curvilinear (ξ, η, ζ) and c) local (x', y', z') coordinate systems.

2.7.6.1 Gauss Point List File

At the beginning of an analysis, VecTor4 generates a complete list of shell element Gauss Point locations in a separate file with the file extension ".GPL". Akin to the definition of nodal coordinates, the locations of Gauss Point for each element are provided on a global VecTor4 model coordinate basis.

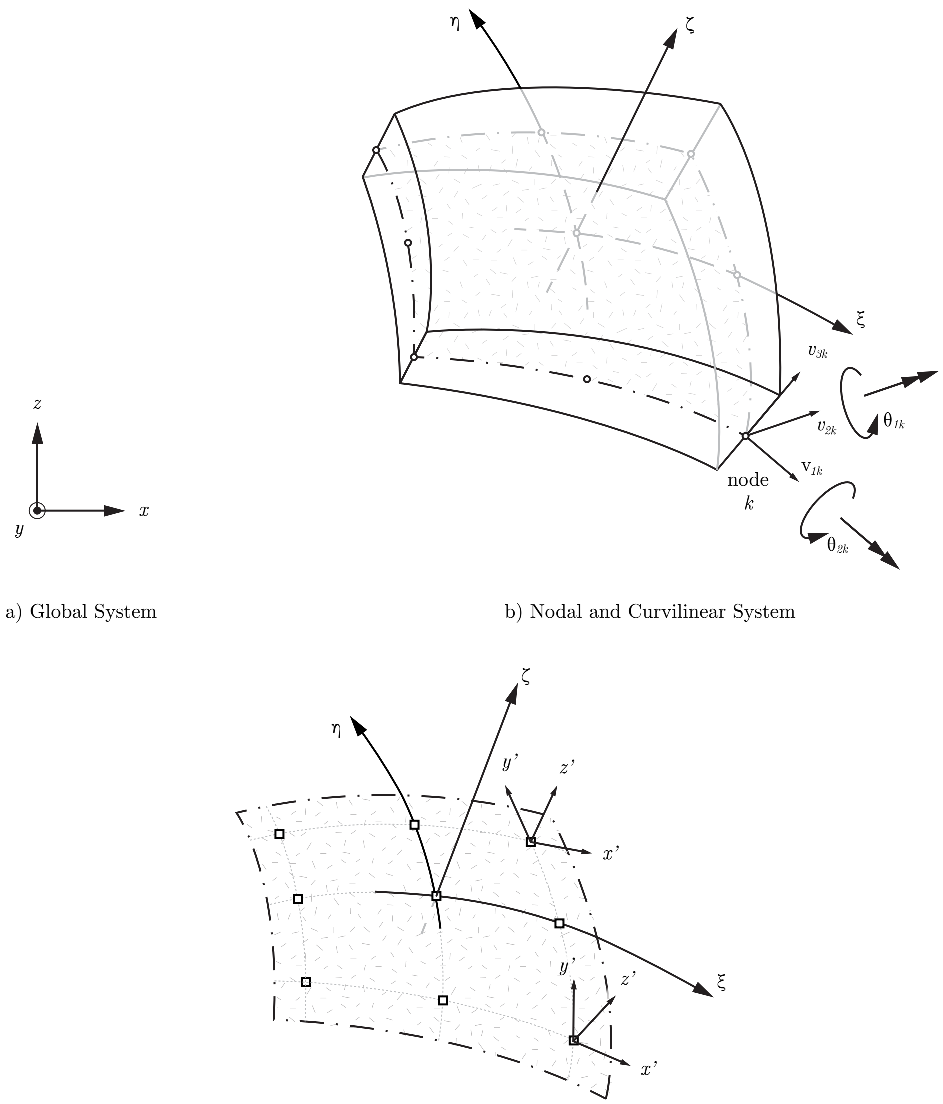


Figure 2.2: VecTor4 Coordinate Systems (adapted from Huang and Hinton, 1986)

2.7.6.2 Node Directional Cosine File

The nodal coordinate system determines a set of local orthogonal axes per node, allowing for rotations and curved surfaces to be accommodated in the overall global structural response of the RC shell structure. All local axes are encapsulated within a separate ".DCN" extension file. Each local axis system is represented using a set of orthogonal unit vectors denoted as v_1, v_2, and v_3. Rotation vectors θ_1 and θ_2 are defined as the rotation about vectors v_2 and v_1, respectively. Directional cosine values of the three axes are provided for each node.

2.7.6.3 Gauss Point Directional Cosine File

Lastly, the Gauss Point directional cosine file, or ".DCG" file, defines the local axis system established at each Gauss Point. The local axes, denoted as x', y', and z', are prescribed using unit vectors in each orthogonal direction. The local x' vector at each Gauss Point is used to define the relative angle of orientation for in-plane reinforcement layers within each shell element at each Gauss Point.

Chapter 3

Janus Post-Processor Program

3.1 Introduction

Janus is a post-processor program developed for supporting the VecTor suite of nonlinear finite element analysis programs, providing a user-friendly interface for comprehensively viewing and synthesizing analysis results as well as verifying model specifications. The overarching role of Janus is to function as a stand-alone program capable of reading analysis output files from all five VecTor programs, visually displaying each type of model in a context that facilitate the predicted needs of the structural analyst or designer.

3.2 Program Structure

3.2.1 Background

Janus is a post-processor program developed using the C++ programming language and compiled using Microsoft Visual C++ Studio 6.0. C++ was chosen for use as a programming language for its overall versatility and ease of use as an intermediate-level object-oriented programming (OOP) language. In general, OOP operates under its fundamental namesake by identifying and organizing groups of data and functions as distinct conceptual types of objects. In C++, objects known as classes are used as a versatile means for programmers to easily identify, store and handle information; anything from a single binary value to entire class objects may be abstracted and contained within a class, or even passed to another class or function. New classes can also be derived from base classes, and programmers may actively select which class characteristics are inherited from one hierarchy level to the next. The overall popularity of the C++ language provides a formidable range of support resources and class libraries that are readily available at the disposal of the programmer.

One such class library used in Janus is the Microsoft Foundation Class (MFC) library, which conveniently encapsulates many aspects of the Windows Application Programming Interface (API) as self-contained classes for manipulation within the programming environment. Among its multitude of uses, the Windows API is used for facilitating interactions between software, Windows and graphical output devices (such as monitors and printers), as well as provide the basic framework for the creation of common program features such as dialogs, controls, and other user interface features. For the given project scope and context of developing Janus as a graphical post-processor program in Windows, the program design is critically influenced by the user interface and the need for integration within the Windows operating system. MFC readily addresses such issues, taking advantage of the existing Windows API to facilitate communications between Janus and Windows while providing powerful and ready-to-use tools for developing user controls.

In order to provide multifaceted rendering capabilities for displaying and manipulating finite elements in 2D and 3D space, Janus also implements the Open Graphics Library (OpenGL) API. As a widely used specification utilized for scientific visualization and technical simulation purposes, OpenGL is an appropriate solution for drawing the complex range of geometric shapes which constitute the entire collective VecTor element library. At a basic level, series of shapes and lines in OpenGL (known as primitives) are rendered by declaring vertex coordinates in 3D space. This coincides well with the system of element and node coordinate definitions employed across all VecTor expanded structure file formats. Combined with a powerful capability to individually modify visual parameters such as line width, face fill, and colour specifications for each rendered primitive, OpenGL facilities provide the programmer with complete control for presenting model geometry and result data in a visual and comprehensive manner.

3.2.2 Compatibility

The original software framework for Janus was conceived by Dr. Hossein Mostafaei at the University of Toronto between 2007 and 2008, then intended for use within the Windows XP operating system and 32-bit memory environment. Progress towards improved functionality and the overall development of Janus was continued between 2011 and 2013 through the collaborative efforts of the author and Akira Jodai. Although the operational stability and post-processing capabilities of Janus have both improved markedly since its prototypical stages of development, display and function-related compatibility issues may be experienced when using Janus on computers with specifications that are inconsistent with the recommended operating system of Windows XP and 32-bit memory environment.

3.2.3 Architecture

The programming structure of Janus is organized in the expected layout of a typical MFC-based program. Due to the project name for Janus originally being titled VecHom, much of the derived class nomenclature of Janus involves the use of the term “VecHom” as a descriptive prefix. The base class, CVecHom, is derived from the fundamental MFC base class CWinApp, and is responsible for integral application functionality such as initializing, opening, running and closing the program (Horton, 2008). MFC-based programs may have the design option of Single Document Interface (SDI) or Multiple Document Interface (MDI). As the terminology implies, SDI programs are utilized for programs that are intended to read and display a single document at a time, while MDI programs may have more than one open at once. Although Janus was originally developed using the MDI design option, display support for simultaneously opening multiple VecTor models is not fully integrated at this time. As a result, singly opening and closing VecTor models in Janus is currently recommended for optimal post-processing purposes.

Within the child window, the presentation of the VecTor model is handled by fundamental class objects known as a document (CVecHomDoc, derived from the MFC base class CDoc) and a view (CVecHomView, derived from the MFC base class CView). The document object is responsible for holding and arranging all the data that the user will ultimately be interacting with, while the view object contains the different ways that the document data may actually be presented to the user within the bounds of the application window. Subsequently, user inputs into MFC-based controls may be used to modify the arrangement of data and activation of functions within the document and view objects.

3.2.3.1 Document Class

In the context of Janus, the CVecHomDoc object is used to hold all the data pertaining to the opened VecTor model. This information includes all the data within the job file itself, as well as the data from the expanded load file, expanded structure file, expanded analysis output file(s) - and in the event that a VecTor4 analysis is opened, information from VecTor4-specific files are read in as well. CVecHomDoc also contains the functions necessary for parsing the raw text data from the opened output files and assigning them into relevant class, arrays and variables for efficient handling and manipulation within the Janus program environment.

Relevant code for interpreting the VecTor model elements and nodes as well as storing them as equivalent OpenGL primitive shapes is also encapsulated inside of the CVecHomDoc object. Within the CVecHomDoc class, model-related data are organized in distinct classes, encapsulating related information such as: node and element coordinate information, stress values, crack specifications, etc. As previously described, the rendering of OpenGL primitives is accomplished by declaring series of coordinates in 3D modelling space. However, the established scale and position of the coordinate axes relative to the VecTor model are freely chosen by the user during the pre-processing stage of analysis, and may not coincide with the OpenGL coordinate system utilized in Janus model space.

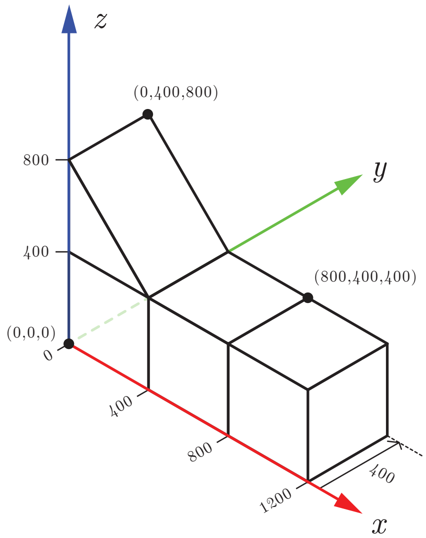
For the purpose of providing consistent visual controls among all viewed models in Janus, coordinate transformations must be applied for conversion from VecTor modelling space to the OpenGL rendering environment. Within Janus, all model coordinates are normalized by the maximum absolute difference between coordinate values on a given axis and centred about the OpenGL coordinate axis origin point. A model demonstration of the contrast between VecTor and OpenGL coordinate systems is provided in Figure 3.1. By inspection, the maximum absolute difference in coordinates exists along the established range of x-axis values, chosen by the user as 0 to +1200. Accordingly, Janus internally transforms the 0 and +1200 lower and upper bounds to -1.50 to +1.50, respectively - with the centre of the model positioned at the origin. All other coordinate values for nodal displacements and model features are transformed and scaled in a congruent manner.

Additional logic statements and functions within the CVecHomDoc class object allows Janus to specify the rendering style as well as toggle which OpenGL primitives are declared and/or omitted at the time of rendering. By integrating user interface control inputs as parameters and arguments to be passed into these functions and logic checks, the user is effectively able to control the manner in which model data are displayed on-screen. This forms the basis upon which Janus operates as a graphical post-processor program.

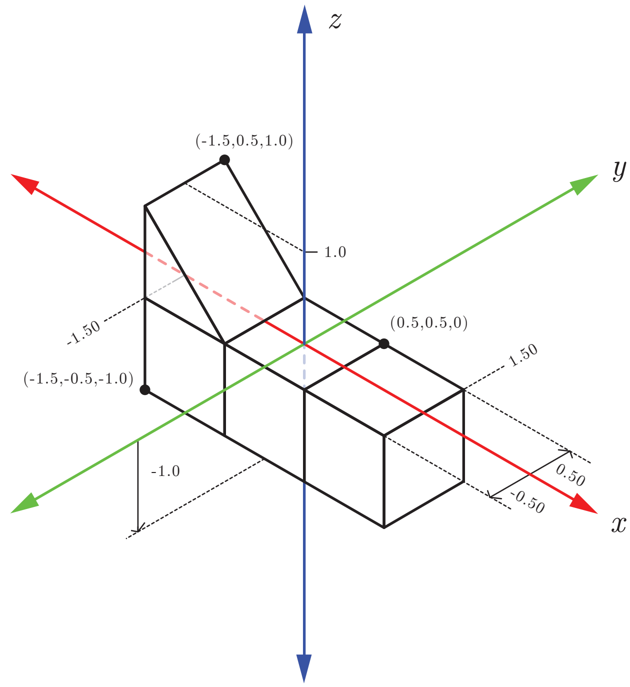
3.2.3.2 View Class

As a complimentary component to its associated CVecHomDoc class object, CVecHomView specifies the way the VecTor model from the document is rendered and presented upon the application screen. In the context of OpenGL specifications, the view object provides facilities for positioning and customizing the conceptual viewing “camera” (Figure 3.2) that presents the on-screen projection of the rendered VecTor model. In order to be visible on screen, shapes must lie within the viewing volume of the camera known as a view frustum (Ahn, 2013). For a typical perspective view, the view frustum resembles a square-based pyramid with a truncated apex. The bounding faces consist of the near and far clipping planes, as well as the top, bottom, left and right clipping planes.

In a similar fashion to controls within the CVecHomDoc object, real-time feedback from user interface



a) VecTor coordinate system



b) Janus/OpenGL coordinate system

Figure 3.1: Comparison of VecTor and Janus/OpenGL Coordinate Systems

controls may be passed into CVecHomView functions to modify the camera parameters. From initially-specified camera variables such as position, orientation and perspective in OpenGL space, the camera may be subsequently modified at the convenience of the user.

3.3 User Interface

The user interface of Janus is presented in Figure 3.3. Asides from numerical and text inputs from the keyboard for specific prompts and/or dialog windows, the majority of user inputs and navigation controls in Janus are implemented through the use of a standard two-button mouse with clickable middle scroll wheel button. The following sections provide an instructional description of the user interface functionality integrated within Janus.

3.3.1 Tool Bar and Menu Area

As shown in Figure 3.4, the Janus toolbar and menu area provides the user interface facilities for viewing the model, displaying analysis output data, as well as invoking or closing various information dialog windows. For each VecTor model type, different toolbar buttons may be activated or disabled

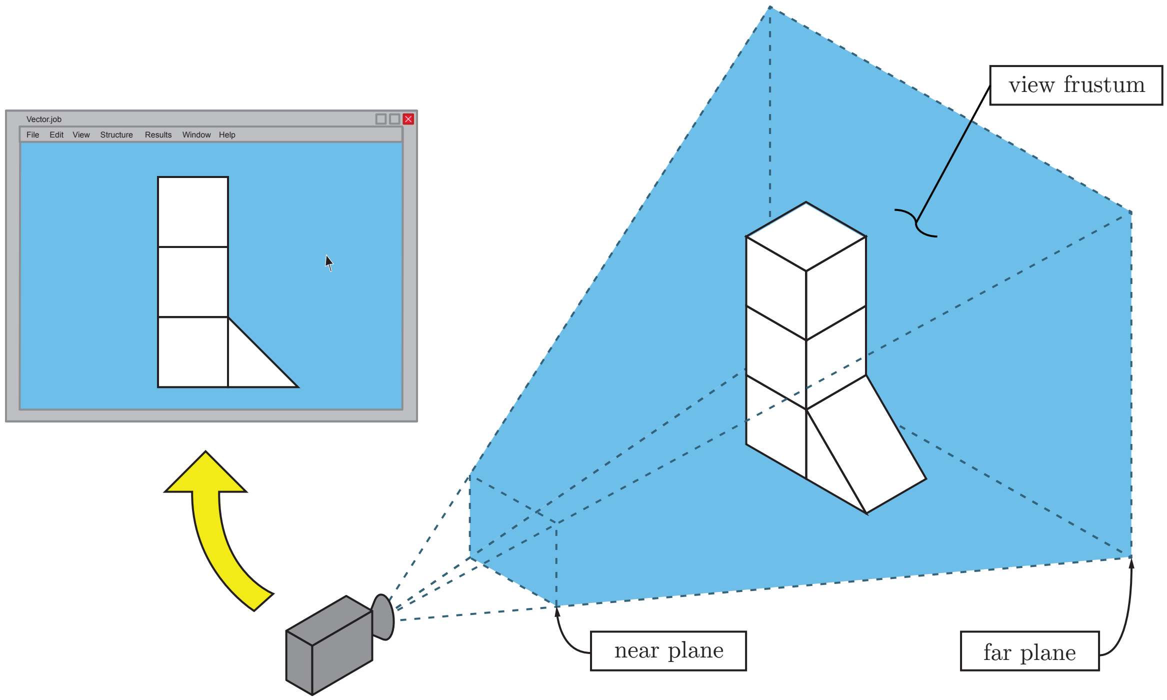


Figure 3.2: Camera View Concept in OpenGL

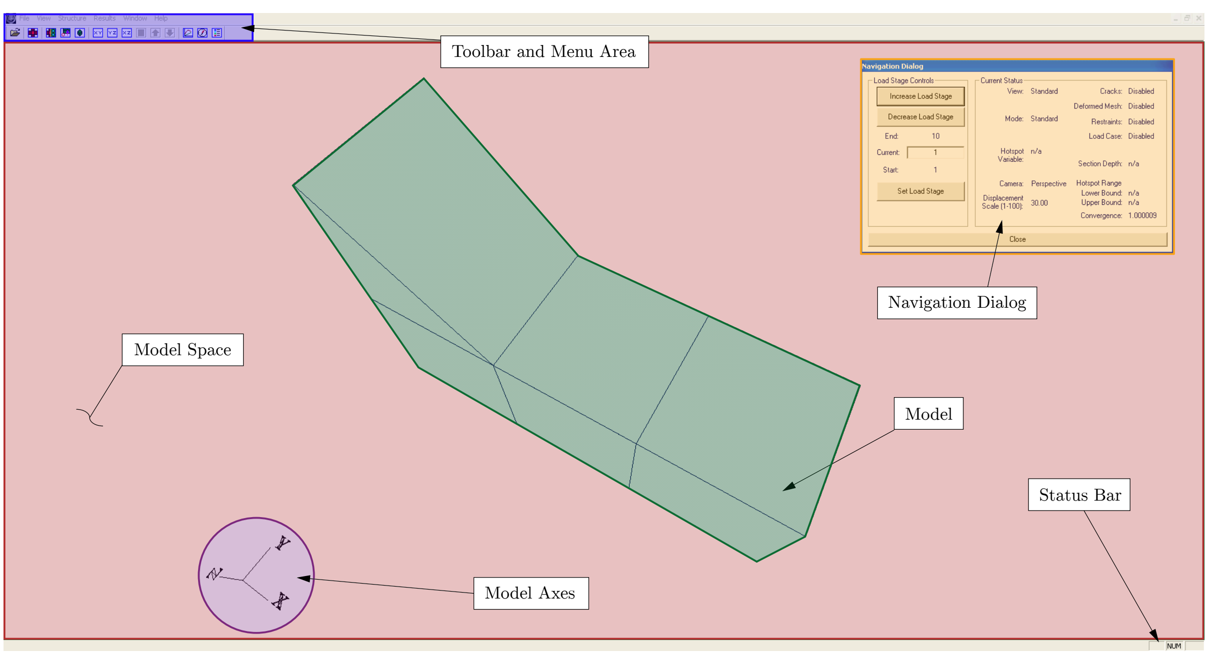


Figure 3.3: User Interface Overview

based on the described function and context of the button. See Table 3.1 for a general description of each toolbar button.

3.3.2 Navigation Dialog

The *Navigation* dialog, shown in Figure 3.5 below, provides general status information and load stage controls for the VecTor model being displayed in Janus. The *Current Status* area describes the existing state of view and mode options as they have been selected by the user; applicable entries will dynamically update as relevant view and mode options are selected and/or modified.

The *Load Stage Controls* area of the *Navigation* dialog provides load stage-related information (*Start*, *Current*, and *End* load stage values) as well as buttons for users to incrementally traverse up or down through the range of available load stages. As the current load stage is modified, the *Current* edit box (as well as the *Convergence Factor* value) will automatically update. The *Current* edit box passively displays the current load stage, but users may also use it to change the current load stage to an arbitrary

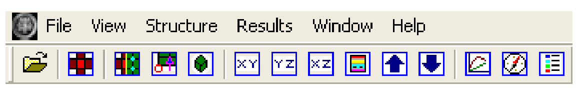


Figure 3.4: Toolbar and Menu Area

Toolbar Button	Icon	Description
Open File		Invoke standard MFC dialog for opening VecTor (.JOB) files
Hotspot Mode		See Subsection 3.8.4 on page 67
Toggle Face Fill		See Table 3.4 on page 43
Toggle Node Feature		
Toggle 3D View		
XY Section		See Subsection 3.6.7.1 on page 37
YZ Section		
XZ Section		
Layer View		See Subsection 3.6.7.2 on page 38
Section Up		See Subsection 3.6.7.3 on page 40
Section Down		
Data Platform		See Section 3.9 on page 70
Toggle Navigation Dialog		Toggle visibility of the *Navigation* dialog (See Subsection 3.3.2)
Toggle Loadcase or Legend Dialog		Toggle visibility of the *Legend* or *Loadcase* dialog

Table 3.1: Toolbar Button Descriptions

chosen load stage value. In order to successfully change the current load stage value using the *Current* edit box:

- The input value must be $\geq$ *Start* and $\leq$ *End* load stage values
- The user must confirm the selection by a) clicking the *Set Load Stage* button, or b) pressing the return key on the keyboard

Alternatively, users may incrementally increase or decrease the current load stage using the page up and page down keys, respectively. For ease of viewing, the *Navigation* dialog may be moved anywhere within the borders of the Janus mainframe. The *Navigation* dialog may also be closed via the *Close* button or using the *Toggle Navigation Dialog* button in the toolbar area. Once closed, the *Navigation* dialog must be re-opened using the *Toggle Navigation Dialog* button.

3.3.3 Model Axes

The Janus model axes are used to depict the current rotational orientation of the VecTor model to the user. The orthogonal axis lines correspond to the positive x-, y-, and z-directions in model space. Upon rotation of the VecTor model, the model axes exhibit an equivalent pivot about the apparent origin point. In doing so, the drawn axes maintain a consistent orientation with the rendered model coordinates, and serve as a spatial navigation guide for the user. The model axes provides particular utility for several commonly encountered post-processing activities, including:

- Free-hand rotation of a 3D VecTor model

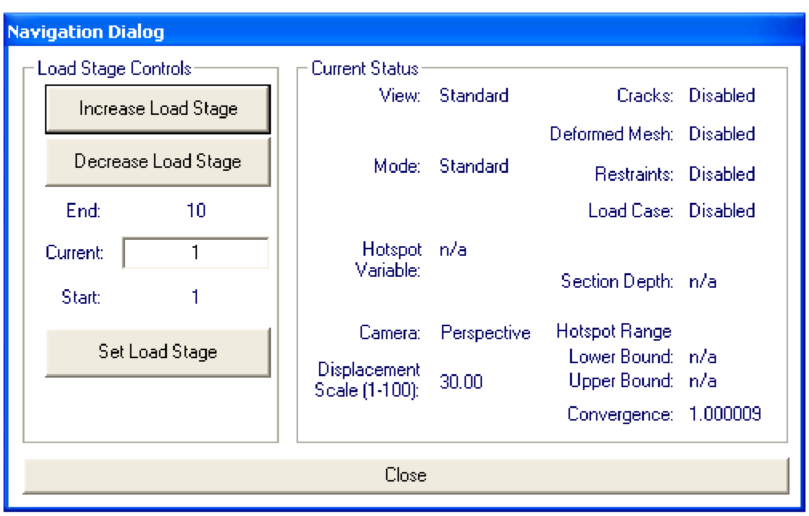


Figure 3.5: Navigation Dialog

- Confirming rotational orientation before/after the camera view rotation is modified via the *Set Camera View* dialog
- Transitioning between the Global Model View and various planar or sectional views

3.3.4 Model Space

Within the bounds of the model space, inputs from the mouse provide specific contextual commands for modifying variables contained within the Janus document and view objects.

When the left mouse button is held and the mouse is dragged, the on-screen display of the model will shift corresponding to the incremental horizontal (left-right) and vertical (up-down) distance moved by the mouse. Upon release of the left button, the model is held at the last recorded position. See Figure 3.6 for a visual demonstration of the left mouse button behaviour in the model space.

Holding the track wheel mouse button while moving the mouse will result in the VecTor model moving in a combined x-, y-, and z-rotation about the fixed camera position, with the magnitude of rotation corresponding to the incremental mouse displacement. The direction of the model rotation is directly proportional to the direction moved by the mouse. For example, with the default main axes of rotation set at as the x- and y-axes, a downward movement of the mouse will "tilt" the model down about the x- and/or y-axis, while a lateral movement to the right will cause the model to rotate counterclockwise about the z-axis. An explanation of main rotational axes is subsequently provided in Subsection 3.6.4.5. Upon release of the track wheel button, the model is held at the last recorded orientation. For a visual depiction of the track wheel mouse button behaviour in the model space, refer to Figure 3.7. Note that manipulation of the model through rotation is exclusively available for use while in Global Model View for VecTor3 and VecTor4 and viewing layered elements in Layer View for VecTor4. Predictably, 2D models in VecTor2, VecTor5, and VecTor6 must remain in plane view in both Global Model View, Section View, and/or Layer View.

Scrolling the track wheel button causes the camera view to zoom inwards and outwards along the given line of sight, with the distance zoomed corresponding to the incremental distance scrolled on the mouse track wheel. Scrolling the track wheel up zooms the camera view in, while scrolling down zooms it out. Once the track wheel ceases scrolling, the camera is held at the last recorded position. See Figure 3.8 for a visual demonstration of how track wheel scrolling is represented within the model space.

Lastly, the right mouse button is used exclusively for the purpose of invoking the *Element Attributes* dialog. When the mouse is right clicked while the mouse cursor is positioned over an element in Global Model View, the element attribute screen appears with applicable model and structural response data listed for the selected element. For user verification, the selected element is highlighted in red, while the remaining elements are coloured in blue for contrast. Once the *Element Attributes* dialog is closed, the model face colours resume the previous display mode. Right mouse button behaviour in Janus is demonstrated in Figure 3.9.

3.3.5 Status Bar

The status bar displays relevant descriptive prompts when the mouse cursor hovers over Janus menu and toolbar area buttons. For viewing convenience, the status bar may be enabled or disabled via **View** ▷ *Status Bar*. By default, the status bar is enabled.

3.4 Loading Options

Upon opening a VecTor analysis with a potentially memory-intensive number of load stages and/or finite elements, Janus provides several options for selectively customizing the data to be loaded into memory. Based on a VecTor-specific memory factor of (Load Stages · Total Element Count), the *Load Stage Range* dialog shown in Figure 3.10 will appear if the threshold value is exceeded. The threshold values for each VecTor program are provided in Table 3.2.

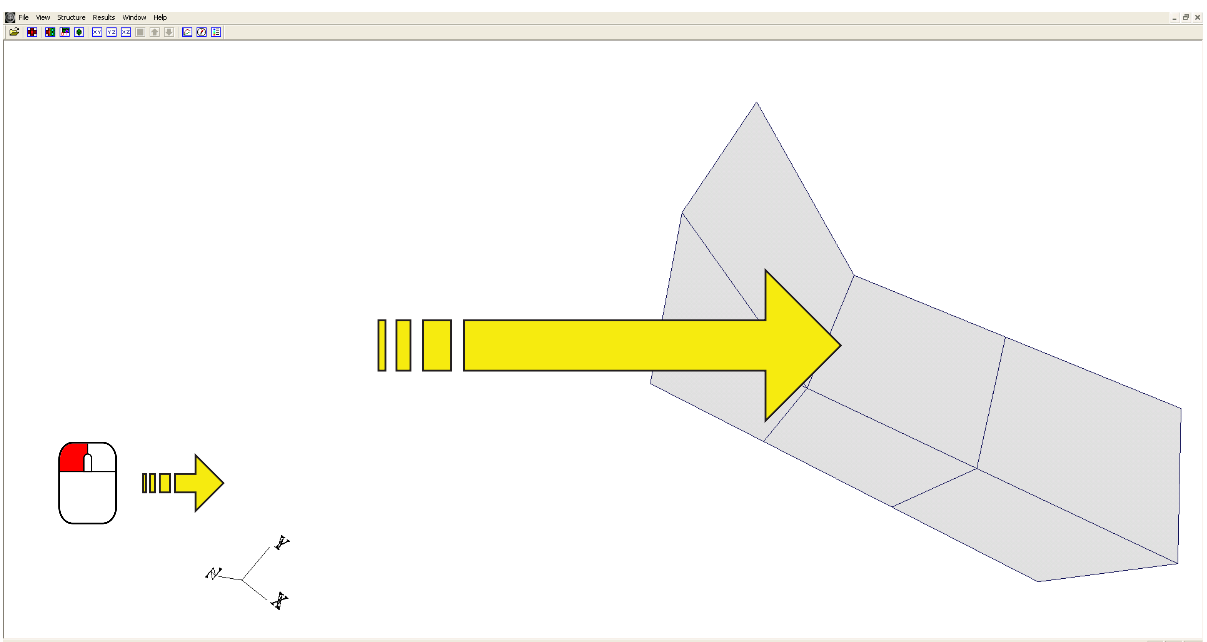


Figure 3.6: Left Mouse Button Usage in Model Space

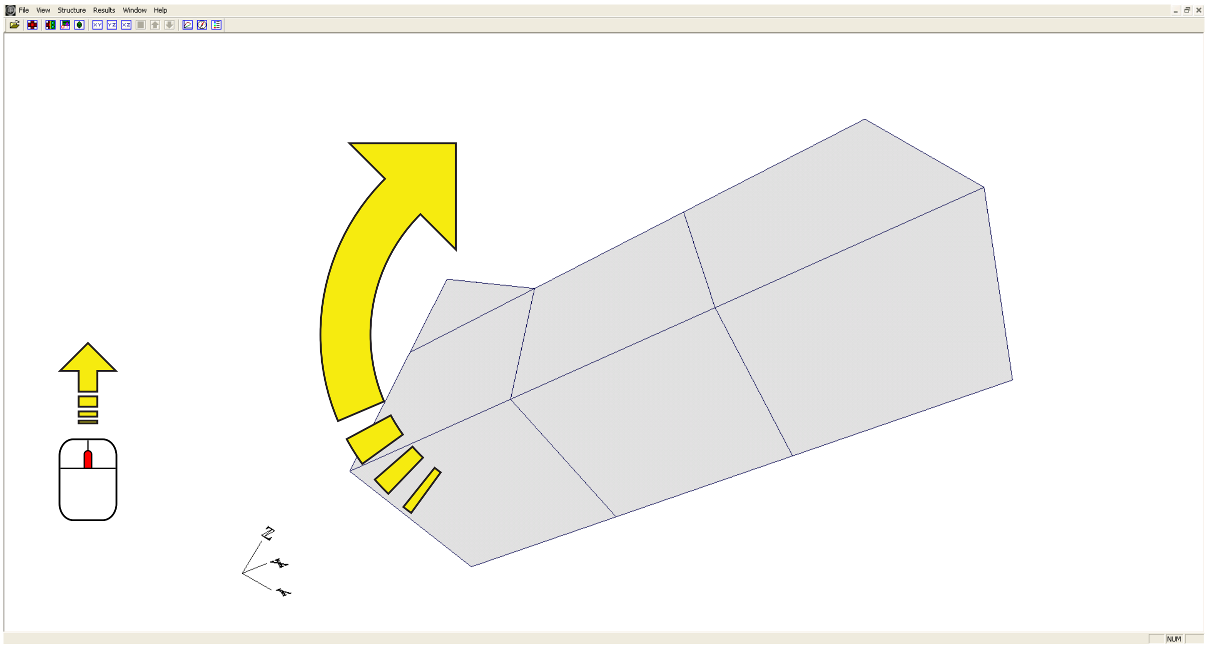


Figure 3.7: Track Wheel Mouse Button Usage in Model Space

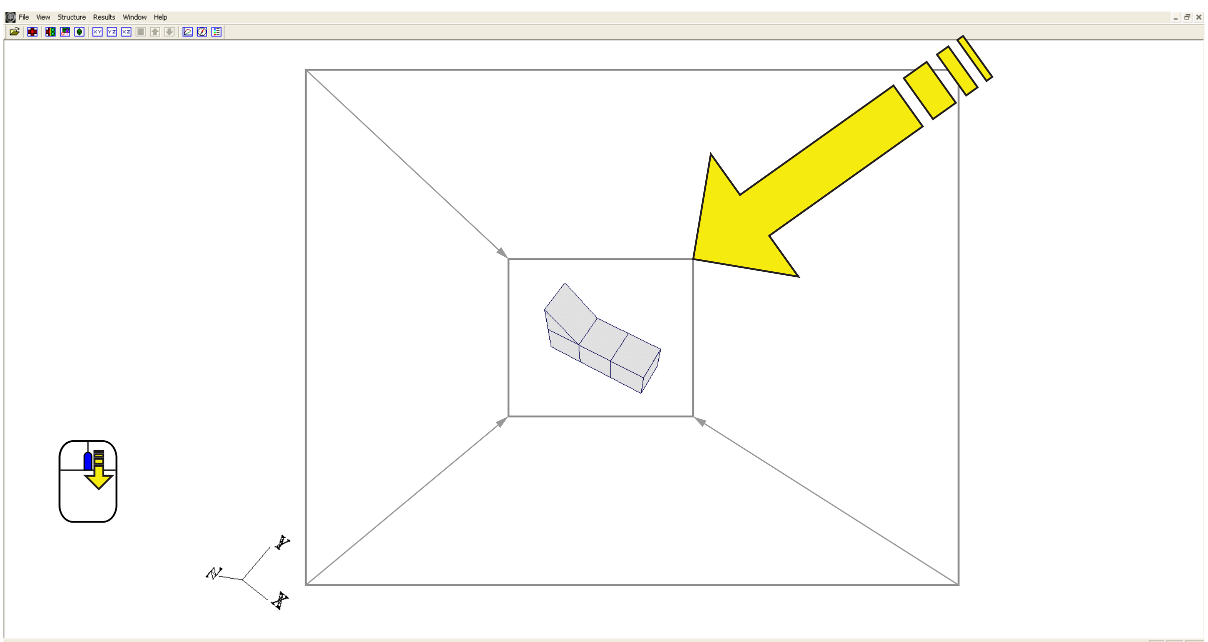


Figure 3.8: Track Wheel Scroll Usage in Model Space

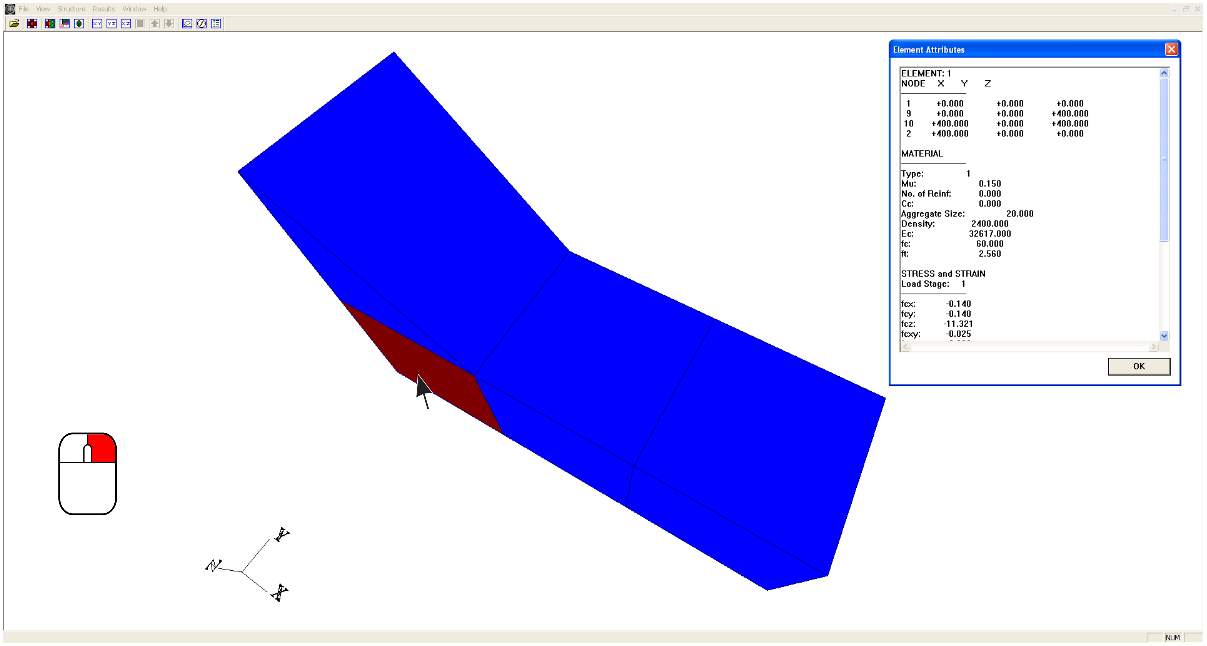


Figure 3.9: Right Mouse Button Usage in Model Space

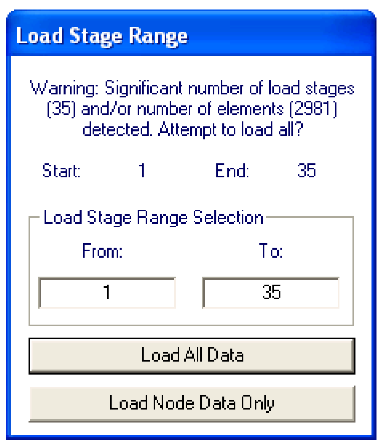


Figure 3.10: Load Stage Range Dialog

Program	Memory Factor[1]
VecTor2	5000
VecTor3	2500
VecTor4	1000
VecTor5	1000
VecTor6	5000

[1] Memory Factor = (Load Stages · Total Element Count)

Table 3.2: Memory Factor Values

Using the *Load Stage Range* dialog, a set range of load stages may be specified by the user. Janus attempts to sequentially open and read the requested expanded analysis output files, beginning with the expanded analysis data file corresponding with load stage number in the "From" edit box. Upon successfully reading the entire analysis data file, Janus incrementally searches for the next file in the local file directory. In the case that an analysis output file is incomplete or missing, Janus will exit the file reading procedure and present the available structural, load and analysis output data as is. Janus also features a simplified and less memory-intensive mode of post-processing, providing the option to solely load the nodal displacement and reaction data for the selected load stage range. See Figure 3.11 for a conceptual decision flowchart of Janus loading options.

It is important to note that Janus is capable of displaying VecTor models without any expanded analysis output files present in the job file directory. This permits users to immediately preview their models as the VecTor program has interpreted them, even prior to any analysis output files being generated. This program feature may be useful for verifying model and analysis parameters at the onset of executing VecTor analyses which may require significant computation time and/or processing power to converge and subsequently run to completion. Additionally, it allows for efficient troubleshooting of various modelling issues, giving users the opportunity to preemptively inspect their model and calibrate their job, structure and load specifications as necessary - saving both time and effort through the course of their investigation.

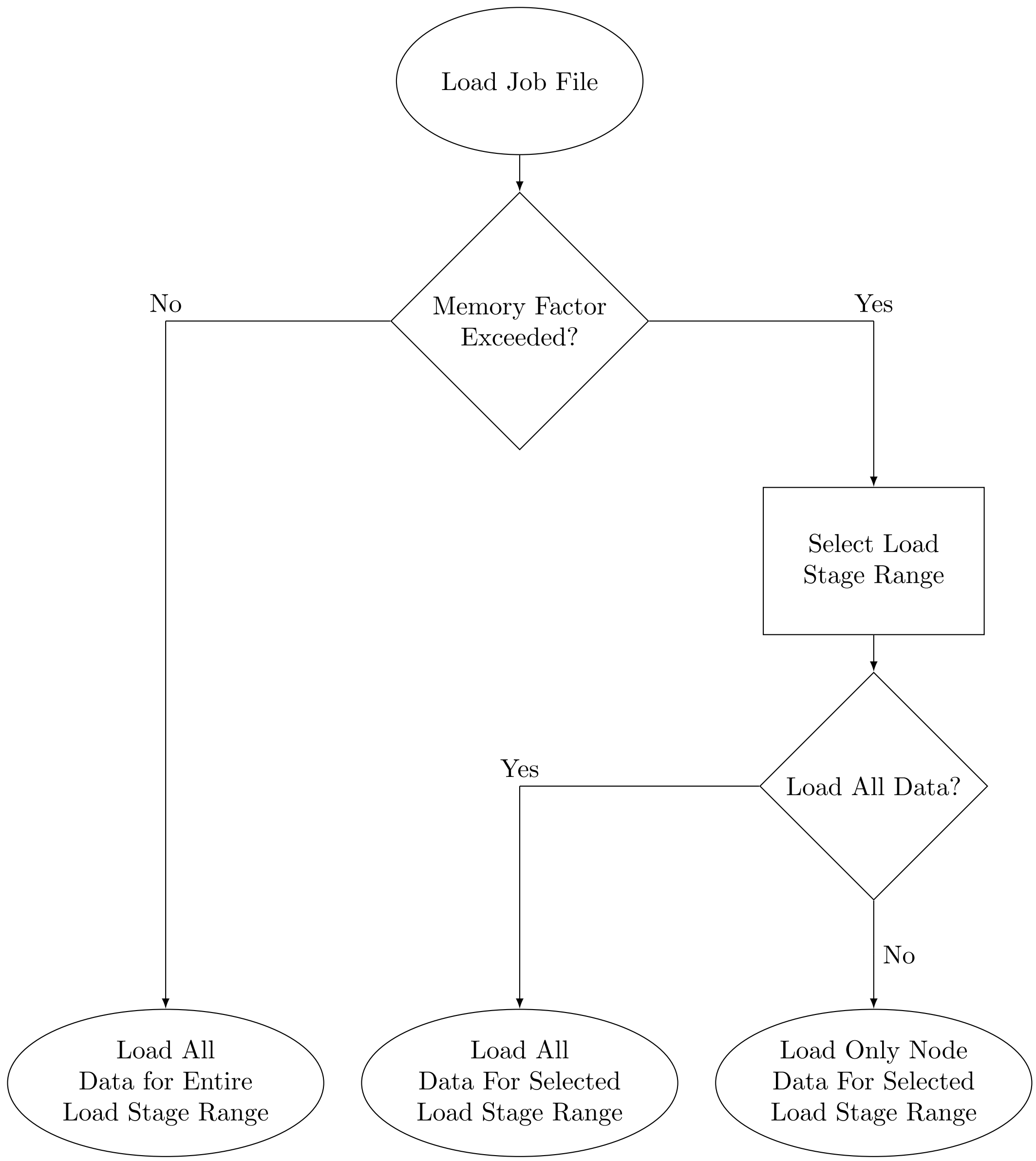


Figure 3.11: Loading Option Decision Flowchart

3.5 File Options

The **File** Menu, shown in Figure 3.12 below, provides standard options for overall program control and file interaction. The available options are described in the following subsections.

Figure 3.12: File Menu

3.5.1 Open

File ▷ *Open* invokes the standard MFC *Open* dialog. The *Open* dialog allows users to navigate through the computer file directory and select a VecTor job file to be read by Janus. Based on the specified VecTor model type and file naming convention specified in the job file, Janus will search the immediate file folder location for the expanded structure file and expanded load file(s) associated with the VecTor job. Depending on the specified load stage conditions, Janus may also search the local directory for expanded analysis output files matching the output file naming convention and load stage range as stated in the job file. Lastly, VecTor job files may also be opened in Janus via “drag and drop”. In order to open the VecTor model, the desired job file icon is clicked and dragged over Janus program window, then released. By reading the job file dropped into the program window, Janus will perform equivalent file opening operations as selecting a job file via the *Open* dialog.

3.5.2 Close

File ▷ *Close* terminates the currently active child window in Janus.

3.5.3 Print

Janus features basic functionality to allow users to send the current screen view to a printing device. By selecting **File** ▷ *Print*, a utility *Print* dialog appears to allow users to select and enter the desired print options.

3.5.4 Print Preview

File ▷ *Print Preview* presents the standard *Print Preview* dialog, allowing users to access a virtual preview of the current program window printed on a page based on the current print setup options.

3.5.5 Print Setup

Standard print and page setup options may be selected by accessing **File** ▷ *Print Setup*. Changes made within the *Print Setup* utility dialog will be reflected in subsequent print and print preview settings.

3.5.6 Recent Job Files

For convenient access, Janus maintains a list of the five most recently opened VecTor job files in the *File* Menu.

3.6 View Options

Janus provides a variety of options for the user to modify the camera view of the OpenGL model as it is presented on-screen. The typical **View** drop-down menu is shown in Figure 3.13.

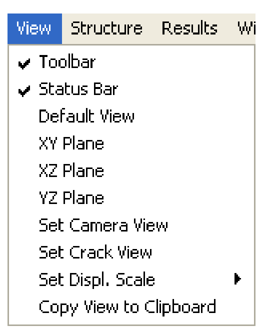


Figure 3.13: View Menu

3.6.1 Toolbar and Status Bar

For viewing convenience, the toolbar and status bar areas may independently enabled or disabled by selecting **View** ▷ *Toolbar* and **View** ▷ *Status Bar*, respectively. The toolbar and status bar are both enabled by default.

3.6.2 Default View

Selecting **View** ▹ *Default View* restores the camera view to the original Global Model View position and rotation that is initially specified upon opening a VecTor model. The default view setting is dependent on the VecTor model type being displayed - 3D models created in VecTor3 and VecTor4 revert back to an skewed perspective view, 2D VecTor2 and VecTor5 models return to an x-y plane view, and VecTor6 resumes an x-z plane view. By default, all VecTor models are presented in Janus using a perspective projection.

3.6.3 Plane View

View ▹ *XY Plane*, **View** ▹ *XZ Plane*, and **View** ▹ *YZ Plane* provide corresponding options to change the rotation of the camera to the requested orthogonal orientation.

3.6.4 Custom Camera View

The **View** ▹ *Set Camera View* option invokes the *Set Camera View* dialog, presented in Figure 3.14. It is used to provide complete customization for a variety of camera view characteristics: position, rotation, zoom, mouse-associated translation and rotation speeds, axes of rotation, and camera projection view. By entering numerical values and/or selecting the appropriate options, the user can precisely fine-tune the on-screen view in Janus to the desired specification. All camera rotation and translation commands are sequentially applied about the x-, y-, and then z-axes. A description of each dialog option is included below.

3.6.4.1 Position

Janus camera position values are entered according to the OpenGL model coordinate system, and inputs are specified relative to the current on-screen view. Hence, instructing a camera translation in the positive x-direction will position the model relatively right of centre on the screen. Accordingly, specifying a negative x-direction moves the model to the left. In a congruent fashion, positive and negative y-direction translations correspond with moving the model to positions relatively above and below the on-screen line of sight, respectively. An input range from -4.0 to 4.0 are reasonably suggested in order for the model to remain visible on-screen after confirming changes to the on-screen view. With z-axis translations, the camera is positioned relative to the origin point of the coordinate system. It should be noted that the camera view frustum orientation is fixed in the positive z-direction - all rotations and/or translations are imposed upon the model itself, while the frustum remains stationary. Hence, a z-axis

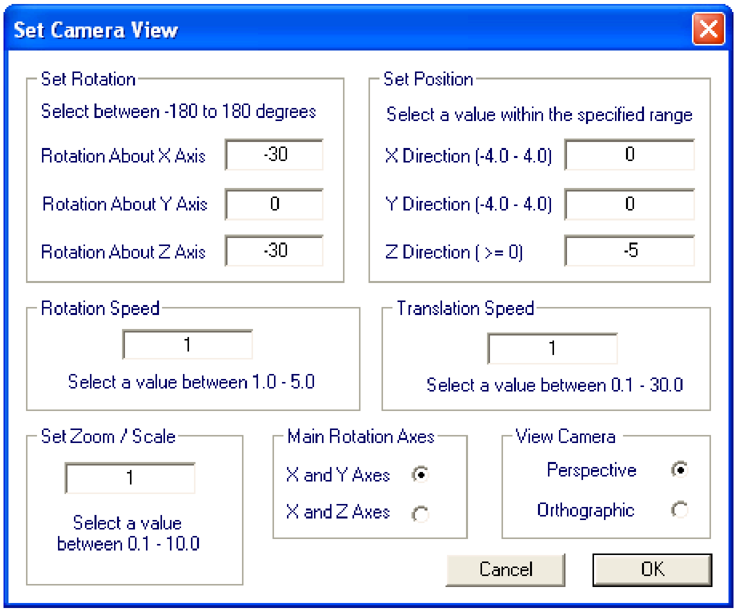


Figure 3.14: Set Camera View Dialog

position less than zero is expected in order for the model to be positioned between the near and far clipping planes of the view frustum (and remain visible to the user). By default, the camera view is positioned at a z-axis position of -5.

3.6.4.2 Rotation

Janus view rotation specifications follow a right-hand rule coordinate system, with a positive angle producing a counter-clockwise rotation about the positive direction of the axis, and vice versa for a negative angle. As demonstrated in the *Set Camera View* dialog, a specified rotation angle between -180.0 and +180.0 degrees may be entered for each axis of rotation. Available axes for specifying camera rotation are specific to each VecTor model coordinate system. VecTor3 and VecTor4 models permit rotation in all x-, y-, and z-axes, VecTor2 and VecTor5 allow rotations about the z-axis, and VecTor6 models may be rotated about y-axis. For 3D finite element model types, the main rotation axes are selected via the *Rotation Axes* options, described below in Subsection 3.6.4.5.

3.6.4.3 Scale

Janus users may also specify the scale that the VecTor model is drawn to. By default, model dimensions are scaled by a numerical factor of 1.0. For viewing purposes, users may specify a factor greater than 0 and less than 1.0 to uniformly shrink the model, or greater than 1.0 to magnify the model.

3.6.4.4 Rotation and Translation Speeds

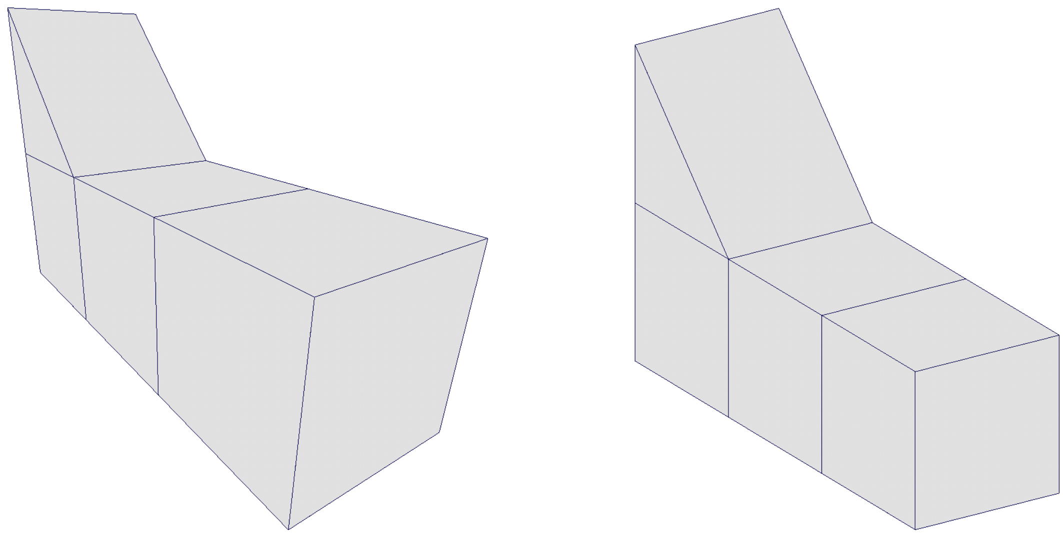
For mouse-controlled rotation and translation specifications (discussed in Section 3.2), the speed at which the model is rotated and/or translated may be customized by specifying numerical factors for each trait. Values greater or less than 1.0 will respectively speed up or slow down the default rate of lateral and/or rotational view displacement.

3.6.4.5 Rotation Axes

Mouse-controlled rotations allow for 3D VecTor models to be rotated about all available axes in a tactile fashion; however, only two rotation axes are specified for the model to rotate longitudinally (up/down) about while moving the mouse in an up and down motion. The model is rotated about the tertiary axis through lateral (left-right) movements of the mouse. Using two main rotation axes helps to establish a dominant model orientation that the user can use to freely rotate and view the model in an intuitive manner. As well, this approach coincides with the finite element models typically employing an axis of symmetry or “ground” surface as a defining plane of reference. Providing alternatives to use either combination of x- and y-axes or x- and z-axes as the main rotation axes gives users versatility in customizing the model rotation controls to the specific model being viewed.

3.6.4.6 Projection Options

Janus provides two options for the type of projection used to display the model on-screen, perspective or orthographic (shown in Figure 3.15). The perspective projection provides the illusion of depth in the view, with finite elements in closer proximity to the camera appearing larger in size than those that are further behind. In contrast, the orthographic projection eliminates any such skewing of shapes and instead presents all visible elements regardless of apparent depth from the camera (in the context of the view frustum, the near and far clipping planes have congruent dimensions). The orthographic projection consistently provides a scale representation of the VecTor model, independent of changes to the camera position and orientation. However, for most practical post-processing purposes, the perspective projection lends a more realistic and intuitively expected representation of a 3D finite element model to the human eye. Hence, the perspective projection view is used as the default setting when opening any VecTor model. Both perspective and orthographic projection options are available for the user to select when customizing the camera view.



a) Perspective Projection b) Orthographic Projection

Figure 3.15: Perspective vs. Orthographic Projection View Settings

3.6.5 Deformation Scale

When viewing structural deformations, the default scale of node displacements may not be appropriate for the intended context or scale of the VecTor model. In some cases, the computed displacements may be too small to be discernible on a macroscopic modelling basis. On the other hand, nodal displacements that are exceedingly magnified may unrealistically distort the represented physical response of the structural model. By selecting **View** ▹ *Set Deform. Scale* ▹ *Select Scale Factor*, Janus provides facilities for allowing users to enter a numerical scale factor used to multiply all nodal displacements. Entering a value between 0 and 1 will reduce nodal displacement values, while a value greater than 1 will conversely magnify them. Alternatively, the default deformation scale factor of 30 may be resumed by selecting **View** ▹ *Set Deform. Scale* ▹ *Default Scale Factor*.

3.6.6 Crack View

When viewing crack data from VecTor programs which support crack pattern data output, several crack pattern options may be customized by the user via **View** ▹ *Set Crack View*. This invokes the *Set Crack View* dialog, as shown in Figure 3.16. The calibration of crack display options such as varying crack width and crack length may be conducted through this dialog window. By default, Janus displays crack lengths as a function of the crack angle projected on the face of element, and variable line width based on ranges of element crack width parameter values. For detailed discussion describing how Janus displays crack patterns, refer to Section 3.8.

3.6.6.1 Alternative Crack View

Given the diverse range of VecTor program modeling capabilities, the default setting of using varied OpenGL primitive line widths may not be able to accommodate for adequately differentiating crack widths on a global scale, or the overall crack width pattern may even be obscured by the resolution capability of the display device being used. In certain modelling scenarios, the *Alternative Crack View* options can be used to provide a visual comparison of crack widths in a more tangible arrangement than representing crack widths using computer-generated line widths. Selecting the *Varying crack length and width* option enables users to specify how Janus displays the crack lengths and widths on element faces.

Instead of the crack length being displayed solely as a function of the projected crack angle and the dimensions of the element face, users may opt for crack lengths to be abstracted as a function of the crack width multiplied by a specified *Crack Length Scale* factor. The default setting applies a crack length scale factor of 1.0, but a custom value may be selected by the user. Additionally, users may also enter a *Crack Width Scale* factor, which magnifies all rendered line widths by the entered value.

3.6.7 Section and Layer View

For ease of access, several buttons in the toolbar area are used for modifying how the finite element model is viewed, depicted in Figure 3.17 below. Depending on the applicable display context of the type of VecTor model being displayed, different toolbar buttons may be active or disabled. See Table 3.3 for a VecTor-specific list of available Section View and/or Layer View related features.

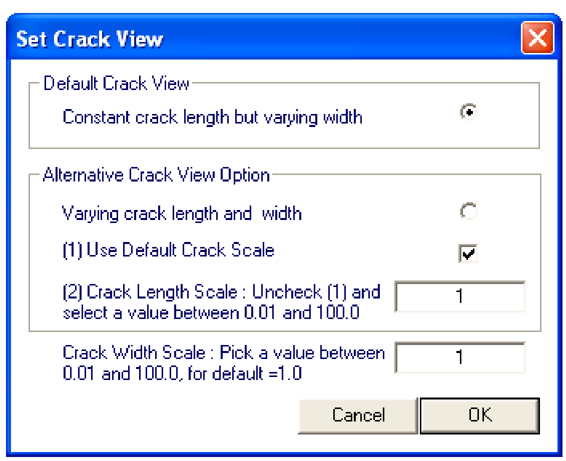


Figure 3.16: Set Crack View Dialog

Figure 3.17: Section and Layer View Toolbar Buttons

Program	*XY Section*	*YZ Section*	*XZ Section*	*Layer View*	*Section Up*	*Section Down*
VecTor2	✓	✓	✓	✗	✓	✓
VecTor3	✓	✓	✓	✗	✓	✓
VecTor4	✓	✓	✓	✓	✓	✓
VecTor5	✗	✗	✗	✓	✗	✗
VecTor6	✓	✗	✗	✗	✓	✓

Table 3.3: Section and Layer View Functionality per VecTor program

3.6.7.1 Section View

The toolbar area buttons *XY Section*, *YZ Section*, and *XZ Section* correspond with activating Section View for the chosen pair of axes. Upon activation of a section view button, the user is prompted to enter a coordinate value for the third, out-of-plane axis. The input coordinate value is based on the user-chosen VecTor model coordinate system. As such, the requested coordinate must lie between the minimum and maximum bounds of the model in the axis of interest.

The resulting on-screen view is a planar representation of the model at the requested coordinate position. All intermediate finite element sections are rendered using linear interpolation of element face and edges that are determined to lie on or intersect the section plane. Linear elements such as truss elements will appear in Section View if the element coordinates either lie on or are intersected by the specified section plane. Accordingly, single-point elements such as ring bar elements and link elements will also appear in Section View, but only if the element nodal coordinate is precisely intersected by the section plane. This rule holds true for both undeformed and deformed model view modes. See Figure 3.19 for reference. Section view may be combined with a variety of result modes, including: Hotspot, Deformations, Crack Pattern, contour mode, etc. Node-specific contour modes like colour-gradient reactions and deformations (discussed in Section 3.8), will not be displayed in Section View. Similarly, nodal features such as nodal load arrow symbols and restraints (discussed in Section 3.7) will only be visible in Section View if the node coordinate is congruent with the section coordinate for the specified axis.

As illustrated in Figure 3.18, a VecTor3 example model is positioned between 0 and 1200 in the x-direction. Thus, only x-coordinate section values between 0 and 1200 are valid for generating a section across the y- and z-axes. Selecting the *YZ Section* button and entering an x-coordinate of 150 yields an intermediate section view of the hexahedral and wedge elements which cross the x-coordinate of 150.

In addition to displaying a new and distinct OpenGL rendering of the VecTor model, the camera rotation values are also modified to match the plane of the selected section orientation.

3.6.7.2 Layer View

VecTor4 and VecTor5 possess unique layered analysis capabilities in addition to providing model global response characteristics. Elements are subdivided into distinct reinforced concrete and discrete reinforcement layers (as determined by material type definitions), each demonstrating unique structural response parameters as a result of globally applied loads. Figure 3.20 illustrates a how layered analysis of finite elements is conceptually employed in VecTor4 and VecTor5. The resulting non-linear distribution stress and strain parameters through the layered depth of an element are provided as additional sections of data in the expanded analysis output files. The *Layer View* button in Janus allows users to view the resulting values of the sectional analysis for a specified element or member.

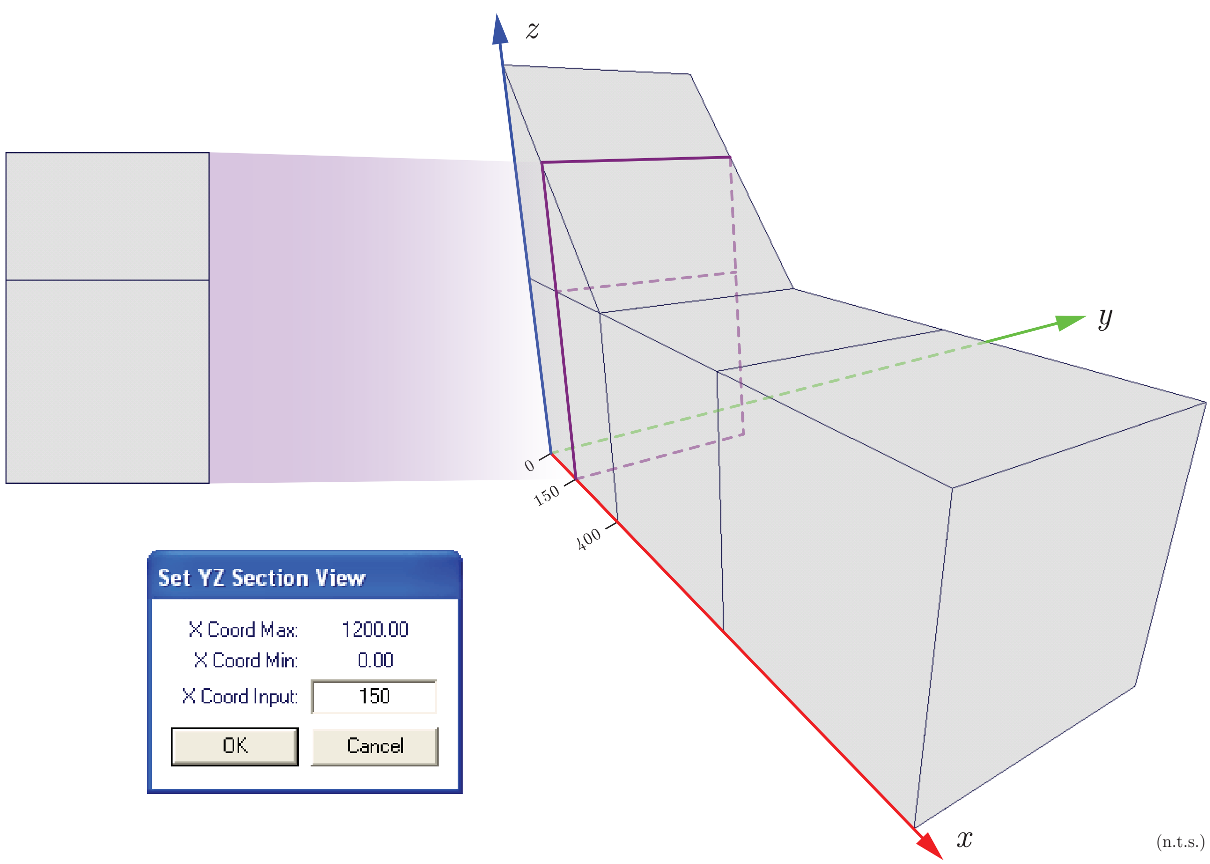
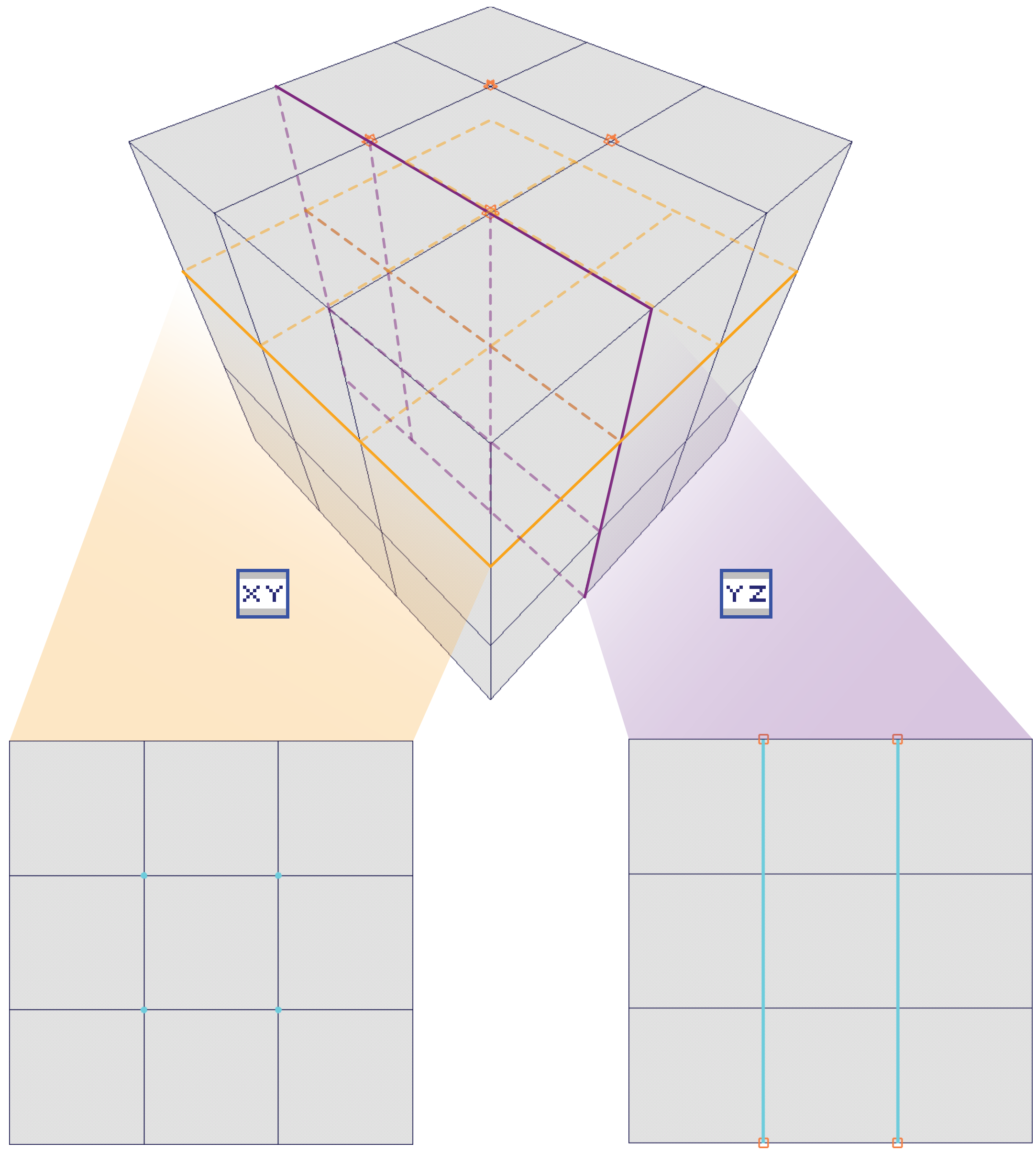


Figure 3.18: Section View of a Simple VecTor3 Model



a) XY Section Through Truss Elements

b) YZ Section On Truss Elements

Figure 3.19: Simple VecTor3 Model Truss Elements On/Through Section View

Once the *Layer View* button is pressed, users are prompted to identify the VecTor4 shell element or VecTor5 output member of interest by selecting the element number from the provided drop-down list. Once an element is selected, the sole element or member section will be re-drawn in Janus - isolated from the rest of the model, as well as magnified and centred in model space for ease of viewing. Distinct individual faces and lines representing concrete and discrete steel layers will be drawn across the thickness of the element/depth of the section. Janus result display functions, as discussed in subsequent sections, are available in the context of presenting sectional response values. See Figure 3.21 for an illustration of selecting a VecTor4 shell element for display using the *Layer View* toolbar button. Alternatively, see Figure 3.22 for a demonstration of Layer View using a VecTor5 example model.

3.6.7.3 Section Up/Down

While Section View is enabled, the *Section Up* and *Section Down* buttons may be used to increase or decrease the current section coordinate to the next nearest model node coordinate for the currently selected out-of-plane axis. For example, referring again to Figure 3.18, the *YZ Section* view is displayed according to an intermediate section x-coordinate value of 150.

As illustrated in Figure 3.23, pressing *Section Up* increases the section x-coordinate to the next-highest node coordinate value of 400, while *Section Down* would decrease it to 0. The section view is then re-drawn at the new coordinate. Any selected view, result and structure settings will persist after the section coordinate is modified via *Section Up* or *Section Down*. Congruent functionality exists for increasing/decreasing the y-coordinate and z-coordinate in the *XZ Section* and *XY Section* views,

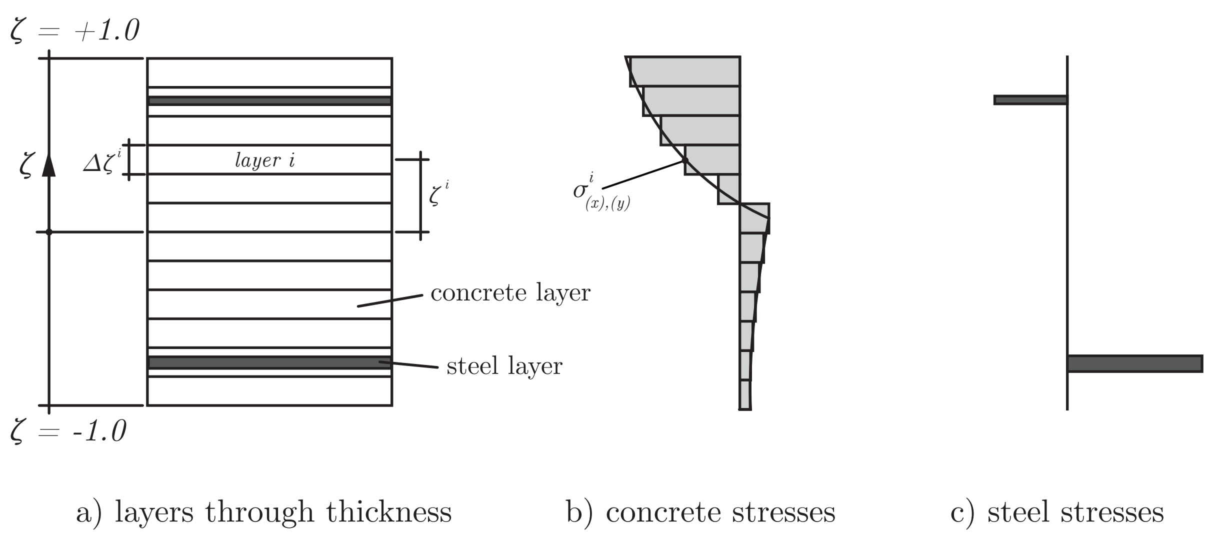


Figure 3.20: Layered Analysis (adapted from Polak and Vecchio, 1993)

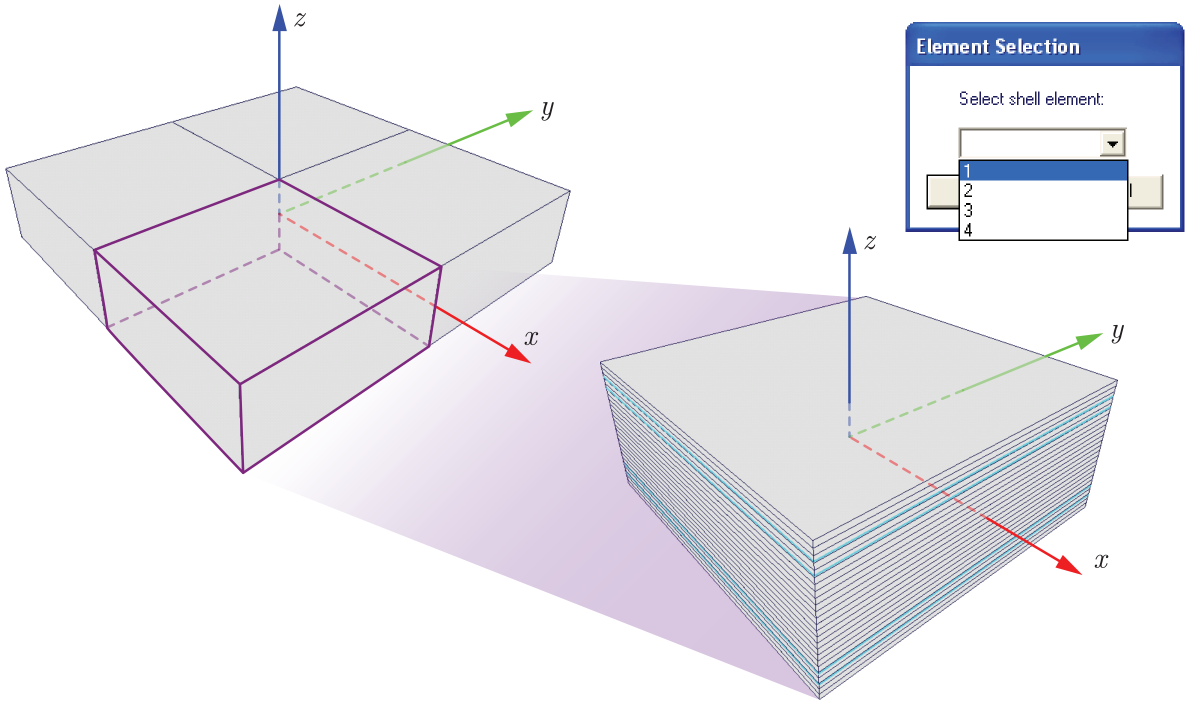


Figure 3.21: Layer View of a Simple VecTor4 Model Shell Element

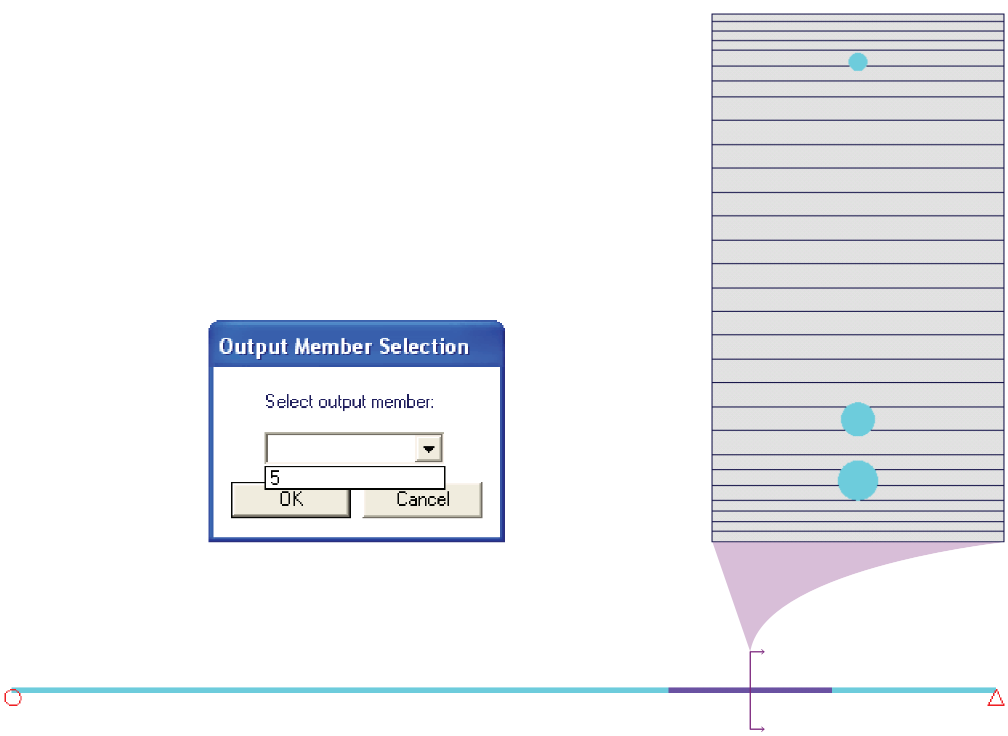


Figure 3.22: Layer View of a Simple VecTor5 Model Output Member

respectively. *Section Up* and *Section Down* will not modify the section coordinate beyond the respective maximum and minimum coordinate values of the model. These two buttons are useful for traversing through the nodal layers of the model and viewing parameters of interest such as: nodal loads, nodal restraints, internally defined truss elements, and orthogonally oriented faces of interior solid elements not visible from Global Model View.

3.6.8 Toggle Features

Within the toolbar area, several buttons are dedicated to toggling view-related features on and off within the model space, as shown in Figure 3.24. In order for any of the toggle feature toolbar buttons to operate as intended, an applicable non-default view feature needs to be previously selected or enabled. Table 3.4 provides a brief overview of each toggle feature.

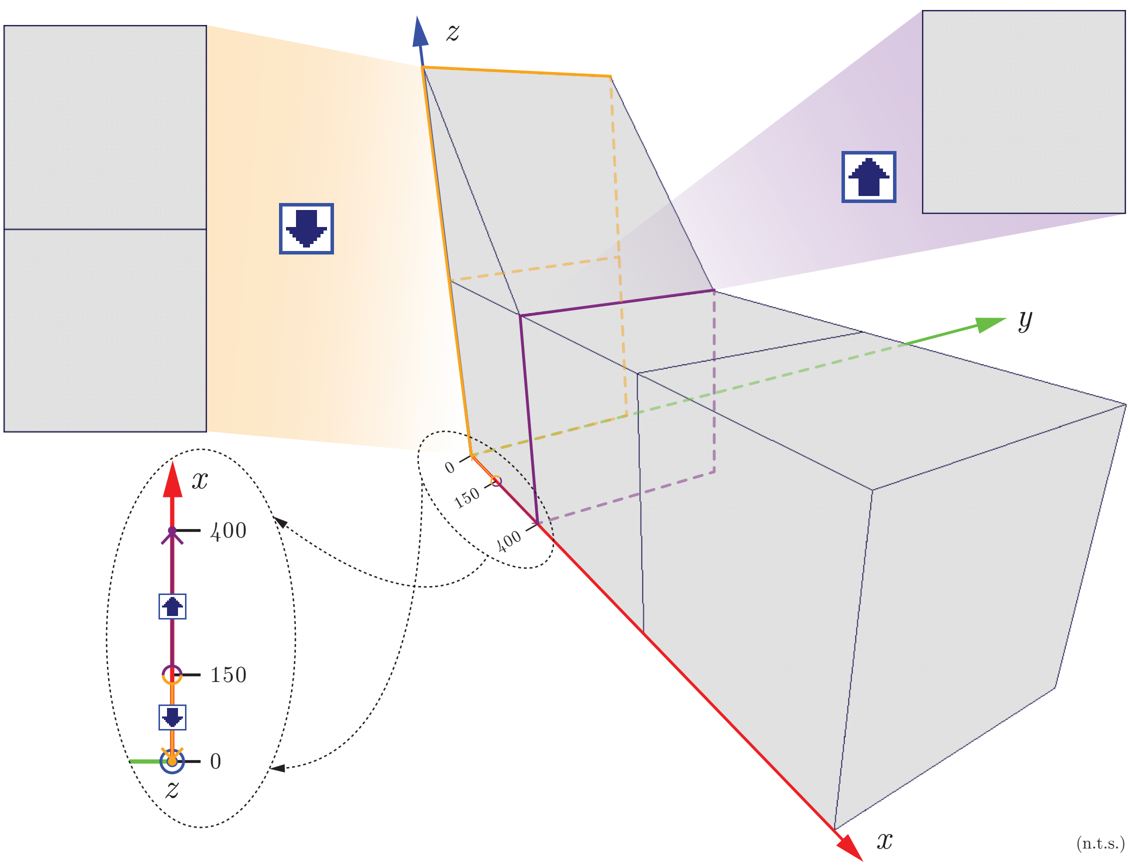


Figure 3.23: Section Up and Section Down Functionality

Figure 3.24: Toggle Features Toolbar Buttons

Toggle Face Fill 	*Toggle Face Fill* alternates the model solid element face colours between the most recent face fill-modifying display mode and the default face fill colour. The face fill switching feature is enabled for the following model states: • Material mode • Hotspot mode • Contour mode • Custom colour wireframe/nodes • VecTor4 Gauss Point mode
Toggle Node Feature 	*Toggle Node Feature* alternates rendering and erasing the most recently invoked node feature. Node features include the following: • Nodal Loads • Nodal Restraints
Toggle 3D View 	*Toggle 3D View* switches the on-screen view between the Global Model View and the most recently enabled Section View (*XY Section*, *YZ Section*, *XZ Section*) or *Layer View* feature. If the user intends to resume the latest section or layer view by clicking the *Toggle 3D View* button, the most recently entered section coordinate or element number is recalled as well.

Table 3.4: Toggle Feature Functionality

3.7 Structure Options

The **Structure** drop-down menu (shown in Figure 3.25) is used to provide users with options for customizing model display options in Janus.

3.7.1 Toggle Elements

For viewing complex VecTor models in Janus, users may wish to categorically enable or disable the rendering of certain types of elements for ease of viewing and isolating the element type(s) of interest. For example, with VecTor3 models, the exterior faces of solid elements such as hexahedral and wedge elements could potentially obscure any internal truss and link elements from the camera view. Conversely, in

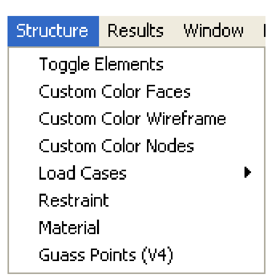


Figure 3.25: Structure Menu

other modelling scenarios, dense arrangements of truss and/or bond elements could also produce similar viewing disruptions. By selecting **Structure** ▷ *Toggle Elements*, users may access the *Toggle Elements* dialog as shown in Figure 3.26. The *Toggle Elements* dialog provides facilities for users to selectively switch the display of certain element types on or off at their convenience. The applicable element option check boxes are activated according to the type of VecTor model being viewed. For example, Figure 3.27 demonstrates the resulting effect of selectively enabling some of the available element options for viewing a VecTor3 model in Janus. Although the element types are progressively disabled from a) through d) for illustration purposes, the *Toggle Elements* dialog allows any combination of the available element type options to be selected or deselected at a time.

3.7.2 Custom Colour Options

Janus users may modify the default element face colour of the finite element model by selecting **Structure** ▷ *Custom Color Faces*. **Structure** ▷ *Custom Color Wireframe* and **Structure** ▷ *Custom Color Nodes* also modify the default element colour of the model, and re-draw the model as a wireframe or series of nodes using the newly chosen colour as well. Selecting any of these custom color options invokes a standard colour selection dialog, as shown in Figure 3.28. The dialog allows users to choose from a default array of basic colours, or customize their own using the provided colour palette and/or numerical input parameters. Upon confirming a new colour, the VecTor model is re-rendered with the solid elements/wireframe/nodes of the model coloured using the selected colour. See Figure 3.29 for an example of the described custom colour options using a simple VecTor3 model.

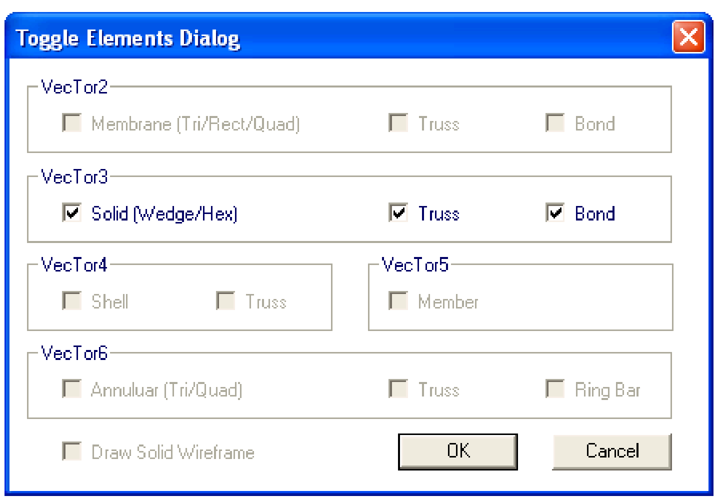
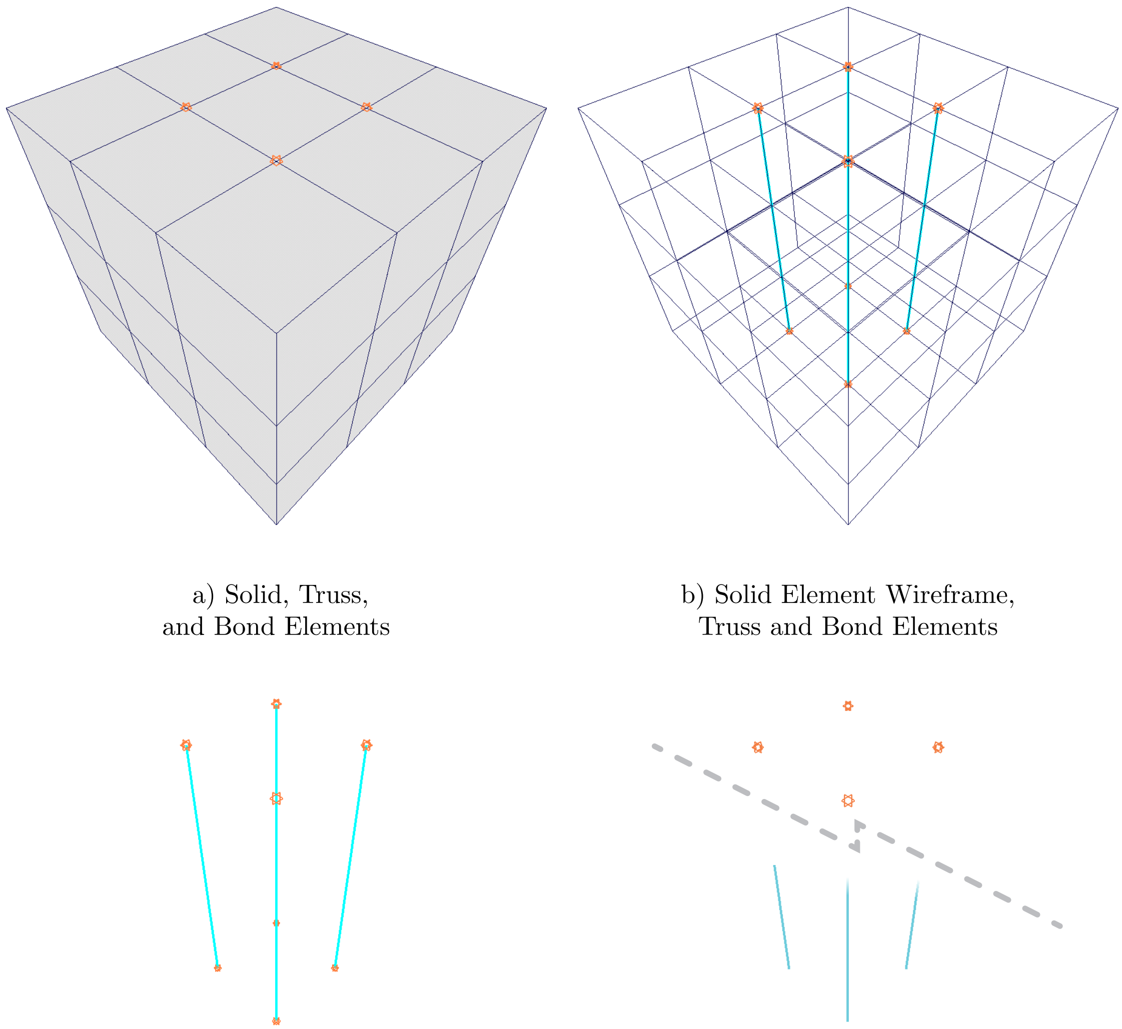


Figure 3.26: Toggle Element Dialog for a VecTor3 Model



a) Solid, Truss, and Bond Elements

b) Solid Element Wireframe, Truss and Bond Elements

c) Truss and Bond Elements

d) Truss or Bond Elements

Figure 3.27: Toggle Element Functionality Using a Simple VecTor3 Model

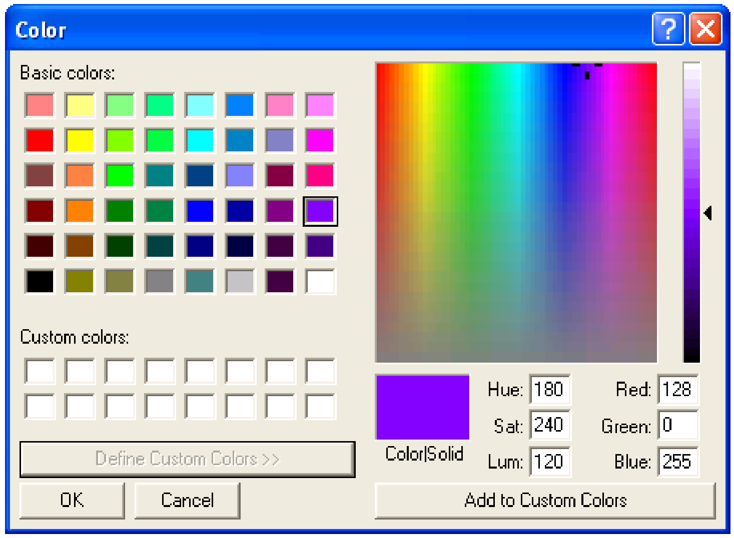
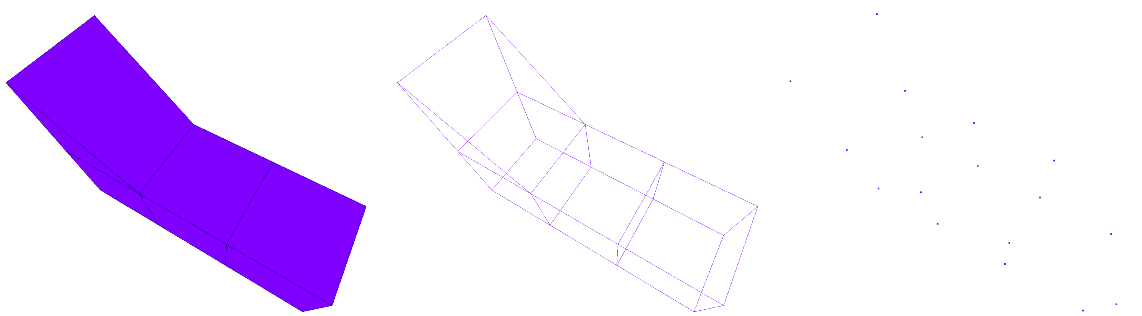


Figure 3.28: Colour Dialog



a) Custom Colour Faces b) Custom Colour Wireframe c) Custom Colour Nodes

Figure 3.29: Custom Colour Options Using a Simple VecTor3 Model

3.7.3 Load Cases

Janus allows users to view nodal loads/moments and nodal displacements/rotations as they are applied to the modelled structure. **Structure** ▷ *Load Cases* presents an option for viewing each active load case, numerically identified 1 through 5 (corresponding to the load case entries listed in the job file). Once an active load case is selected for viewing, load arrows will be rendered at each node specified in the expanded load file. For distinct visual differentiation, all load arrows and other load-related symbols are rendered in pink outlines. When the load is listed in the *Load Case* dialog (see Figure 3.31), the arrow symbols are filled with a solid colour corresponding to its colour-coded entry. A detailed description of the *Load Case* dialog is provided in Subsection 3.7.3.4.

3.7.3.1 Loads and Displacements

For all VecTor models, nodal loads and displacements in the x-, y-, or z-directions are denoted using single-headed arrow symbols oriented by their assigned degree of freedom and pointing in the corresponding direction of magnitude. For example, a nodal load applied in the positive x-direction will be represented by an arrow shape also pointing in the positive x-direction in Janus model space. When displaying 3D VecTor models (VecTor3 and VecTor4), Janus renders the arrow symbols in two orthogonal directions to maximize visibility in 3D space. Accordingly, load arrow symbols for 2D VecTor model types will only be rendered on the plane that the structure lies on. See Figure 3.30 a) and b) for examples of 2D and 3D load arrows, respectively.

3.7.3.2 VecTor4 Moments and Rotations

VecTor4 permits nodal moment and rotations applied about the local in-plane axes of a shell element. In order to accommodate the freely chosen orientation (and local axes) of 3D shell elements in VecTor4, applied moments and rotations are represented in Janus with double-headed arrow symbols oriented perpendicular to the plane of rotation (See Figure 3.30 c)). Following the right-hand rule notation, the direction of the double-headed arrow signifies the rotational direction that the moment/rotation is applied about the node.

3.7.3.3 VecTor5 Moments and Rotations

In contrast to VecTor4 applied moments, VecTor5 only permits in-plane applied nodal moment and rotational loads as a 2D frame analysis program. Since VecTor5 models are globally viewed on a planar basis, all load symbols must be visible in-plane as well. Due to these defining characteristics, nodal moments/rotations of VecTor5 models in Janus are represented using a circular arrow with the arrow heads denoting the direction of rotation about the node. See Figure 3.30 d) for a model example of a VecTor5 nodal moment/rotation load arrow.

3.7.3.4 Load Case Dialog

The *Load Case* dialog is the accompanying window for providing data and visual association for a load case being displayed in Janus. When an active load case is selected through **Structure** ▷ *Load Cases*, an instance of the *Load Case* dialog appears on-screen. The dialog is capable of providing load data for 20 distinct load conditions at a time, each attributed to a distinct arrow fill colour. A load condition is described as a unique combinations of loads/moments or displacements/rotations applied at a node or set of nodes. Load conditions are determined through an internal sort procedure within Janus, sequentially numbered starting from 1.

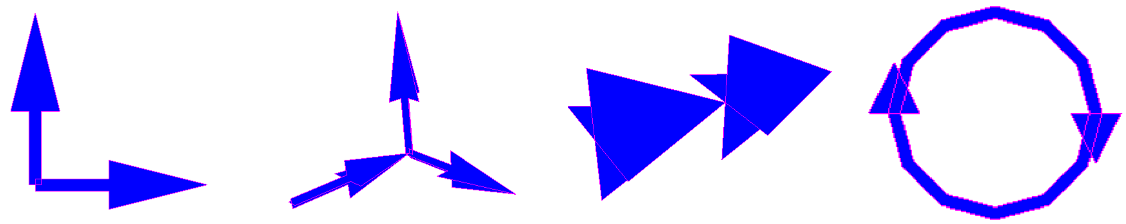
For example, referring to Figure 3.31, the *Load Case* dialog displays three different load conditions, demonstrated by distinct point load combinations in the x-, y-, and z-directions, and identified as load conditions 1-3. Each load condition is associated with a distinct arrow fill colour, providing a visual method of attributing the numerical load condition data presented in the *Load Case* dialog with the load arrows rendered on the VecTor model. It is important to note that the concept of numerically identified load conditions is a organizational method provided in Janus for users to navigate through when the display capacity of the *Load Case* dialog is exceeded, and bears no meaning or relation outside of the context of displaying load cases in Janus.

For VecTor models with complex load case specifications that exceed the 20 load condition display capacity of the *Load Case* dialog, the *Load Condition Selection* area allows users to traverse through the list of load conditions. Otherwise, the area is disabled since all load conditions are capable of being displayed and listed at once. When the *Load Condition Selection* entry value is modified, the on-screen load arrows will accordingly "fill" and "empty" to consistently represent the range of 20 load conditions currently listed in the *Load Case* dialog.

Users may use the + and - buttons to incrementally increase or decrease the range of viewed load conditions, or enter a load condition number value in the edit box to directly shift the start of the viewed 20 load condition range to the requested value. For example, if there are 40 unique load conditions for the load case being viewed, the user may specify "13" to view the 20 load conditions 13 through 32, inclusive. If the requested load condition range exceeds the maximum number of load conditions found, the selected load condition will be hard-set at an upper-bound value of ((Last Load Condition) - 19). Therefore, in the same example scenario of 40 load conditions, any request for viewing a load condition value ≥ 21 would result in displaying the range of load condition 21 through 40. Similarly, inputting a load condition value less than the lower bound value of 1 will return the value of 1.

3.7.4 Restraints

Janus users may view nodal restraints assigned to the VecTor model. Selecting **Structure** ▹ *Restraints* toggles the view of global model restraints on and off. Janus expresses the restrained degrees of freedom for a node as a combination of roller, pin, and (when applicable) fixed moment symbols. Planar roller and pin symbols are used to represent restrained displacements, and fixed moment symbols are used to represent restrained rotations. See Figure 3.32 for an overview of restraint symbols and support reactions



a) 2D Nodal Load Arrows

b) 3D Nodal Load Arrows

c) VecTor4 Nodal Moment/Rotation Arrow

d) VecTor5 Nodal Moment/Rotation Arrow

Figure 3.30: Nodal Load Arrow Symbols

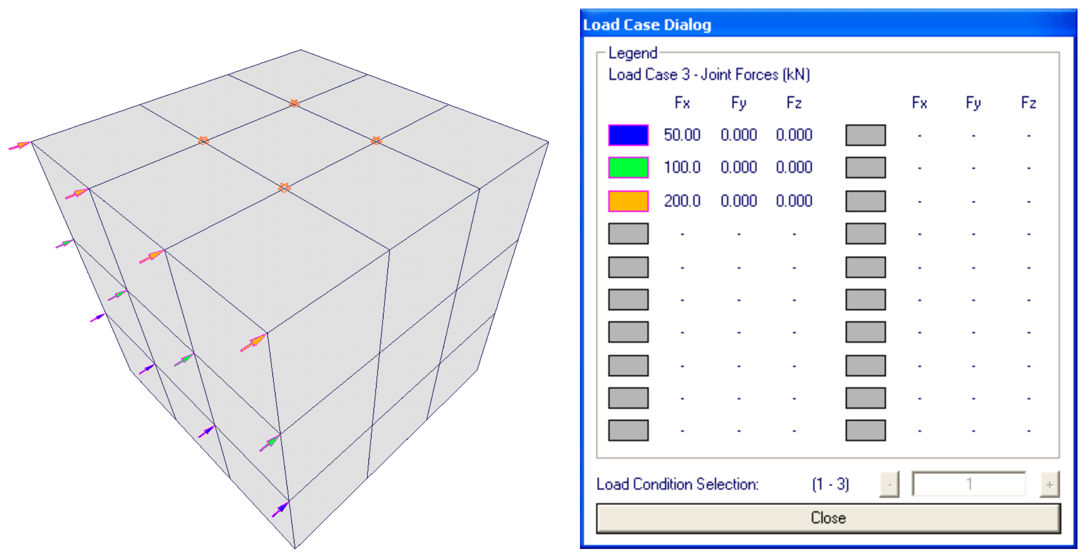


Figure 3.31: Load Case Arrows and Load Case Dialog

as they are represented in Janus.

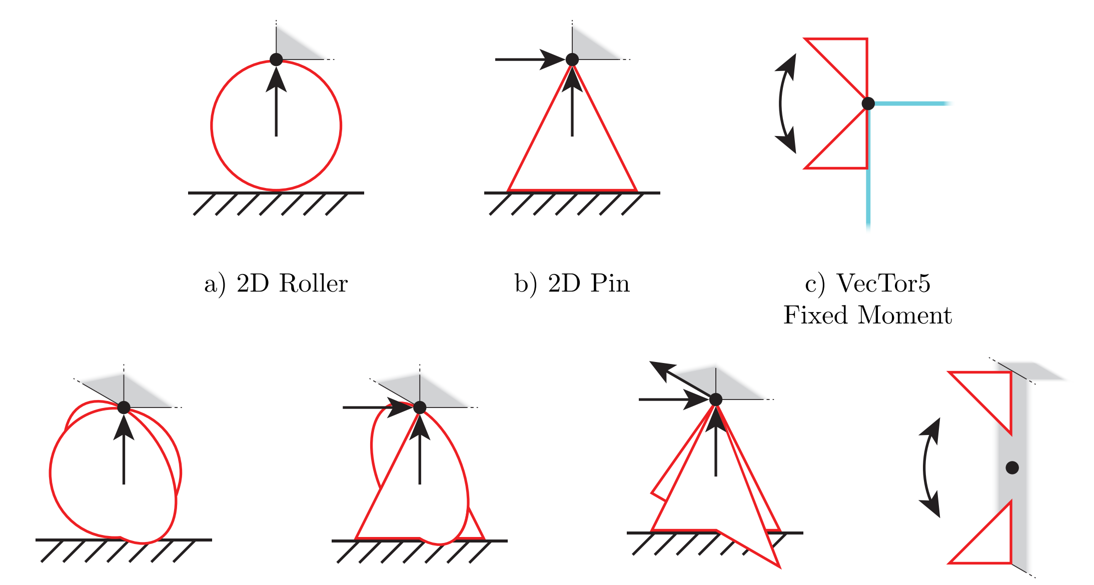
For roller restraints, the spatial position and planar orientation of the roller symbol relative to the node is used to represent the degree of freedom that is restrained. For example, if a roller symbol is positioned below or above a node in the z-direction, the node is being restrained solely in the z-axis. Similarly, the axes of the plane that a pin support symbol lies on identify the two degrees of freedom that are restrained. A pin support symbol oriented in the x-y plane implies node restraints are assigned in both the x- and y-directions. The same definition of pin and roller symbols are applied to representing restraints in 3D space as well, with combined pairs of pins and/or rollers in orthogonal directions (See Figure 3.32 d) through f)).

VecTor4 and VecTor5 are capable of accommodating restraints against applied nodal moments and rotation. In VecTor4, shell element nodes are defined at mid-depth through the element thickness.

Accordingly, Janus moment restraint symbols for these nodes are located at the top and bottom edges of the shell element - symbolizing a restrained element edge as well as the rotational direction that the node is being restrained. See Figure 3.32 g) for an example of a fixed moment restraint in VecTor4. In contrast, VecTor5 2-noded member elements are portrayed in Janus as thick width lines with uniform thickness, spanning between nodal coordinates. With no representation of element depth in the Global Model View, fixed moment symbols for VecTor5 models are drawn centred on the restrained node. Figure 3.32 c) provides an example of a VecTor5 fixed moment restraint symbol.

3.7.5 Material Mode

Another important post-processor feature within the **Structure** menu is the *Materials* option, which allows users to display the structure file element material type assignments as interpreted by the VecTor program. Using a 20-colour palette as shown in the *Legend* dialog in Figure 3.33, Janus is capable of differentiating up to 20 unique types of materials on-screen. The 20 material limit includes reinforced concrete, discrete steel, and bond material types. The material mode may be toggled on and off using the *Toggle Face Fill* toolbar button.



a) 2D Roller b) 2D Pin c) VecTor5 Fixed Moment

d) 3D Roller/Roller e) 3D Pin/Roller f) 3D Pin/Pin g) VecTor4 Fixed Moment

Figure 3.32: Restraint Symbols

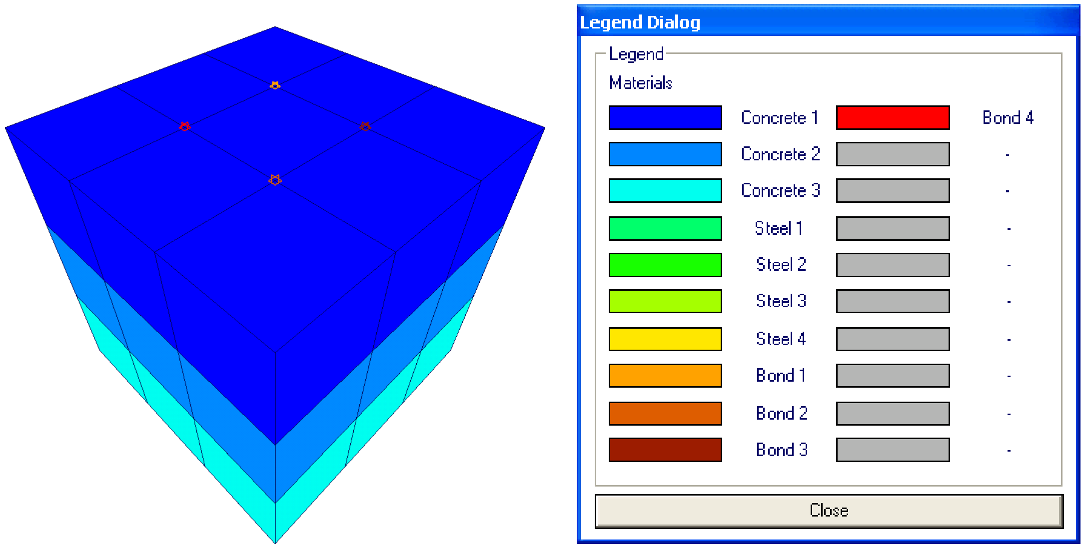


Figure 3.33: Simple VecTor3 Model in Material Mode and Legend Dialog

3.7.6 VecTor4 Gauss Points Mode

For VecTor4 models, Janus allows users to view the Gauss Point arrangement of all shell elements by selecting **Structure** ▷ *Gauss Points (V4)*. Within each shell element, each Gauss Point is displayed in a unique colour so that it may be differentiated from its neighbouring points as well as the Gauss Points of adjacent shell elements. Since Gauss Points occur at mid-depth within the shell element, the VecTor4 model is rendered as a simplified wireframe - this allows the user to see the precise position of each Gauss Point with respect to the external boundaries of the containing shell element. The Gauss Points mode may be toggled on and off using the *Toggle Face Fill* button. Currently, the Gauss Points mode is only accessible from the Global Model View.

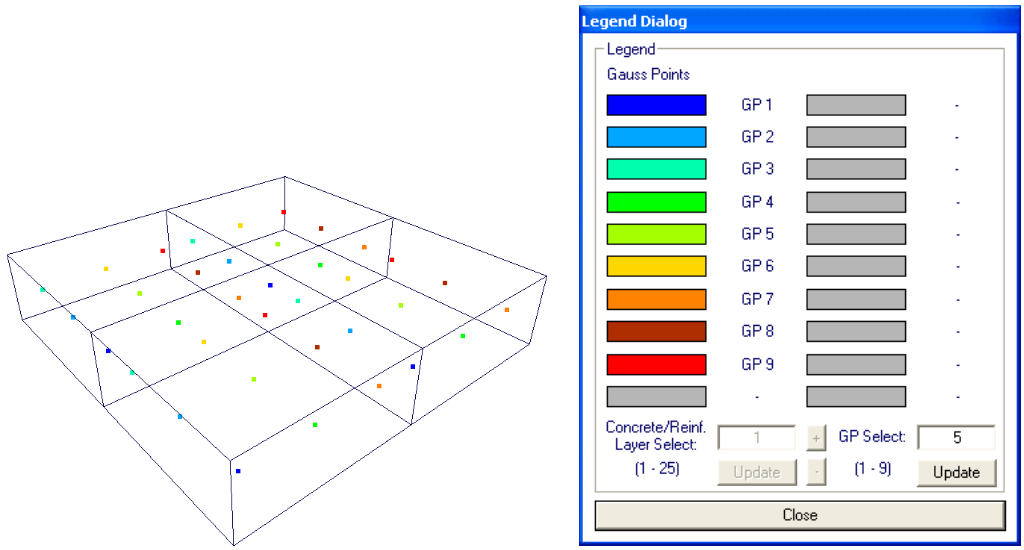


Figure 3.34: Simple VecTor4 Model in Gauss Points Mode

3.8 Analysis Results

One of the main primary functions of Janus is to communicate the analysis results of VecTor models to users in an intuitive yet comprehensive manner. The **Results** menu (shown in Figure 3.35) as well as the *Hotspot* button (accessed from the Janus Toolbar) provide a variety of mechanisms for presenting structural response characteristics as determined by VecTor software. Where applicable, analysis result options in Janus may also be utilized in tandem with alternate model view options like Section View and/or Layer View, providing additional versatility for the user. The following section discusses the various result-related display options and features of Janus.

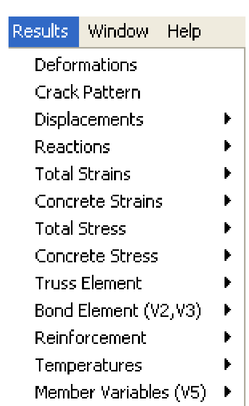


Figure 3.35: Results Menu

3.8.1 Deformations and Rotations

Nodal displacements (and in the case of VecTor4, rotations) are universally included as part of the output for a successfully converged load stage solution. In addition to providing the actual visualization of the structure's movement due to prescribed loading conditions, the presentation of global structure deformations and rotations may serve as a useful preliminary gauge of overall model behaviour. Viewing the elementary global motion of a loaded structural model may even help users to pinpoint areas requiring further detailed analysis, as well as indicate particular regions or elements under structural distress. For this reason, Janus is capable of displaying structural deformations for all VecTor model types as an integral part of its analysis result display functionality. Deformations mode may be enabled or disabled by selecting **Results** ▷ *Deformations*.

Upon activation of Deformations mode, the active structural model is re-drawn with all nodal coordinates translated by the orthogonal dx, dy, and dz displacement values given for the current load stage. Since all nodal displacements are provided in terms of VecTor model units, each value must be transformed and scaled to an equivalent quantity in OpenGL model space. In order to perform the necessary displacement value conversion, data reading functions in Janus utilize the same nodal coordinate transformation and scaling procedure previously described in Section 3.2. Each transformed displacement value is also multiplied by a uniform deformation scale factor so that they may be easily seen by the user. By default, the deformation scale factor is set to 30. See Figure 3.36 for an illustration of how nodal displacement values are linearly scaled based on the deformation scale factor, denoted as D.

As previously discussed in Section 3.6, the deformation scale factor may be customized by selecting

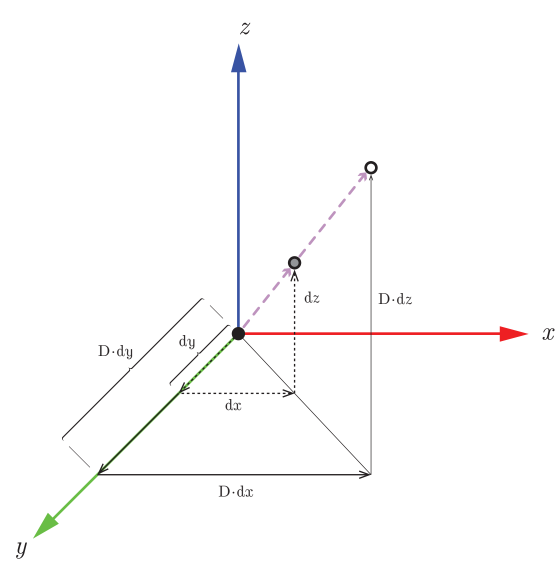


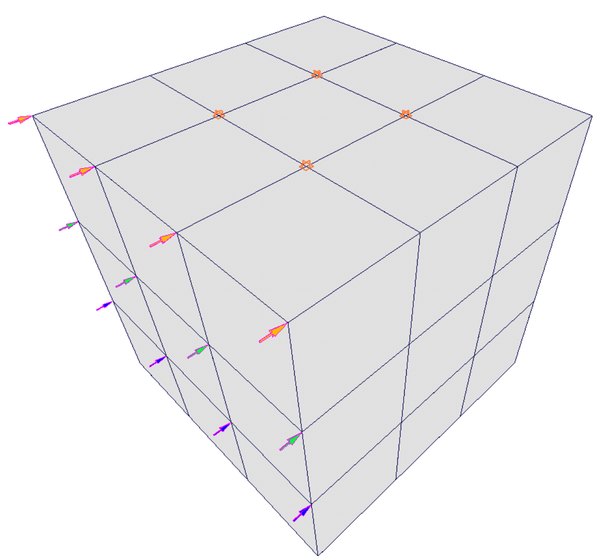
Figure 3.36: Scaled Nodal Deformations

View ▷ *Set Deform. Scale* ▷ *Select Scale Factor*. Where applicable, Deformations mode may be combined with Section View and a variety of other contour modes. An illustrative example of Deformations mode is provided in Figure 3.37.

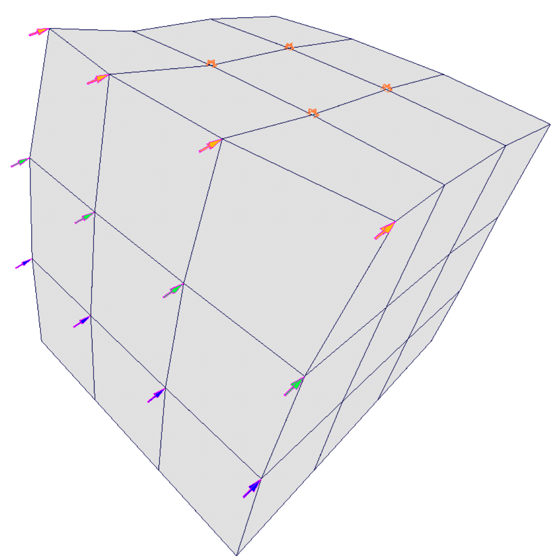
3.8.1.1 VecTor4 Nodal Rotations

In addition to determining displacements, VecTor4 also calculates the local rotation angles at each node for a given load increment. VecTor4 shell element vertices are defined on a centreline basis in the expanded structure file, with each pair of element vertices located a distance of (*element thickness/2*) perpendicularly "above" and "below" the node in the local v_3 direction. In order to provide a consistent rendition of the analytically predicted shell/plate deformation behaviour in VecTor4, Janus is able to depict shell element vertices rotated about their respective mid-depth node locations. The orientation of in-plane rotational deformation, θ_1 and θ_2, are given with respect to the local v_2 and v_1 nodal coordinate system vectors at each node (see Figure 2.2 b)).

For reference, the following procedure of converting nodal rotations into scaled displacements is illustrated in Figure 3.38 below, depicting a simplified shell element top vertex undergoing a general counterclockwise rotation θ about the mid-depth node k. Using the element thickness and the angle of rotation, Janus decomposes the rotational displacement of the vertex into series of linear x-, y-, and z-components using small-angle approximation and standard trigonometric relationships. The orthogonal



a) Deformations mode disabled



b) Deformations mode enabled

Figure 3.37: Simple VecTor3 Model in Deformations Mode

deformations due to rotations are denoted as $d_{\theta x}$, $d_{\theta y}$ and $d_{\theta z}$. Once obtained, the resulting displacement components are also magnified by the same displacement scale factor (D) applied to the linear nodal displacements.

This procedure is executed for both local in-plane rotations about the node, θ_1 and θ_2. Lastly, the scaled displacement components from nodal rotations θ_1 and θ_2 are combined into a single orthogonal displacement value. These combined displacement values are subsequently used to modify top and bottom vertex coordinate positions when presenting the deformed VecTor4 shell element model. It is important to note that the true shell element thicknesses at rotated VecTor4 element edges are not conserved due to the displacement components being linearly scaled instead of rotationally magnified. However, based on the following points of rationalization, this aspect of imprecision is deemed acceptable for general post-processing purposes:

- As initially outlined in the beginning of the section, the intended utility of Deformations mode is to present the overall global motion of the structure under specified loading conditions. Approximating shell element edge rotations in a consistent fashion and scale with the linear nodal displacements is adequate for conveying the general structural behaviour of the VecTor4 model.
- For most practical analysis applications, nodal displacements and rotations are typically several orders of magnitude smaller than the structure itself. The distortion of element dimensions from magnifying displacements in a linear versus radial fashion is decidedly negligible.

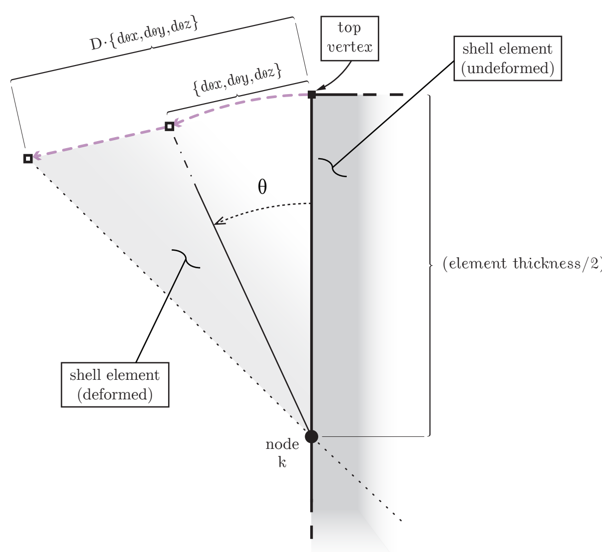


Figure 3.38: Scaled Nodal Rotations in VecTor4 Models

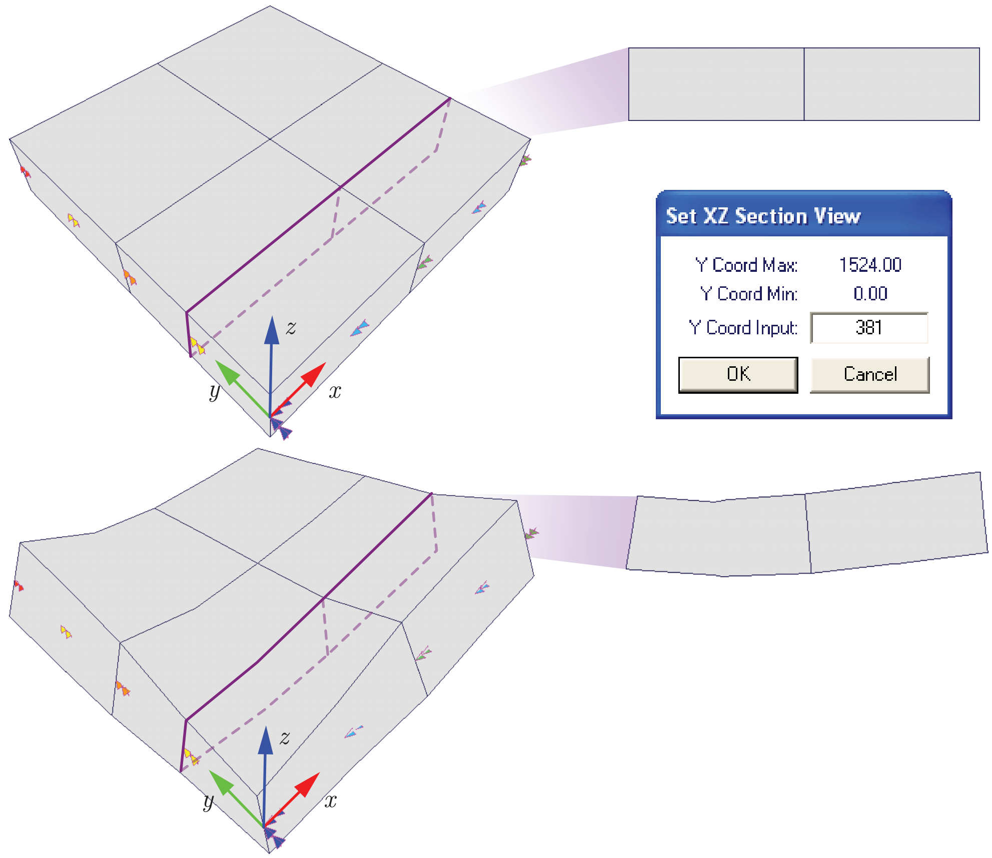


Figure 3.39: Simple VecTor4 Model with Nodal Rotations

3.8.2 Crack Pattern

Janus is capable of presenting crack pattern data of RC elements in VecTor2, VecTor3, and VecTor6 programs, accessed through **Results** ▹ *Crack Pattern*. All RC element cracks in Janus are represented as pairs of coloured line segments extending outward from the centroid of the cracked element face(s) in opposite directions. The crack line is drawn parallel to the plane of each element face. For viewing model crack patterns from the 2D analysis programs VecTor2 and VecTor6, this logically corresponds to a single crack line being drawn for each crack per element. In order to accommodate for freely manipulated 3D viewing of VecTor3 models in Janus, crack lines must be projected upon each face of a cracked RC solid element.

Since crack line establishment relies solely on the vertex coordinates and centroid of each element face, the Crack Pattern mode may be enabled at the same time as Deformations mode. However, it is important to note that the crack line orientation is solely established based on the global model axes; the presented angles of crack lines on deformed element faces do not compensate for any apparent rotations due to scaled nodal displacements. In terms of overall analysis capabilities, up to two cracks may be displayed for VecTor2 and VecTor6 models, and up to three cracks for VecTor3 models. For each element, cracks are differentiated using a distinctly coloured line - the cracks in direction 1 are denoted using red lines, direction 2 cracks are represented as green, and direction 3 cracks are coloured in blue. See Figure 3.40 for a conceptual demonstration of crack patterns enabled with default crack view settings. Additionally, see Figure 3.41 for a conceptual depiction of how crack patterns are displayed for 2D (VecTor2 and VecTor6) and 3D (VecTor3) finite element models in Janus.

Aside from using colour to differentiate between multiple cracks on an element face, Janus utilizes three predominant visual parameters in order to present a qualitative overview RC element crack information to users - line width, line length, and angle of orientation. As previously discussed in Subsection 3.6.6, the menu option **View** ▷ *Set Crack View* may be used to customize the displayed width and length properties of the crack for ease of viewing. Currently, there are no applicable options available for modifying how crack angles are presented in Janus. The following subsections present the methodology in which each crack line property is determined and visualized by Janus.

3.8.2.1 Line Angle

Regardless of the VecTor program, all RC element cracks in Janus are positioned on the centroid of the element face. From this point, two equal-length and opposite-direction line segments extend outward in the corresponding direction of the crack, parallel to the plane of the face. Together, the two line segments represent the crack line projected upon the element face. The display of crack orientation for 2D VecTor programs (VecTor2 and VecTor6) is straightforward, with the crack angle value directly

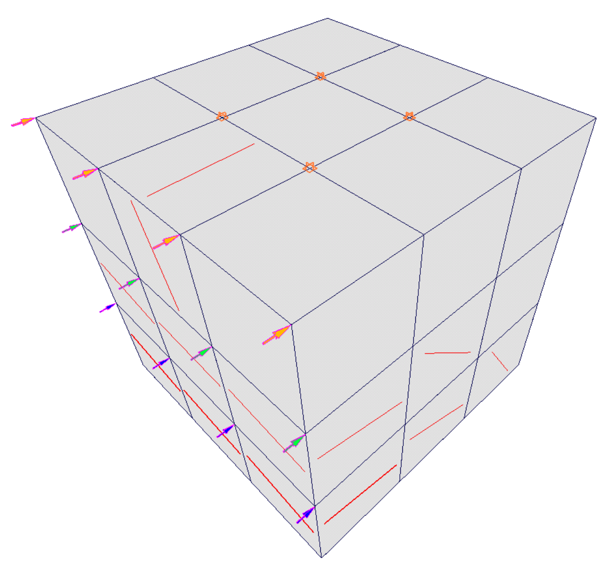
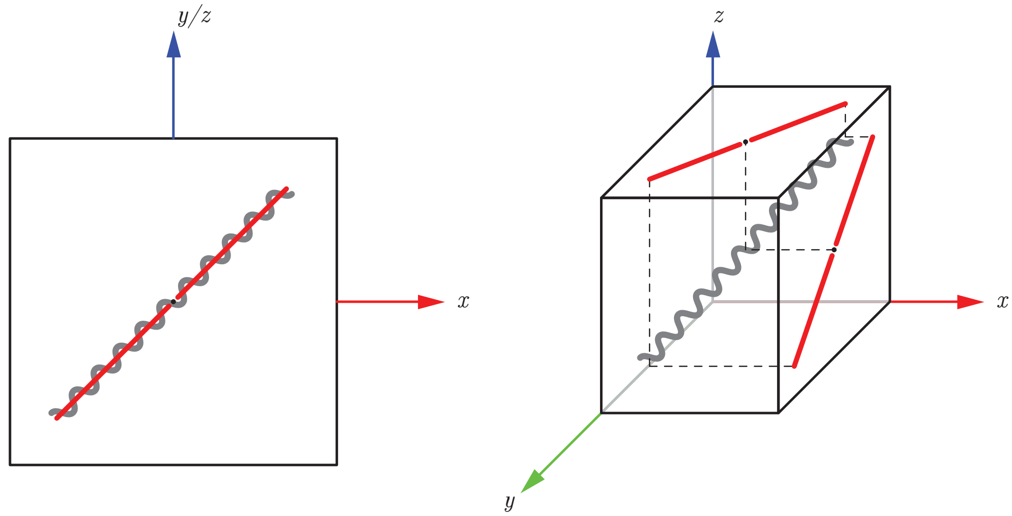


Figure 3.40: Simple VecTor3 Model in Crack Pattern Mode



a) Crack lines in VecTor2 & VecTor6 Models a) Crack lines in VecTor3 Models

Figure 3.41: 2D and 3D Crack Patterns in Janus

corresponding to the orientation of the line segments extending from the element centroid. With 3D elements in VecTor3, the relative orientation of each set of line segments per element face is determined as the projection of the crack angle vector upon the geometric plane of the face.

3.8.2.2 Line Width

Depending on the selected crack view settings, the crack width parameter could influence the width and/or length of crack lines rendered in Janus. By default, Janus displays the rendered OpenGL crack line widths based on specified ranges of crack width values. As shown in Table 3.5, the rendered line thickness is specified to widen as the crack width increases in severity. For an upper limit of crack widths exceeding 3.0 mm, the rendered line width is limited to a maximum of 3 on-screen line width units - in doing so, overall model visibility can be maintained while still indicating that a significant crack is present at a particular element. All crack line widths may also be scaled up or down using a uniform *Crack Width Scale* factor, modified using the *Set Crack View* dialog.

3.8.2.3 Line Length

By default, Janus displays crack lengths as a function of the crack angle vector projected on the planar element face and the inner "radius" of the face (i.e. the maximum distance that the crack line may extend from the centroid of the face without intersecting an edge). Essentially, the crack lines are bounded by the perimeter of a circle inscribed within the element face. As previously discussed in Subsection 3.6.6, there

Crack Width, W_{cr} (mm)	**OpenGL Rendered Line Width**
$0 < W_{cr} \leq 0.25$	1.0
$0.25 < W_{cr} \leq 0.5$	1.5
$0.5 < W_{cr} \leq 1.5$	1.9
$1.5 < W_{cr} \leq 3.0$	2.3
$W_{cr} > 3.0$	3.0

Table 3.5: Default Crack Width Ranges and Corresponding Displayed Line Thicknesses

may be certain analysis scenarios where it may be beneficial to use the *Alternative Crack View* setting to abstract the lengths of all rendered crack lines as a function of the crack width parameter multiplied by a specified *Crack Length Scale* factor. It is important to note that while this alternative crack view setting is active, cracks with a crack width parameter of approximately 0 will not be displayed - predictably, the effective line length will be computed as 0 regardless of *Crack Length Scale* factor choice. Conversely, due to the fact that crack lengths are determined solely on a scalar crack width value, extreme crack width values could result in displayed crack lengths unrealistically extending beyond the perimeter of the element face.

3.8.3 Contour Mode

Upon selection of a user-specified stress and strain-related value from the **Results** menu, the VecTor model enters the contour mode. Within the contour mode, elements or nodes are re-coloured in a colour contour pattern corresponding to a range of numerical values presented in an external, general-purpose *Legend* dialog. Element types that do not apply will be coloured black. The *Legend* dialog uses a 20 colour array to establish a spectrum of intermediate values between the local minimum and maximum for the selected result parameter and load stage. As the load stage is changed, the *Legend* dialog values and on-screen colours will accordingly update for the currently selected contour mode parameter. The contour mode is compatible for use in combination with both Deformations and Crack Pattern modes, and, where applicable, may be implemented within the Section View and/or Layer View as well. Nodal features such as nodal loads and restraints may also be toggled on and off in conjunction with the contour mode. See Figure 3.42 for an example of the relationship between the contour mode and *Legend* dialog. Throughout this section, nodal and elemental variables will be listed in conjunction with the respective Janus menu notation provided in brackets.

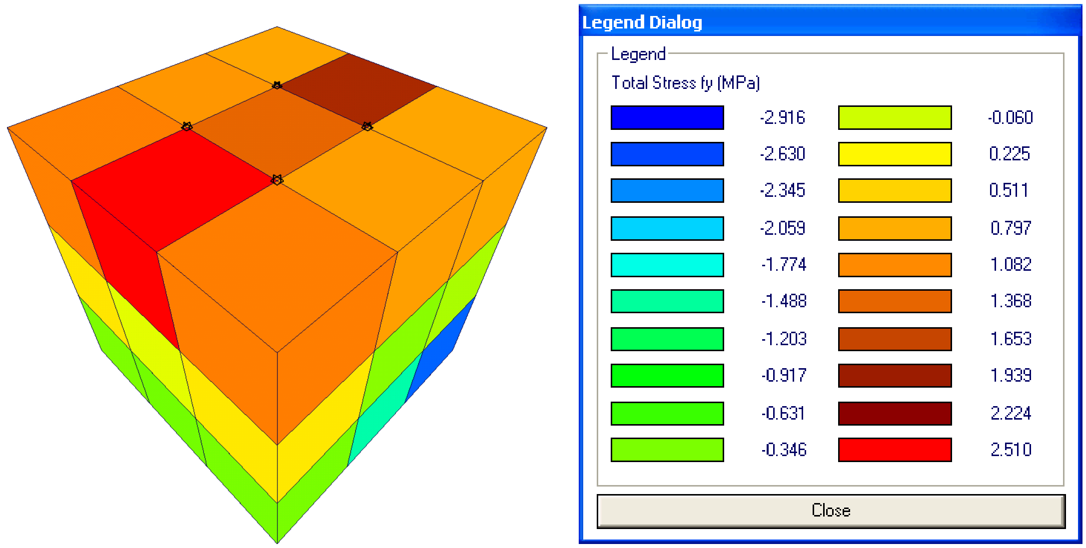


Figure 3.42: Simple VecTor3 Model in RC Element Contour Mode and Contour Dialog

3.8.3.1 Nodal Displacements and Rotations

The contour mode for nodal displacements and/or rotations allows for users to select and view individual deformation variables as a smooth colour gradient. In contrast to Deformations mode, where all nodal displacement values are displayed simultaneously as actual model deformations, users may use the contour mode to visualize and compare nodal displacement variables in isolation from each other. Colours are assigned to each model node based on the selected displacement variable, and the resulting colours are displayed on the model solid element faces. For visual continuity of colours between nodal points, linear colour transitions occur between all nodes connected by a common element face. See Figure 3.43 for an example of nodal displacement contour mode demonstrated using a simple VecTor3 model. Displacement and rotation variables are selectively enabled and disabled from the **Results** menu based on the prescribed degrees of freedom available for the opened VecTor model type. See Table 3.6 for an overview of available node-related contour mode options available per VecTor program. Contour mode for nodal displacement and rotation variables are not available while Section View or Layer View are active.

3.8.3.2 Nodal Reaction Forces and Moments

In an analogous manner to displaying nodal displacements and rotations, users may also choose to enable relevant nodal reactions and/or moment variable contour modes. Reactions and/or moment values occur at restrained node(s) that have resisted some portion of the load applied to the structure. The contour mode assigns colours to each restrained node based on the minimum and maximum value

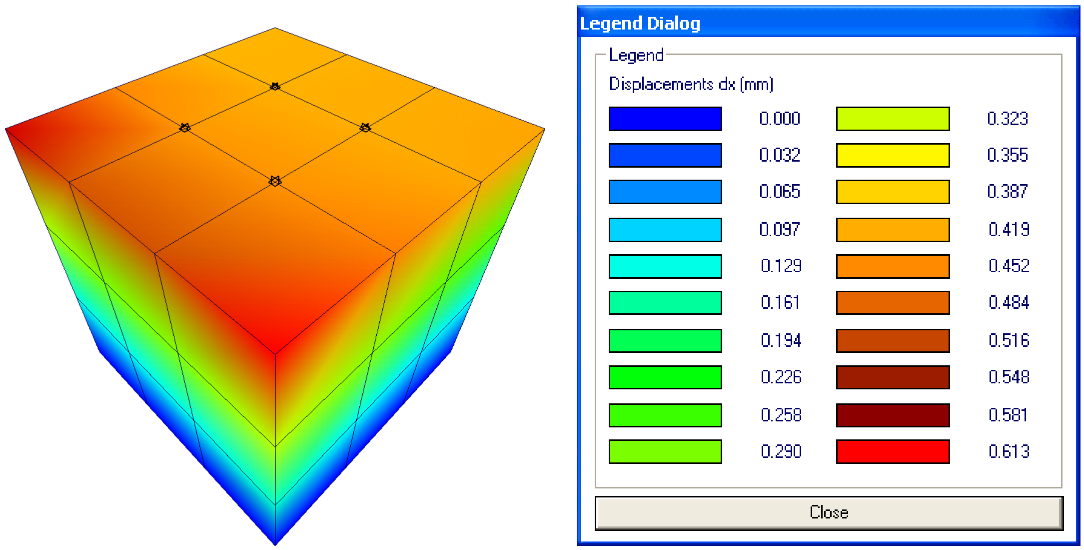


Figure 3.43: Simple VecTor3 Example in Nodal Contour Mode and Contour Dialog

for the requested variable and current load stage, with smooth linear colour gradient transitions between adjacent nodes sharing a common element face. Using the same colour distribution, all non-restrained nodes values are also assigned a colour value corresponding to 0. Like the contour mode for nodal displacements and rotations, the contour mode options for reaction and moment variables are organized by orthogonal direction and axis of rotation. Within the **Results** menu, options are accordingly activated or disabled based on the available degrees of freedoms of the VecTor model. Refer to Table 3.6 for an overview of available nodal reaction and moment contour mode options per VecTor program. Contour mode for nodal reaction and moment variables are not available while Section View or Layer View are active.

Program	**Displacements**			**Rotations**			**Reactions**			**Moments**		
	δ_x (dx)	δ_y (dy)	δ_z (dz)	θ_1 (r1)	θ_2 (r2)	θ_z (rz)	R_x (Rx)	R_y (Ry)	R_z (Rz)	M_1 (M1)	M_2 (M2)	M_z (Mz)
VecTor2	✓	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗
VecTor3	✓	✓	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗
VecTor4	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
VecTor5	✓	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗	✓
VecTor6	✓	✗	✓	✗	✗	✗	✓	✗	✓	✗	✗	✗

Table 3.6: Available Node-Related Contour Modes per VecTor Program

3.8.3.3 RC Element Variables

Janus displays stress and strain-related result values for RC elements by re-colouring element faces as solid colours corresponding to the array of colours and numerical values presented in the *Legend* dialog. Available variables vary by VecTor program and available model dimensionality, but generally include:

- Total axial and shear strain values
- Total principal strain values
- Total axial and shear stress values
- Total principal stress values
- Concrete net axial and shear strain values
- Concrete net principal strain values
- Concrete net axial and shear stress values
- Concrete net principal stress values

In addition to being compatible with Deformations and Crack Pattern modes, element-based contour mode parameters such as RC element stresses and strains may be used in conjunction with available Section View and/or Layer View options. See Figure 3.44 for an example of total stress contour mode viewed at an intermediate element section coordinate using *YZ Section* view. An overview of available RC element stress and strain variables per VecTor program is provided in Table 3.7 below.

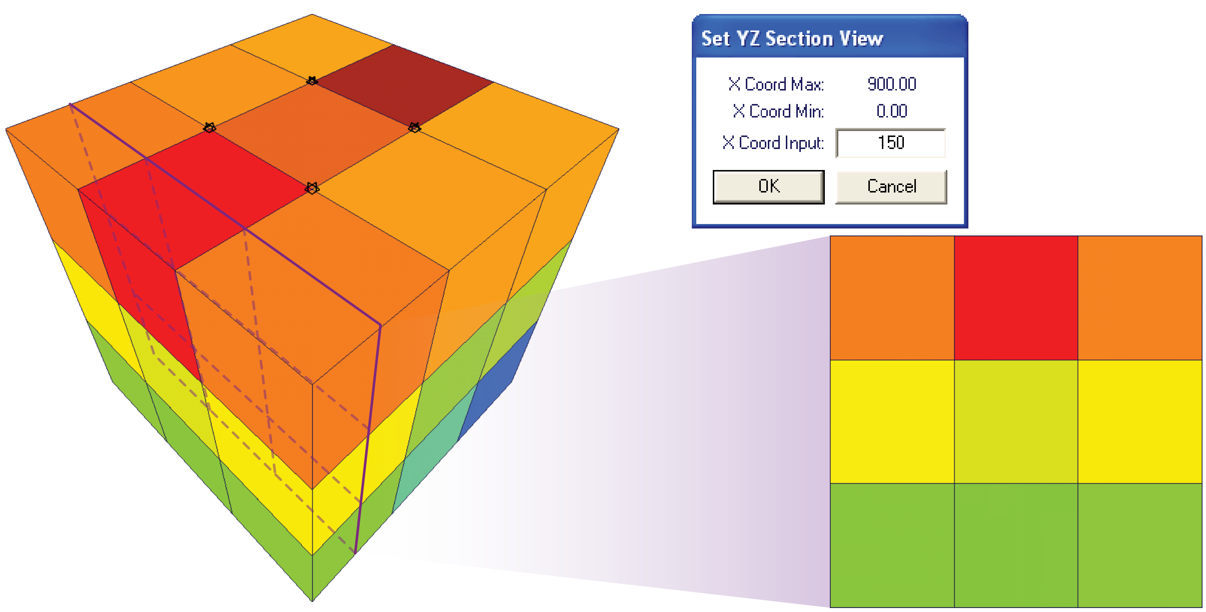


Figure 3.44: Simple VecTor3 Model in Contour Mode and Section View

Program		VecTor2	VecTor3	VecTor4	VecTor5	VecTor6
Total Strains	ε_x (ex)	✓	✓	✓	✗	✓
	ε_y (ey)	✓	✓	✓	✗	✓
	ε_z (ez)	✗	✓	✓	✗	✓
	γ_{xy} (exy)	✓	✓	✓	✗	✓
	γ_{yz} (eyz)	✗	✓	✓	✗	✓
	γ_{xz} (exz)	✗	✓	✓	✗	✓
	ε_1 (e1)	✓	✓	✓	✓	✓
	ε_2 (e2)	✓	✓	✓	✓	✓
	ε_3 (e3)	✗	✓	✓	✗	✓
Concrete Net Strains	ε_{cx} (ecx)	✓	✓	✓	✓	✓
	ε_{cy} (ecy)	✓	✓	✓	✓	✓
	ε_{cz} (ecz)	✗	✓	✓	✗	✓
	τ_{cxy} (ecxy)	✓	✓	✓	✓	✓
	τ_{cyz} (ecyz)	✗	✓	✓	✗	✓
	τ_{cxz} (ecxz)	✗	✓	✓	✗	✓
	ε_{c1} (ec1)	✓	✓	✓	✓	✓
	ε_{c2} (ec2)	✓	✓	✓	✓	✓
	ε_{c3} (ec3)	✗	✓	✓	✗	✓
Total Stresses	σ_x (fx)	✓	✓	✓	✗	✓
	σ_y (fy)	✓	✓	✓	✗	✓
	σ_z (fz)	✗	✓	✓	✗	✓
	τ_{xy} (vxy)	✓	✓	✓	✗	✓
	τ_{yz} (vyz)	✗	✓	✓	✗	✓
	τ_{xz} (vxz)	✗	✓	✓	✗	✓
	σ_1 (f1)	✓	✓	✓	✗	✓
	σ_2 (f2)	✓	✓	✓	✗	✓
	σ_3 (f3)	✗	✓	✓	✗	✓
Concrete Net Stresses	σ_{cx} (fcx)	✓	✓	✓	✓	✓
	σ_{cy} (fcy)	✓	✓	✓	✓	✓
	σ_{cz} (fcz)	✗	✓	✓	✗	✓
	γ_{cxy} (vcxy)	✓	✓	✓	✓	✓
	γ_{cyz} (vcyz)	✗	✓	✓	✗	✓
	γ_{cxz} (vcxz)	✗	✓	✓	✗	✓
	σ_{c1} (fc1)	✓	✓	✓	✓	✓
	σ_{c2} (fc2)	✓	✓	✓	✓	✓
	σ_{c3} (fc3)	✗	✓	✓	✗	✓

Table 3.7: Available RC Element Stress and Strain Contour Modes per VecTor Program

3.8.3.4 VecTor-Specific Element Variables

A variety of additional RC element attributes specific to certain VecTor programs may also be viewed via contour mode, such as element temperatures and unique member element parameters for VecTor5 frame analyses. Refer to subsequent program sections for VecTor-specific discussion of variables such as element temperatures as well as member end force and member deformation variable contour modes.

3.8.3.5 Truss Element Variables

Truss element stress and strain values may also be viewed in Janus as an extension of the colour contour mode functionality. Due to the fact that discrete steel elements are linear elements that are positioned adjacent to or between solid RC elements, providing visibility of these elements is a high priority while their respective contour modes are active. For this reason, activation of any truss-related stress and strain contour mode switches the appearance of solid elements in the VecTor model to a wireframe mode. As shown in Figure 3.45, truss elements are highlighted in thick line widths, coloured using the contour mode colour spectrum. Accordingly, the *Legend* dialog displays the selected truss element parameter values and colour array. Truss element contour mode is capable of displaying the following discrete steel element parameters:

- Average truss strain, ε_s (es)
- Average truss stress, σ_s (fs)
- Truss strain at crack, ε_{scr} (esc)
- Truss stress at crack, σ_{scr} (fsc)

Although present discussion of stress and strain contour mode parameters is designated for “truss elements”, it should be noted that VecTor6 ring bar elements are also included as part of the same contour mode. This is due to the fact that ring bar elements are fundamentally designed to represent out-of-plane discrete reinforcement in axisymmetric structural elements.

3.8.3.6 Bond Element Variables

Bond element response properties may be viewed in contour mode via the menu option **Results** ▹ *Bond Element (V2, V3)*. In a similar graphic manner to truss element contour mode, bond elements in VecTor2 and VecTor3 are highlighted using a colour pattern to represent bond slip characteristics for each element and current load stage in the *Legend* dialog. See Figure 3.46 below for a depiction of bond

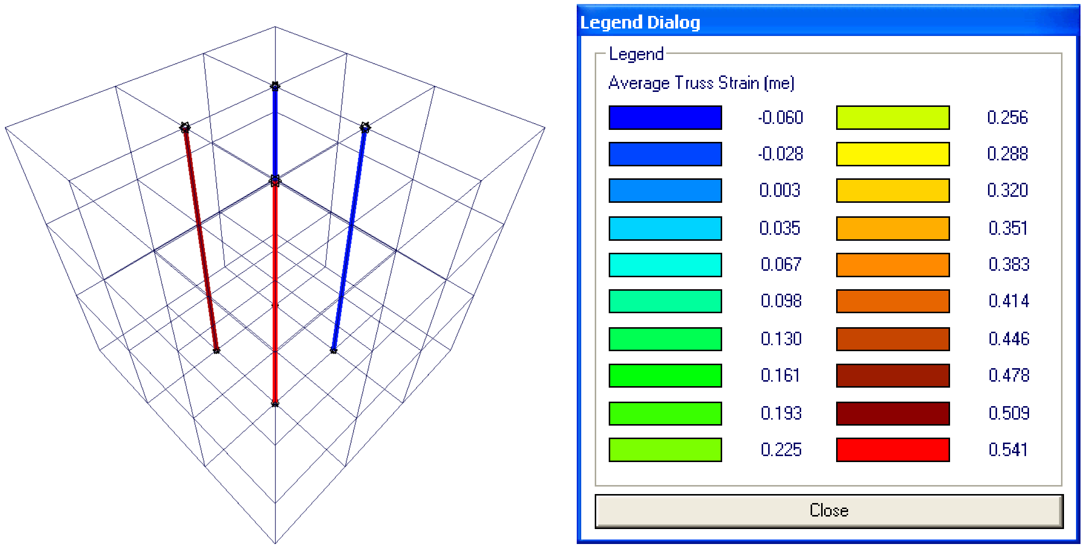


Figure 3.45: Simple VecTor3 Model in Truss Element Contour Mode and Contour Dialog

Program	VecTor2	VecTor3
Slip Factor (S/Sm)	✓	✓
Δ_1 (Slip1)	✓	✓
Δ_2 (Slip2)[2]	✓	✗
τ (FxAvg)	✓	✓

[2] Contact elements only

Table 3.8: Available Bond Element Contour Modes per VecTor program

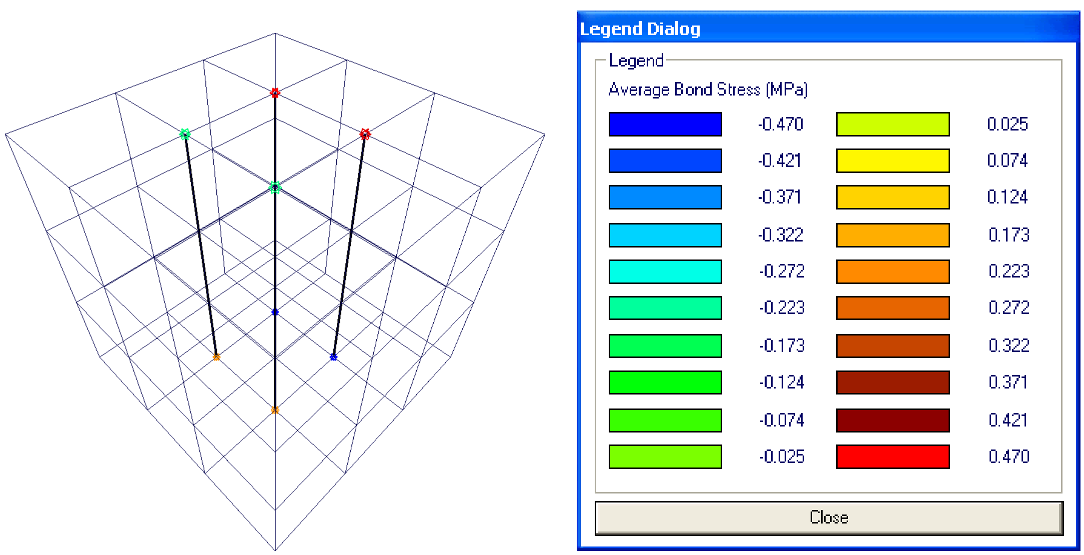


Figure 3.46: Simple VecTor3 Model in Bond Element Contour Mode and Contour Dialog

element contour mode using a simple VecTor3 model. A list of bond element parameters available in contour mode is provided in Table 3.8 below.

3.8.3.7 Smeared Reinforcement

As an alternative to implementing discrete reinforcement elements, VecTor programs feature the ability to represent reinforcement as a smeared material property within RC elements. Based on the prescribed material properties, reinforcement orientation, and the individual dimensions of each assigned RC element, each instance of smeared reinforcement is capable of providing supplemental structural strength and resistance properties analogous to an equivalent quantity of discrete reinforcement positioned within the element. In a likewise manner to inspecting the performance of discrete reinforcement objects, the stress and strain characteristics of each smeared reinforcement type may be of interest to the user. Hence, contour mode options are provided for viewing the local stress-strain state of each smeared reinforcement type, consisting of the following options:

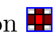
- Average smeared reinforcement strain, ε_s (es)
- Average smeared reinforcement stress, σ_s (fs)
- Smeared reinforcement strain at crack, ε_{scr} (esc)
- Smeared reinforcement stress at crack, σ_{scr} (fsc)

Since smeared reinforcement is treated as an internal property within RC elements, contour modes for smeared reinforcement stress and strain values are displayed on an element face basis. Like the contour mode for elemental stresses and strains, elements corresponding to the requested smeared reinforcement type will exhibit colours associated with the range of values presented in the *Legend* dialog. In the case that an element does not possess the applicable smeared reinforcement property requested by the user, it will be coloured black for visual clarity.

The smeared reinforcement contour mode options may be accessed through **Results** ▷ *Reinforcement*. For VecTor2, VecTor3, and VecTor6, a maximum of four smeared reinforcement types are permitted per RC material type. For consistency with the format of smeared reinforcement listings in the expanded structure file, smeared reinforcement types are sequentially identified as Direction 1 through 4. Correspondingly named menu options are provided in Janus for viewing each type, and options are enabled and disabled based on the maximum number of smeared reinforcement types found. Available contour mode menu options may also vary based on the VecTor model type being viewed; due to unique model and dimension characteristics, Janus facilitates the display of VecTor4 and VecTor5 smeared reinforcement contour modes in an alternative manner. Refer to discussion in the VecTor4 and VecTor5 unique feature sections (Sections 6.2 and 7.2, respectively) for further details.

3.8.4 Hotspot Mode

Although the contour mode is useful for visually representing the global distribution of structural parameter values across the entire VecTor model, users may wish to investigate which nodes or elements precisely fall within or exceed a subset of specified values. The Hotspot mode provides this functionality, allowing users to view and isolate a material parameter based on an upper and lower bound of threshold values. The Hotspot mode is activated by completing the following steps:

1. Invoke the *Hotspot* dialog (See Figure 3.47) by clicking the *Hotspot* toolbar button 

2. Select the desired node or element-specific parameter from the list of options.
   Note: certain variables may be enabled or disabled based on the VecTor program, applicable material type(s) and/or available element specifications.

3. Observing the current minimum and maximum value for the selected parameter and load stage, enter applicable lower and upper bound values into the appropriate edit boxes within the *Hotspot Range* area.

4. Press OK.

Once the Hotspot mode is activated, the VecTor model is displayed as a wireframe. Depending on whether the selected structural parameter or variable is node-related or pertains to a certain type of element (RC, truss, or bond), the corresponding type of model component is highlighted based to the chosen Hotspot upper and lower bound values. Applicable elements/nodes that exceed the specified upper bound are drawn in red, and elements/nodes that lie within the lower and upper bound are highlighted in brown. Elements or nodes that satisfy neither Hotspot mode condition are disregarded.

The resulting visual effect of red and brown highlighting in a wireframe allows users to precisely identify which elements or nodes in the VecTor model match or exceed the specified range of values for a given load stage. As an additional feature, the highlighted Hotspot elements/nodes will automatically update upon modification of the current load stage as well. Thus, users may traverse through the range of available load stage data, and incrementally observe the behaviour of the model with respect to the chosen variable and established numerical threshold. The Hotspot mode may be used in conjunction with Deformations mode as well as applicable Section View and/or Layer View options. Hotspot mode may be toggled on and off using the *Toggle Face Fill* toolbar button.

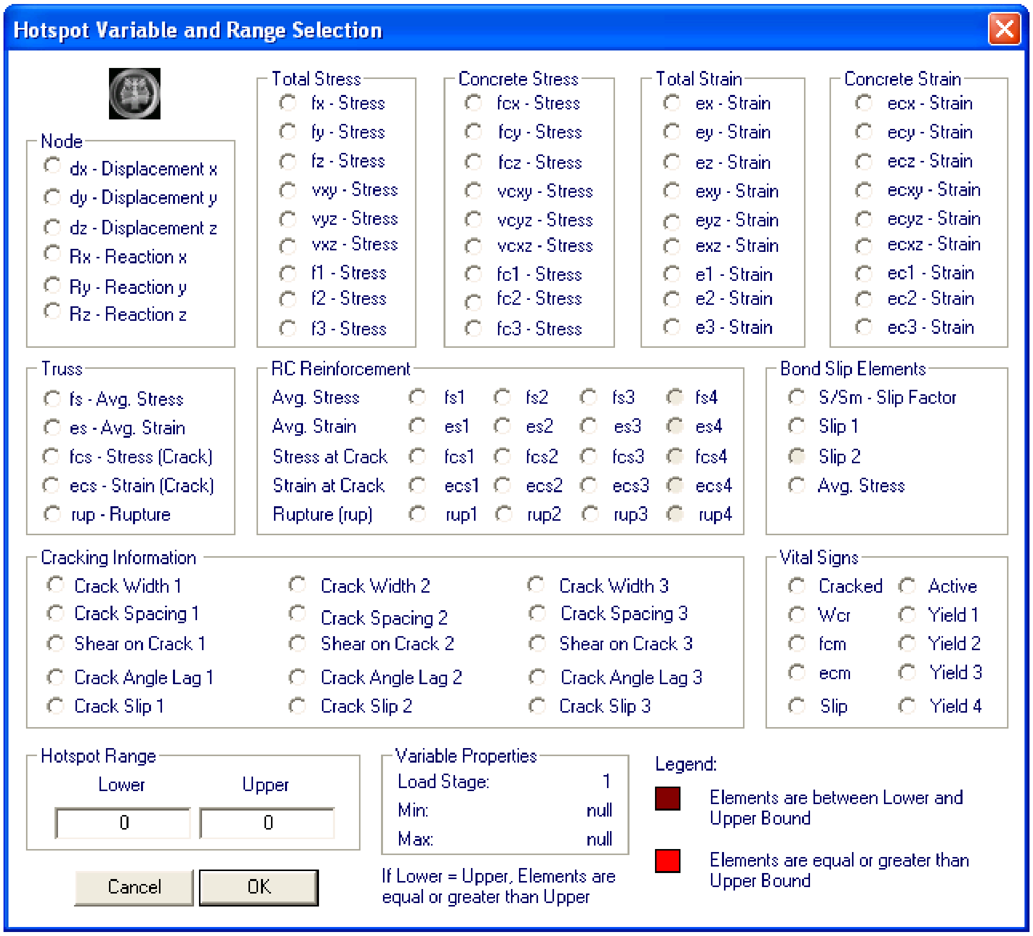
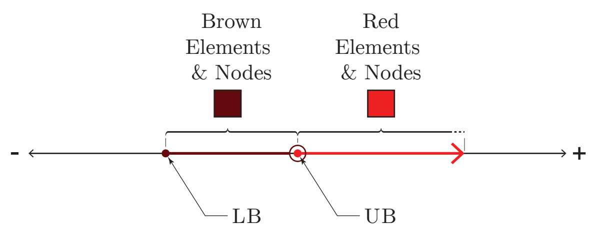


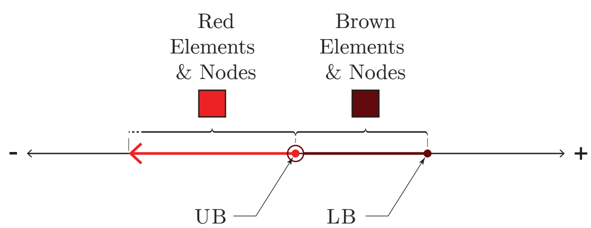
Figure 3.47: VecTor3 Hotspot Dialog

3.8.4.1 Upper and Lower Bound Selection

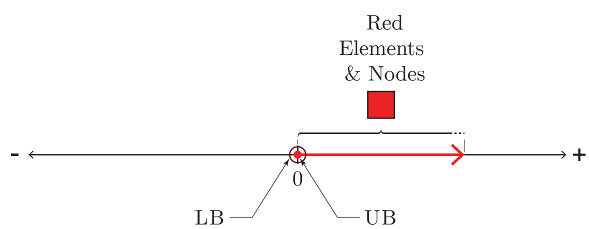
The upper and lower bound values work in conjunction to represent a set number range. The upper bound represents the numerical threshold of the selected node or element Hotspot parameter, and the lower bound value (relative to the upper bound value) establishes the "direction" that the parameter values are checked against to exceed compared to the upper bound value. The upper bound and lower bound edit boxes accept either positive and negative real numbers, and Janus automatically determines the intended direction of exceedance based on their relative positions on a number line. The number line relationships between the Hotspot upper bound and lower bound values are visually demonstrated in Figure 3.48 below. Elements or nodes with properties that equal or exceed the upper bound value are coloured red. Elements or nodes that fall between the upper and lower bound value ranges are coloured brown.



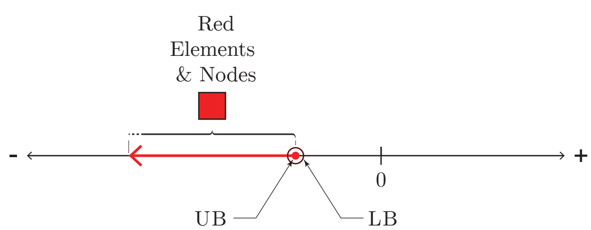
a) Upper bound (UB) > Lower bound (LB)



b) Upper bound (UB) < Lower bound (LB)



c) Upper bound (UB) = Lower bound (LB), UB ≥ 0



d) Upper bound (UB) = Lower bound (LB), UB < 0

Figure 3.48: Hotspot Lower and Upper Bound Functionality

3.8.4.2 Hotspot Mode in Global Model View

In the global model view, activation of the Hotspot mode reduces the VecTor model RC elements to a wireframe representation. Unless discrete reinforcement or bond-related element parameters are selected in the *Hotspot* dialog, finite elements such as truss, ringbar, contact and link elements will not appear as part of the simplified wireframe model. See Figure 3.49 for a simple VecTor3 model with Hotspot mode enabled for different types of elements.

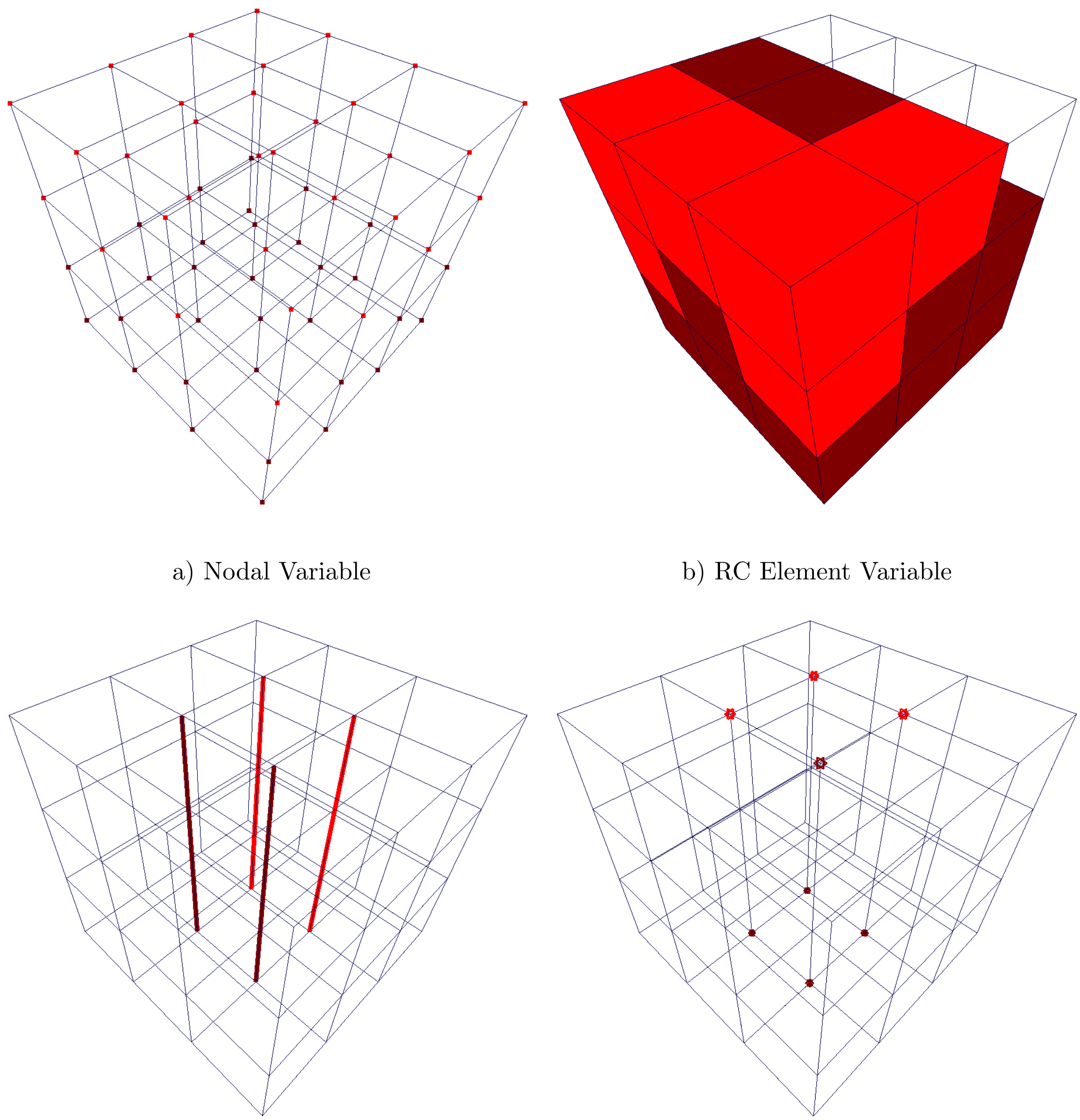
3.8.4.3 Hotspot Mode in Section/Layer View

In a similar capacity to the contour mode, applicable Section View and Layer View options may be utilized in conjunction with Hotspot mode. Section views may be particularly useful when viewing 3D VecTor models in Janus, where layers of highlighted solid elements have the potential to visually obscure any Hotspot-related highlighting of interior solid elements positioned behind the foremost element face. Consistent with standard Section View properties, Hotspot-highlighted linear elements such as truss elements will only appear in Section View if the element is intersected by or lies on the section plane. Similarly, Hotspot mode highlighting of point-based model features such as nodes and link elements will only appear when the nodal coordinate is intersected by the section plane. See Figure 3.50 below for sample section views of a simple VecTor3 model in Hotspot mode for different element types.

In contrast to using the Hotspot mode in Section View, which has equivalent functionality as Global Model View, Hotspot mode in Layer View provides specialized post-processing viewing capabilities to the user. Within Layer View of VecTor4 shell elements and VecTor5 output member elements, a layer-specific form of the Hotspot mode is activated. The Hotspot mode provides highlighting on a RC or discrete reinforcement layer basis. Specific descriptions of Hotspot modes in Layer View for VecTor4 and VecTor5 are subsequently provided in Section 6.2 and Section 7.2, respectively.

3.9 Data Platform

For general data manipulation and graphing purposes, the *Data Platform* dialog allows users to organize and present structural data for specified nodes or elements in a variety of formats. Numerical data may be output for up to five distinct structural variables for a set range of load stages, available in either raw text or excel file formats. Janus also provides an in-program preview function for convenient graphing of the selected variables as linear plots. Users may also assign axis variables and axes ranges as desired. Access to all Janus data platform functionality is provided through the *Data Platform* dialog, which is



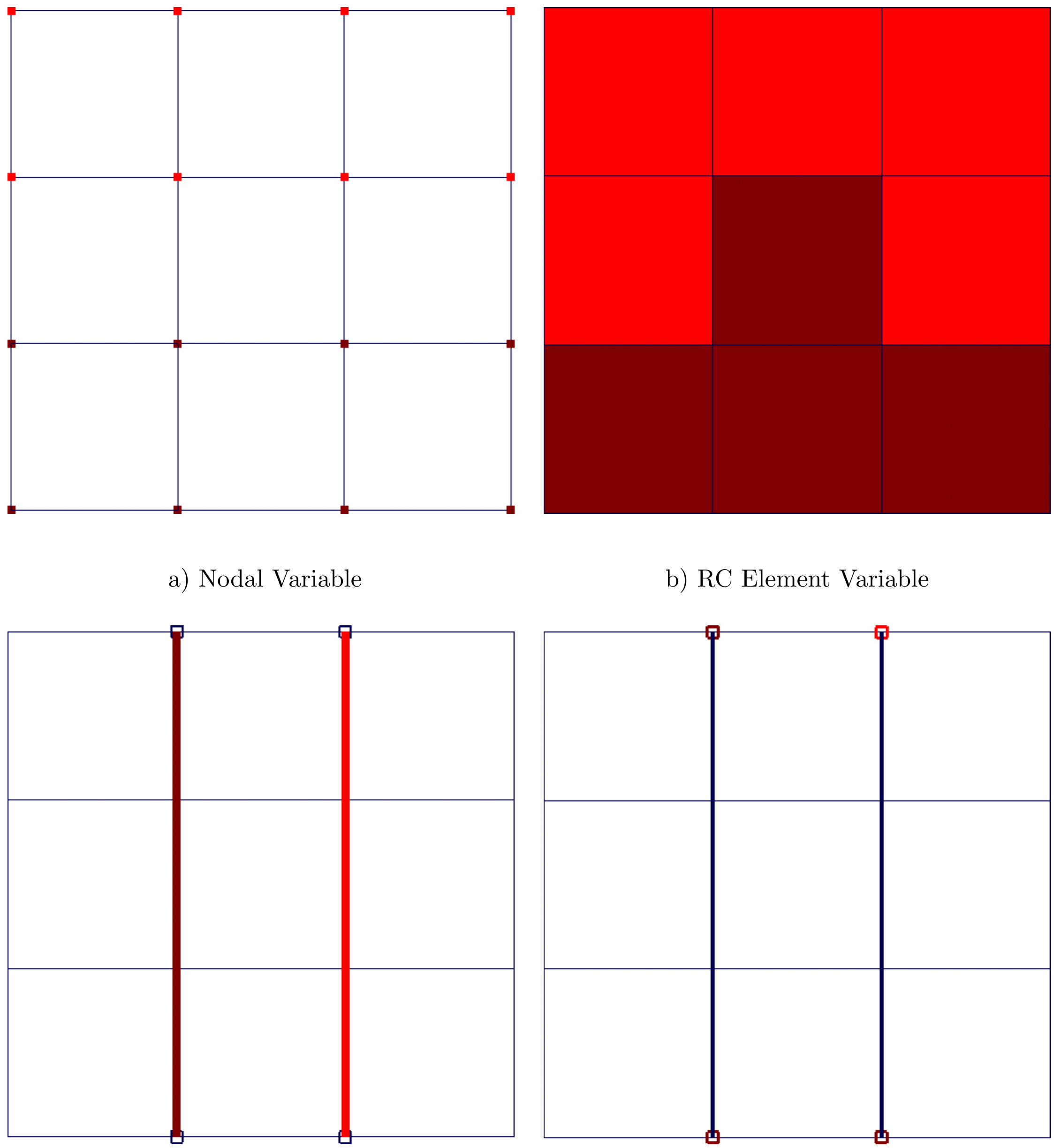
a) Nodal Variable

b) RC Element Variable

c) Truss Element Variable

d) Bond Element Variable

Figure 3.49: Various Hotspot Modes Using a Simple VecTor3 Model



a) Nodal Variable

b) RC Element Variable

c) Truss Element Variable

d) Bond Element Variable

Figure 3.50: Section Views of Various Hotspot Modes Using a Simple VecTor3 Model

opened by clicking the *Data Platform* toolbar button, . See Figure 3.51 for a typical representation of the *Data Platform* dialog.

3.9.1 Load Stage Selection

For general post-processing convenience, each instance of the *Data Platform* dialog initially specifies the entire load stage range loaded for the VecTor model by default. However, for viewing analyses with a significant number of load stages, or in the interest of extracting a certain subset of data, users may wish to customize the load stage range. The *From* and *To* edit boxes in the *Load Stages* area provide corresponding entries for users to specify custom starting and ending load stage values. Users must provide a valid range of load stage values for data acquisition purposes, satisfying the following conditions:

- The *From* and *To* load stage entries must be greater or equal to the starting load stage value, and less or equal to the final load stage value
- the *From* load stage entry value must be greater or equal to the *To* load stage entry value

Selecting the *Include Load Stages* option includes the listed load stages as an additional data column in the output data.

3.9.2 Variable Selection

The *Data Platform* dialog provides users with an interface consisting of five independent input columns. Each column is capable of accepting a unique data variable choice and the corresponding node, element or load case number of interest.

Each of the column buttons, titled *Variable1* through *Variable5*, invokes an instance of the general variable selection dialog displayed in Figure 3.52. Using this dialog, a different variable may be assigned to each the corresponding five node/element/load case entries below. Accordingly, a valid node/element/load case number must be assigned to each entry. Janus will retrieve data based on the context of the chosen parameter. Each entry column is individually activated by selecting the respective box corresponding to the“Check box to include data” label. For the purpose of customizing a graphic plot in Janus, one of the active variables may also be selected as the x-axis by selecting the corresponding box for “Check box to set X-axis”.

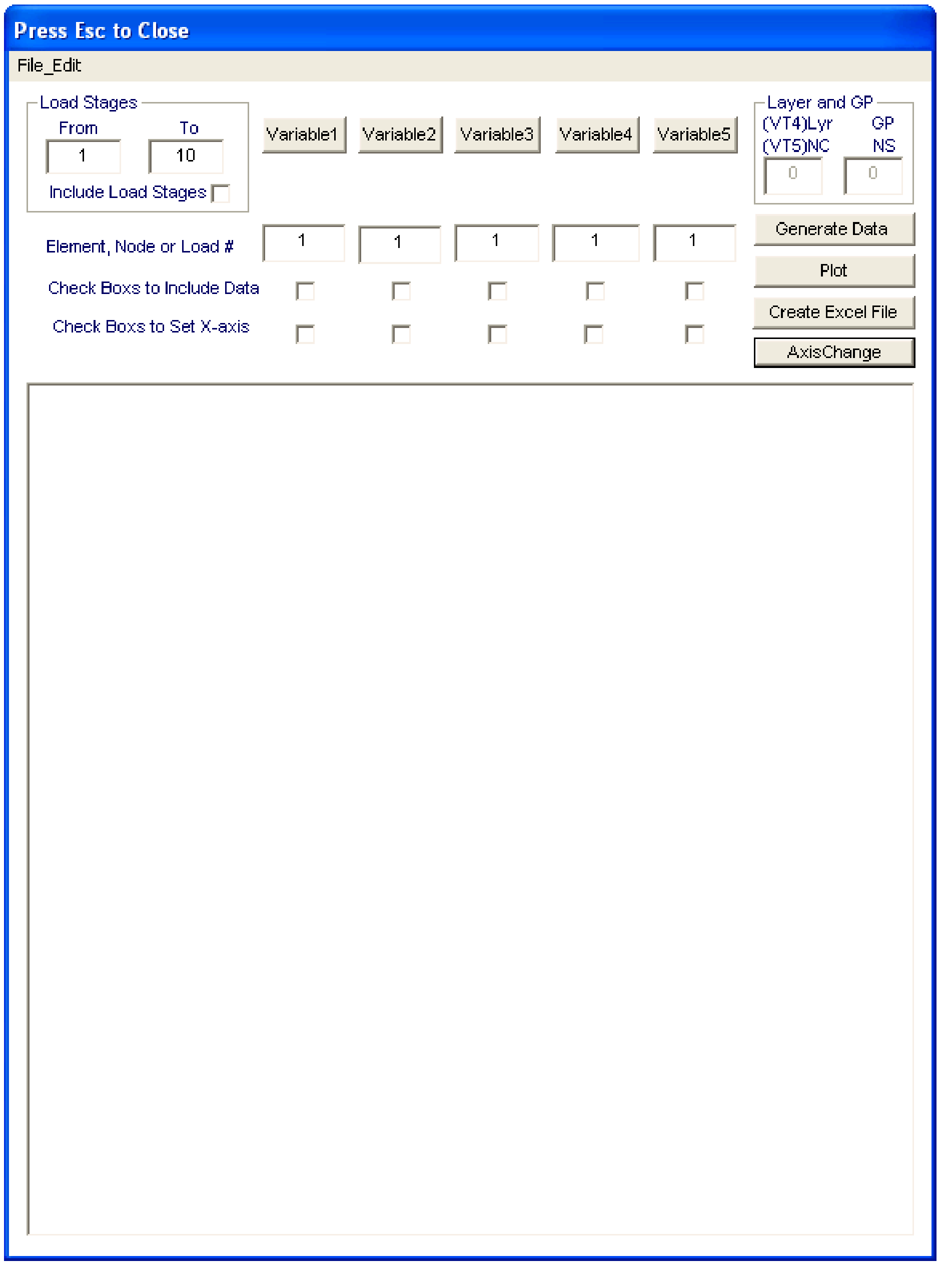


Figure 3.51: Data Platform Dialog

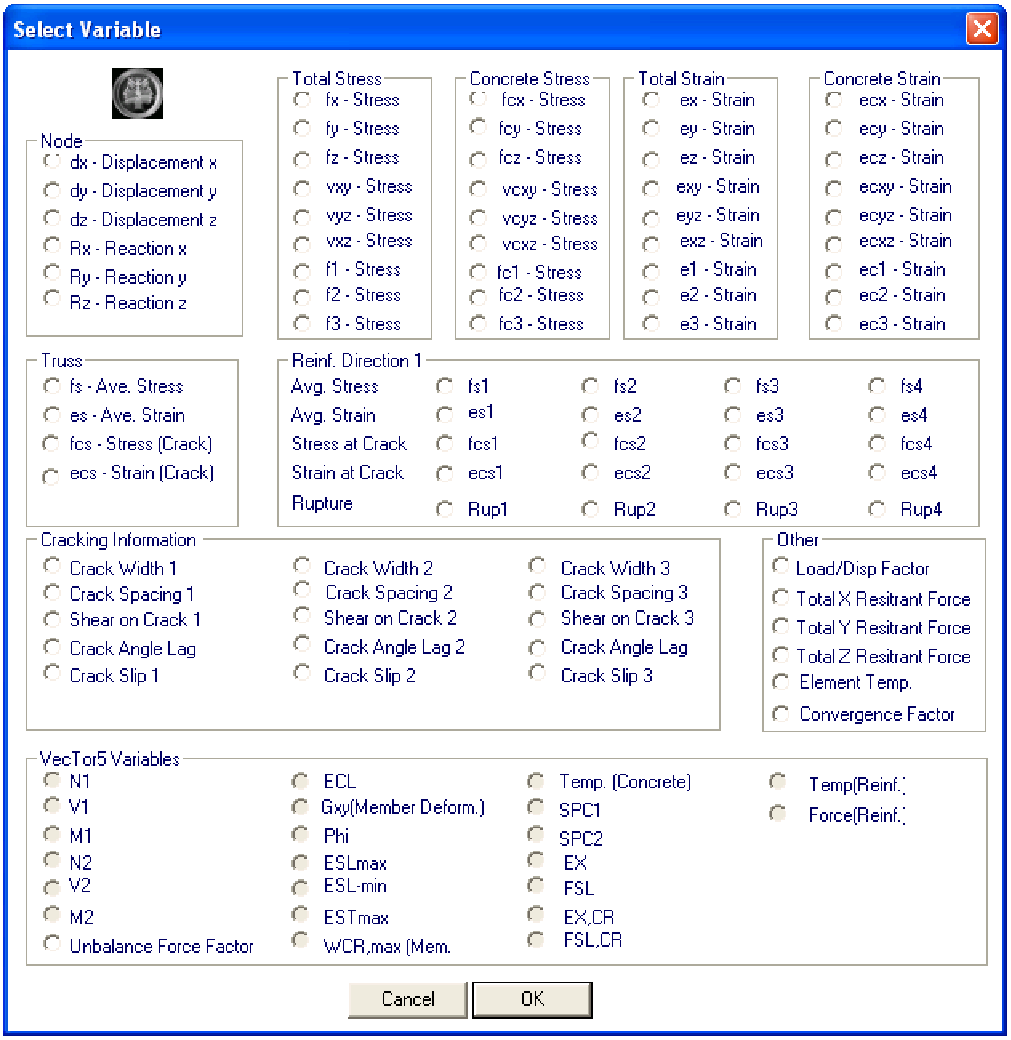


Figure 3.52: Variable Selection Dialog

3.9.3 Output Options

Once all appropriate values are selected in the *Data Platform* dialog, Janus hosts a variety of options for users to view the selected data range(s). Users may output the data within the *Data Platform* dialog as raw text, in a visual on-screen graphic plot, or as column-separated data in a external file as a Microsoft Excel format ".xls" spreadsheet.

3.9.3.1 Text Format

Upon selecting *Output Data*, data for currently selected variables in active input columns will be retrieved and displayed on screen according to their occurrence from left to right. Referring to Figure 3.53, the *Variable1*, *Variable3* and *Variable5* columns are active. The three selected variables, "Load", "fx", and "dx", are correspondingly printed as raw text data in the text field below. As specified from the *Load*

Stages area of the *Data Platform* dialog, the variable values are provided from load stages 1 through 10, inclusive.

3.9.3.2 Graphic Plot Format

Alternatively, users may opt to preview the selected data as a series of line plots. In order for the plot function to successfully execute, at least two variables (or a single variable and load stage) must already be specified. By selecting *Plot*, the Janus model space is converted to a 2D graph, with the selected variables plotted in separate colour-coded lines. By default, the horizontal axis represents incremental load stages, and the vertical axis represents the selected variable(s). Alternatively, one of the selected variables may be selected as the horizontal x-axis value. See Figure 3.54 below for an illustrative example of the same variables "Load", "fx", and "dx" plotted with respect to the load stage value.

3.9.3.3 Microsoft Excel File Format

Lastly, users may export the requested variable data in a column-separated format. This function may be convenient for subsequent graphing and data manipulation purposes that fall outside the capabilities

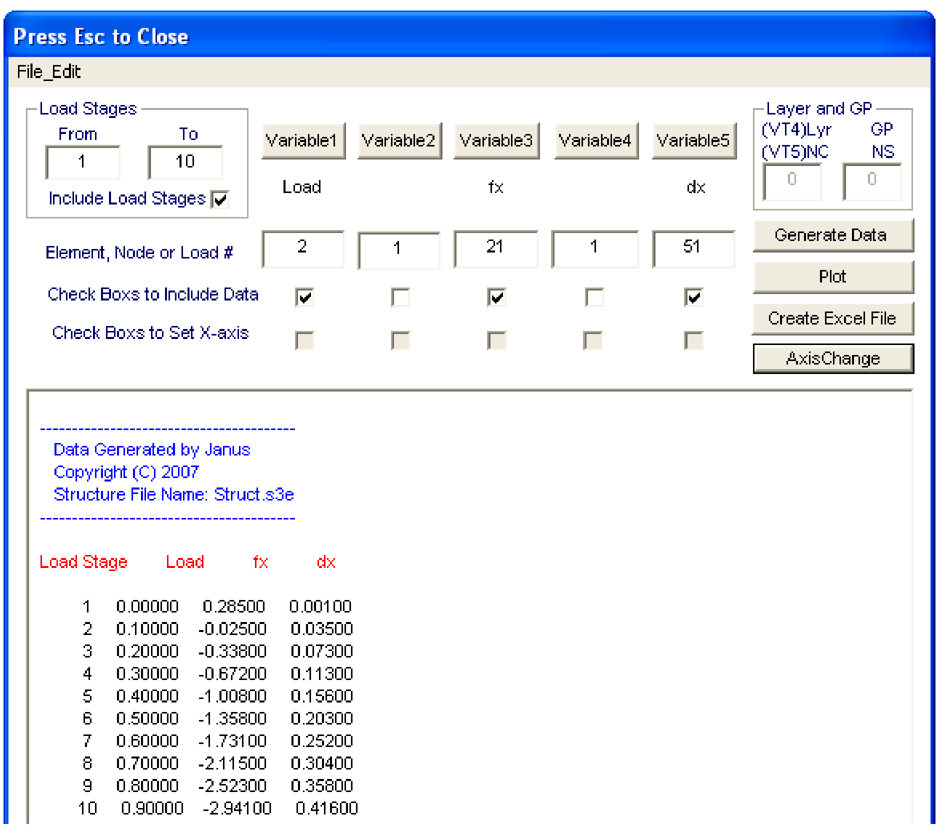


Figure 3.53: Data Platform Dialog

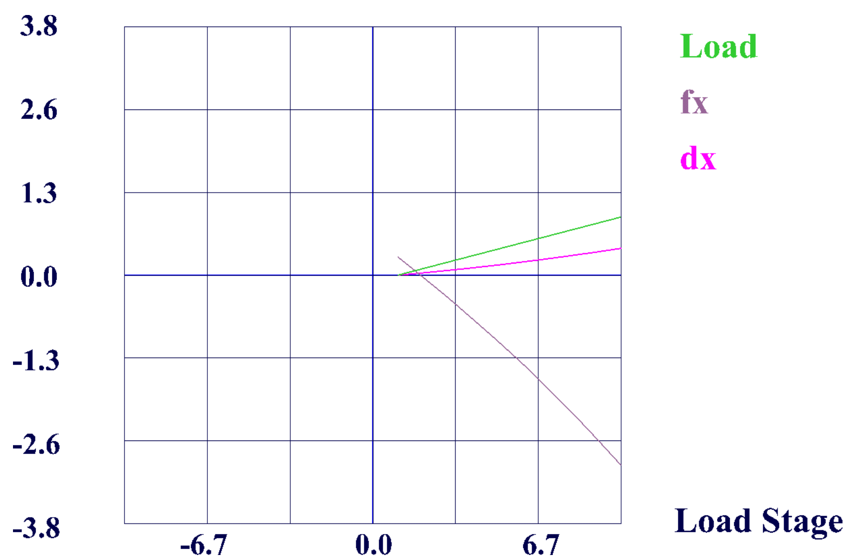


Figure 3.54: Data Platform Plotting Function

of Janus as a post-processor program. See Figure 3.55 for an equivalent example of the same structural data opened in the spreadsheet program Microsoft Excel.

3.9.3.4 Axis Change

Users may modify the graphic plot formatting using the *Axis Change* dialog. Refer to Figure 3.56 below for the available plot customization options. Users may modify data plot parameters by entering values corresponding to the edit boxes described in Table 3.9.

	A	B	C	D
1	Structure File Name: Struct.s3e			
2	Load Stage	Load	fx	dx
3				
4	1	0	0.285	0.001
5	2	0.1	-0.025	0.035
6	3	0.2	-0.338	0.073
7	4	0.3	-0.672	0.113
8	5	0.4	-1.008	0.156
9	6	0.5	-1.358	0.203
10	7	0.6	-1.731	0.252
11	8	0.7	-2.115	0.304
12	9	0.8	-2.523	0.358
13	10	0.9	-2.941	0.416

Figure 3.55: Data Platform Excel File Format

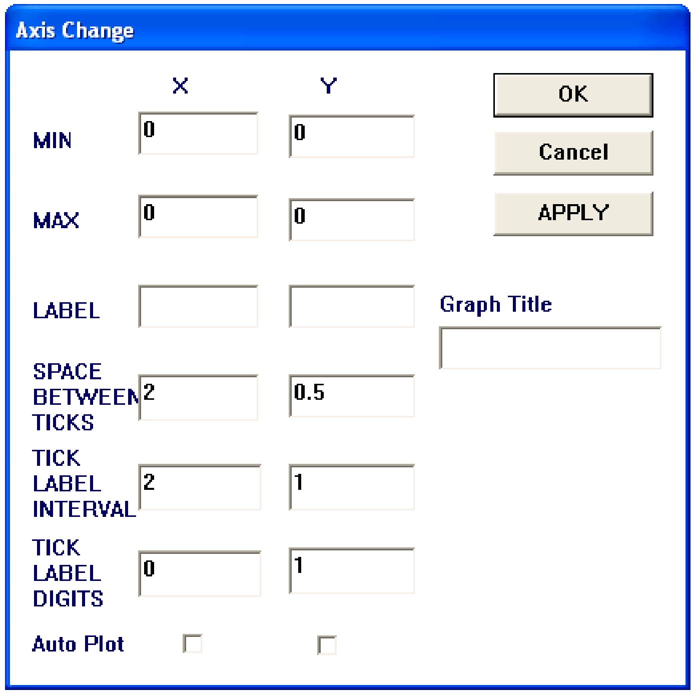


Figure 3.56: Axis Change Dialog

Edit Box Label	**Description**
MIN/MAX	Minimum and maximum x- and y-axis bounds
LABEL	x- and y-axis text labels
Graph Title	Graph title label
SPACE BETWEEN TICKS	Interval distance between graph ticks
TICK LABEL INTERVAL	Interval between tick labels
TICK LABEL DIGITS	Numerical precision of graph tick labels
Auto Plot	Resume default plot settings for respective axis

Table 3.9: Axis Change Dialog Functionality

3.10 Element Attributes

As previously mentioned in Subsection 3.3.4, the *Element Attributes* dialog is used to display the model data and current stress-strain state of a selected element.

3.10.1 Element Selection

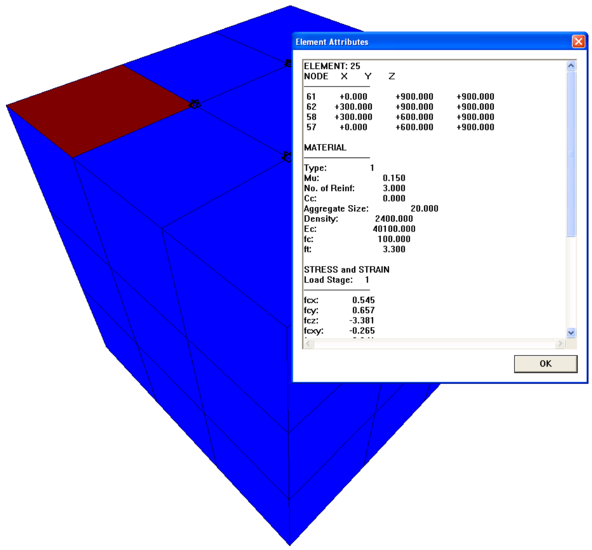
Currently, the *Element Attributes* dialog is only supported for right click selection of RC and truss elements within the global model view. The *Element Attributes* dialog only appears when the mouse is right clicked while the cursor is positioned over a valid RC element face or truss element line.

3.10.2 Element Attribute Data

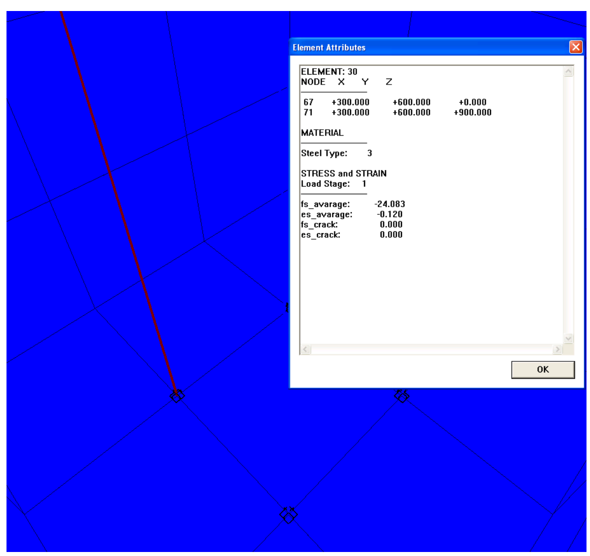
The *Element Attributes* dialog text field displays applicable model data and structural properties for the selected element and current load stage, including:

- Element number
- Element vertex node numbers and nodal coordinates
- Material specifications
- Stress and strain values

Refer to Figure 3.57 for demonstrations of the *Element Attribute* dialog with a solid RC element and truss element.



a) Solid RC Element



b) Truss Element

Figure 3.57: Element Attribute Dialog using Solid RC/Truss Elements of a VecTor3 Example Model

Chapter 4

VecTor2 Models in Janus

This chapter of the manual is provided to present Janus display features in the specific context of VecTor2 as a 2D finite element analysis program. Particular emphasis is provided in describing visualization tools that display VecTor2 model-related information in a distinct fashion from the other VecTor model types.

4.1 Appearance

Due to models being solely specified about the x-y plane, the graphical representation of VecTor2 models in Janus is customized according to the following subsections.

4.1.1 Axes

As a 2D analysis program, VecTor2 model nodes and elements are defined about the x- and y-axes, and are accordingly displayed as such in Janus. The z-axis is utilized for demonstrating out-of-plane RC material thickness in the *YZ Section* and *XZ Section* section view options.

4.1.2 Elements

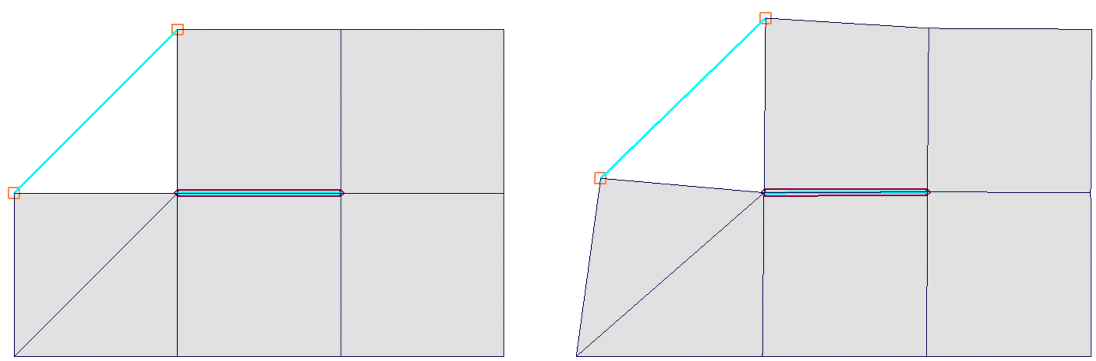
Models in VecTor2 consist of the following finite element types, with two linear degrees of freedom per node:

- three-noded triangular elements
- four-noded rectangular elements
- four-noded quadrilateral elements
- two-noded truss elements

- two-noded link and four-noded contact elements

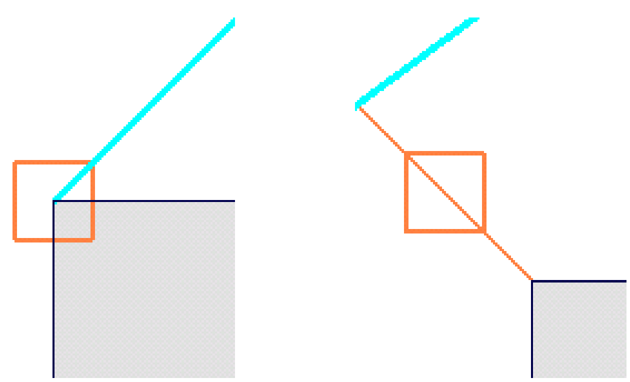
VecTor2 model elements are represented in Janus in a simple planar manner. Membrane elements such as triangular, rectangular and quadrilateral elements are displayed as geometric shapes with black outlines and coloured faces. The default face fill colour for RC membrane elements is dark green. For contrast from other rendered lines and objects, linear truss elements are displayed as thick-width lines, coloured in cyan by default. See Figure 4.1 for a typical view of VecTor2 elements represented in Janus.

Link elements are shown as a simple orange square box drawn around each pair of coincident model nodes. As specified in the VecTor2 and FormWorks User's Manual, one node of the link element is attributed with an RC element while the other corresponds with a connected truss element (Wong et al., 2012). Should slip occur and the nodes displace from each other, an orange line is drawn between the separated nodes to maintain visual association. The centroid of orange link element box symbol is positioned at the midpoint of the resulting line linking the two nodes. See Figure 4.2 below for a visual example of link element node displacement.



a) Standard Mode b) Deformations Mode

Figure 4.1: Simple VecTor2 Model in Deformations Mode



a) Standard Mode b) Deformations Mode

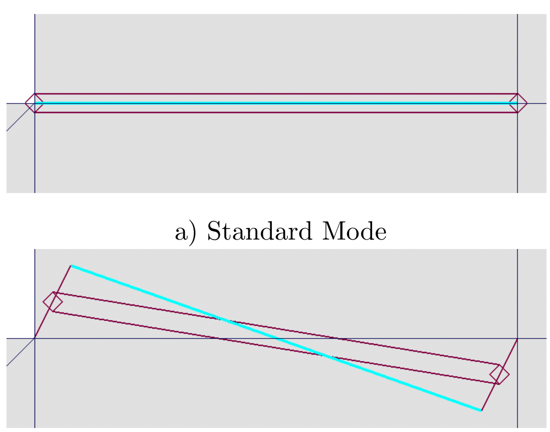
Figure 4.2: VecTor2 Link Elements

Lastly, contact elements are represented using a combination of simple shapes and a default dark red colour. A diamond outline is rendered at each of the two pairs of coincident nodes, denoting the contact interface between a truss element and its surrounding RC elements. In order to show each pair of diamonds as a single graphic symbol, corresponding dark red lines are also drawn between the each set of diamond outlines. In this way, the linear truss element associated with the contact element is visually bounded within the contact element lines. In the event that separation of either node pair occurs, an additional dark red line connects the displaced nodes, and the diamond outline for that node pair is positioned on the centroid of the newly drawn connecting line. For a sample demonstration of VecTor2 contact element deformations as displayed in Janus, see Figure 4.3 below.

4.1.3 Restraint Symbols

Nodal restraints in VecTor2 may be applied in the x- and/or y-directions. Singly restrained nodes are represented with roller symbols oriented in the direction of restraint, signifying that the node is permitted to "roll" in the transverse in-plane direction. For example, a node restrained exclusively in the y-direction would be represented by a roller restraint symbol positioned either above or below the node in the y-direction, showing that the node is still free to displace in the lateral x-direction.

Nodes restrained in both the x- and y-direction degrees of freedom are represented using pin support symbols. Unlike roller restraint symbols, the orientation of the pinned restraint symbol does not have any implications on the represented restraint directions - pin support symbols unconditionally depict that the indicated node is restrained in both the x- and y-directions. 2D pin and roller restraints in Janus may be viewed in Figure 4.4 below.



a) Standard Mode

b) Deformations Mode

Figure 4.3: VecTor2 Contact Elements

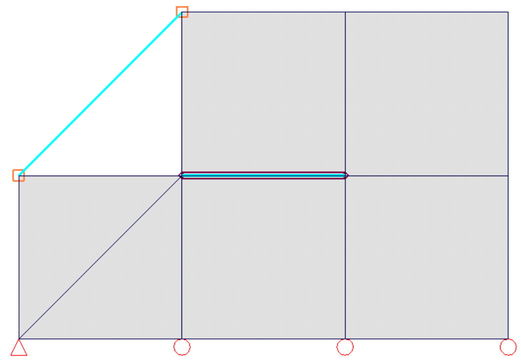


Figure 4.4: VecTor2 Restraints

4.1.4 Load and Displacement Symbols

VecTor2 models in Janus are capable of displaying nodal load cases as a series of nodal load and displacement symbols. Load and displacement symbols are represented as orthogonally-oriented, single-headed 2D arrows, drawn at the assigned node and in the direction of applied loading. As displayed in Figure 4.5, arrow symbols may interchangeably abutt the loaded node by either head or tail end, depending on: a) the general position of the loaded node in the model space, and b) the sign of the applied load or displacement value.

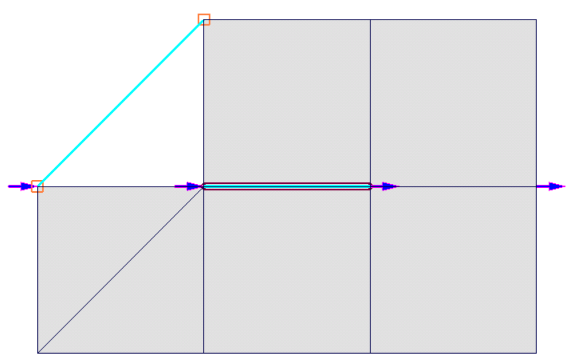


Figure 4.5: VecTor2 Load Arrows

4.2 Unique Features

For the purpose of displaying 2D VecTor2 models in Janus, several post-processing features are customized based on the implied planar context of the provided model and analysis parameters.

4.2.1 Deformations

Each node in VecTor2 is assigned two degrees of freedom, prescribed in the x- and y-directions. Janus correspondingly represents the orthogonal displacements of these model nodes as actual on-screen x- and y-axis node coordinate translations, linearly magnified by a uniform deformation scale factor (previously discussed in Section 3.6). Element vertices and/or node features associated with each node are accordingly re-rendered based on the displaced coordinates.

4.2.2 Crack Patterns

RC element crack patterns are reported on a similar planar basis, with the crack line rotated about the centroid of the element to represent the orientation of the crack with respect to the global model x-axis.

4.2.3 Section View

4.2.3.1 YZ Section and XZ Section View

Although VecTor2 is a two-dimensional analysis program, each RC membrane element is assigned a uniform out-of-plane thickness material property. Hence, in the interest of presenting VecTor2 sectional views in an analogous format to a 3D model, the out-of-plane thicknesses of VecTor2 RC elements are conceptually displayed as centred on the z-axis for applicable section views directly involving the z-dimension (i.e. *Section YZ* and *Section XZ* section views).

Linear and node-based elements such as truss, link and contact elements will be visible in section views involving the z-axis if they are intersected by, or lie on the entered section coordinate. Due to the fact that truss, link and contact elements lack an applicable out-of-plane thickness parameter, the sectional representation of these elements will be consistently drawn as centred on the z-axis and applicable x- or y-coordinate as determined through linear interpolation of the section coordinate value. Similarly, node-based features such as nodal loads and restraints may be viewed in Section View, but will only be visible when the input section coordinate and node coordinate coincide with the specified section plane. Refer to Figure 4.6 below for sample out-of-plane *XZ Section* and *YZ Section* section views of a VecTor2 example model.

4.2.3.2 XY Section View

Within the *XY Section* view, the out-of-plane thicknesses of RC membrane elements are indirectly represented based on the z-direction section coordinate choice. If the input section coordinate exceeds $\pm$(*element thickness/2*) for a particular material type, elements corresponding to that material type

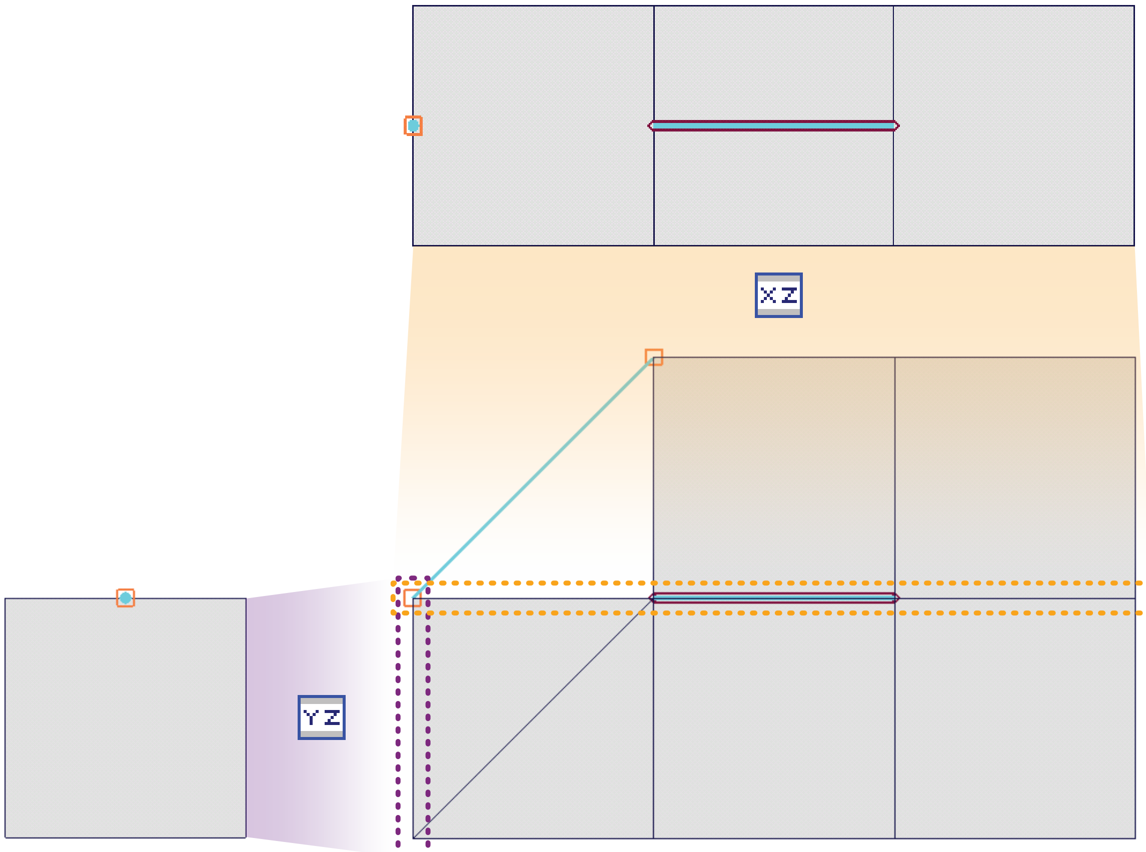


Figure 4.6: XZ Section and YZ Section View of a Simple VecTor2 Model

will not be visible. All other types of elements (truss, link, contact) and nodal features (load arrows, restraints) will only be visible at a z-coordinate of 0.

4.2.3.3 Sectional Deformations

Deformations mode may be used in conjunction with all Section View alternatives. Although Section View is enabled for all three orthogonal plane combinations in VecTor2, sectional deformations will only be demonstrated in the x- and y-directions. Although VecTor2 considers out-of-plane strains due to Poisson's effects, no out-of-plane displacements are provided as part of the analysis results. Regardless of deformation, the out-of-plane thicknesses of RC elements are constant and centred about the z-axis.

4.2.3.4 Sectional Crack Patterns

Due to RC element cracks only occurring in-plane within the VecTor2 analysis environment, crack patterns within Section View are only displayed using the *XY Section* view. RC element section cracks will only be rendered if the cracked element face is visible, i.e. the *XY Section* z-coordinate falls within the ±(*element thickness/2*) range of the cracked element's material type.

4.2.3.5 Sectional Contour Mode

With the exception of node-related contour mode variables (i.e. colour gradient representation of nodal displacements and reactions), all other solid-colour contour modes may be activated within Section View. For consistency with established Section View display mechanics, only applicable elements which lie on or are intersected by the requested section plane will be coloured according to the context of the selected contour mode variable. Contour mode in Section View is presented in an analogous manner to the contour mode in Global Model View. Selecting a RC element-related variable will result in coloured membrane element section faces, with truss and bond elements coloured in black. Conversely, truss and bond element-related contour modes result in colour-highlighting of the applicable element type, with RC membrane element sections portrayed as wireframes.

4.2.3.6 Sectional Hotspot Mode

All VecTor2 Hotspot mode variables are accessible within Section View. Within the context of Hotspot mode combined with Section View, node and element highlighting will only appear if the exact node coordinate or element boundaries are intersected by the section plane. For membrane RC elements, the section coordinate must fall between the local minimum and maximum range of coordinates of the element for the axis that the section is specified on. Similarly, highlighting of linear elements such as truss and contact elements will only appear if intersected by, or lie on the requested section plane. Highlighted nodes and point-based elements will only appear at the precise section coordinate that the nodes or elements lies on.

Chapter 5

VecTor3 Models in Janus

Compared to other VecTor model types, several user interface controls and visualization features for VecTor3 models have unique connotations when activated in Janus. The following chapter provide an overview of post-processing functions specifically customized for displaying VecTor3 models in a relevant and intuitive manner.

5.1 Appearance

VecTor3 is a 3D finite element modelling program with an extensive library of available element types. In order for Janus to visually display numerical data in 3D space for the diverse range of element shape geometry, some aspects of VecTor3 analysis output results are presented in Janus in a distinct manner compared to other VecTor programs.

5.1.1 Axes

In the pursuit of true graphic representation, Janus displays VecTor3 models as a series of rendered geometric shapes in 3D space. As previously described in Section 3.2.3.1, elements are established using user-defined coordinates read in from the associated VecTor3 expanded structure file. In order to provide consistent user interface controls for displaying and manipulating the 3D VecTor3 model in Janus, these exact model coordinates are universally scaled and transformed such that the resulting VecTor3 model is centrally positioned and appropriately magnified for view in Janus model space. Subsequently, model rotation controls using the right mouse button or *Set Camera View* dialog results in the VecTor3 model pivoting about the central model space axes.

5.1.2 Elements

Comparing the available element types among all VecTor programs, VecTor3 exhibits a wholly diverse range of solid RC element shapes as well as linear and point-based element types. VecTor3 consists of the following finite element types, with three linear degrees of freedom per node:

- six-noded wedge elements
- eight-noded regular hexahedral elements
- eight-noded isoparametric hexahedral elements
- two-noded truss elements
- two-noded link elements

In order to display solid RC elements in three dimensions, wedge and hexahedral elements are discretized as a series of triangular and/or quadrilateral polygonal faces; in turn, each polygon shape is further encapsulated as sets of nodal coordinates corresponding to each face vertex. Altogether, each set of element faces is used to collectively represent the stress- and strain-related properties corresponding to their respective element. See Figure 5.1 for an typical view of solid RC hexahedral elements in a RC element-related contour mode, displaying concrete strain values using constant coloured faces.

Within the Global Model View, Janus presents a simplified version of the overall structure by only rendering the outermost layer of RC solid element faces; for the purpose of conserving system memory and overall program performance, internal faces of RC solid elements are omitted from being rendered. The exclusion of internal RC element faces also facilitates visibility for internally specified non-RC finite elements such as truss and/or link elements, as well as relevant nodal features such as load arrows and restraint symbols. The exterior RC solid element faces of 3D VecTor3 models may be bypassed by zooming the camera view inwards, either through the use of the *Set Camera View* dialog or mouse track wheel scroll functionality.

In contrast to RC solid element faces being selectively chosen for rendering, all truss and link elements are displayed in Janus Global Model View. As simple linear elements, VecTor3 truss elements are displayed in a congruent visual manner to other VecTor programs - thick-width line segments established using two node coordinates, and coloured cyan by default.

Link elements in VecTor3 are represented in a similar visual manner to their VecTor2 link element counterpart, using the same symbolic orange colour and hollow square box icon. However, in the interest of facilitating visibility of link element symbols from any arbitrarily defined camera view position and orientation in model space, orange hollow boxes are rendered in all three planes around the pair of

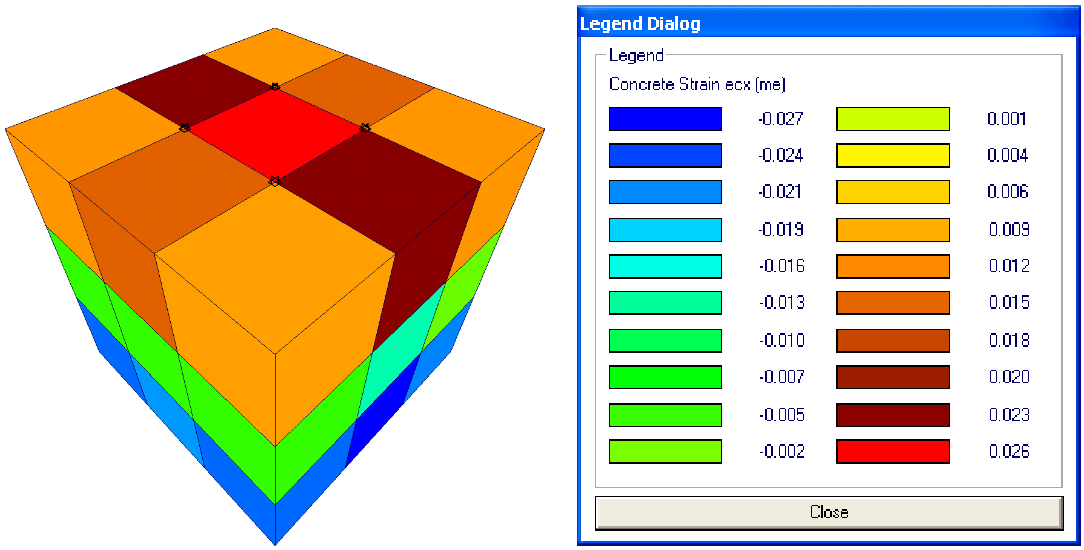
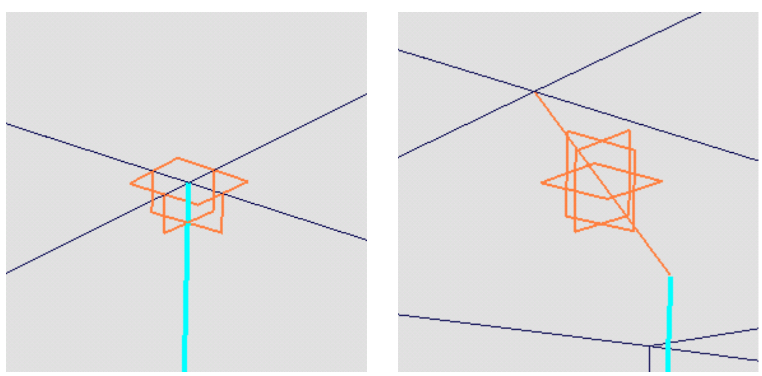


Figure 5.1: Simple VecTor3 Model in RC Element Contour Mode

coincident nodes which define the link element. In order to maintain visual association in the event that the two linked nodes slip and displace from each other, an orange line is drawn connecting the nodes. Subsequently, the trio of hollow orange boxes associated with the link element are positioned on the centroid of the line connecting the two linked nodes. For an illustrative example of a deformed link element, refer to Figure 5.2 below.



a) Standard Mode b) Deformations Mode

Figure 5.2: VecTor3 Link Elements

As a consequence of rendering external RC solid element faces as solid shapes in 3D, internally specified elements such as truss and link elements may be visually obscured by surrounding exterior solid RC element faces. As well, nodal features such as load arrow and restraint symbols may be similarly hidden within the VecTor3 structure. In order to overcome any obscured internal features, several user interface controls are available to the user in Janus. Firstly, external RC element faces may be bypassed by using the mouse track wheel scroll function to zoom in the camera view (see Subsection 3.3.4 for further instructions). Secondly, as discussed in Subsection 3.6.7.1, applicable Section View features may be used to obtain an isolated plane view of the internal element(s) of interest. Lastly, the *Toggle Elements* dialog demonstrated in Subsection 3.7.1 may be used to selectively disable RC solid element faces and/or wireframe as necessary.

5.1.3 Restraint Symbols

Nodal restraints in VecTor3 may be applied in all three orthogonal directions. As an extension of the 2D restraint symbols utilized in VecTor2, the representation of VecTor3 restraints in Janus uses pairs of nodal restraint symbols aligned in perpendicular orthogonal directions. Each planar restraint symbol independently acts as a simple 2D restraint, representing the restrained axis or axes for the plane that the particular restraint symbol is lying on. Relative to the actual restrained node, nodal restraint symbols are oriented in a manner that logically complies with the previously outlined conventions for representing restrained nodal movement.

For example, a node solely restrained in the z-direction is displayed in Janus using a pair of roller restraint symbols. For optimum visibility to the user, Janus may interchangeably display the pair of restraints as oriented either above or below the node in the z-direction. The pair of roller symbols aligned above/below the node in the z-direction is intended to indicate the following:

- In the x-z plane, the node is free to move in the x-direction and is restrained in the z-direction
- In the y-z plane, the node is free to move in the y-direction and is restrained in the z-direction

Accordingly, the substitution of a pin restraint for a roller restraint would represent an additional restrained degree of freedom in the plane that the pin symbol lies on. Hence, a combination of pin and roller symbols represents two restraints overall, while a restraint consisting of dual pinned restraint symbols represents a node restrained in all three orthogonal directions. See Figure 5.3 below for a depiction of nodal restraints in VecTor3.

5.1.4 Load and Displacement Symbols

Janus displays applied nodal loads and displacements as a series of 3D arrow symbols. For increased visibility in 3D space, load arrow symbols for VecTor3 models are drawn as a pair of overlapping single-headed 2D arrows, aligned on perpendicular axes. The direction and orientation of the load arrow symbol is established based on the orthogonal direction of loading, as well as the sign of the load value.

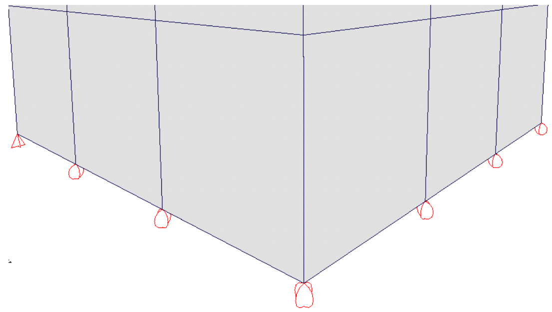


Figure 5.3: VecTor3 Restraints

See Figure 5.4 for an illustrative example of VecTor3 load arrows. In accordance with *Load Case* dialog display mechanics described previously in Subsection 3.7.3, both arrow shapes in the load arrow symbol will either appear as "empty" pink outlines, or "fill" with a solid colour corresponding to displayed range of load conditions in the *Load Case* dialog.

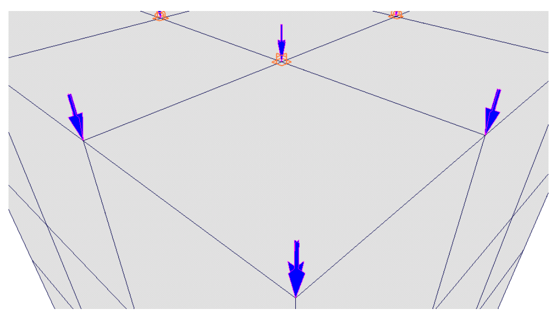


Figure 5.4: VecTor3 Load Arrows

5.2 Unique Features

5.2.1 Deformations

Nodes in VecTor3 analyses are capable of accommodating linear displacements in all x-, y-, and z-directions. Upon activation of Deformations mode in Janus, all constitutive element faces, edges, and/or vertices will be re-rendered using 3D nodal coordinates modified with the load stage-specific displacement values reported in the VecTor3 analysis output files. By default, all orthogonal displacements are magnified by a deformation scale factor of 30.

5.2.2 Crack Patterns

Among the VecTor software suite, VecTor3 is the only finite element analysis program which currently provides crack output data for 3D elements. Each crack is reported in terms of its orientation with respect to the global axes. In the context of VecTor3 analysis output files, the crack orientation is provided as a 3D unit vector comprised of directional cosine values. In order to represent such an internal element characteristic in three-dimensional space, the geometric components of the crack line vector are projected onto the planar surface of each visible element face. As previously described in Subsection 3.8.2, crack lines displayed on element faces are composed of two equal and opposite-direction line segments, extending outward from the centre of the element face. In doing so, the crack line is conceptually represented as a central single linear entity passing through the geometric centroid of the element. By default, the length of the crack lines rendered on each face is determined as a function of the crack line projection and the radius of a circle inscribed within the perimeter of the element face. The presentation of crack lines on each element face may be modified by invoking the *Set Crack View* dialog using **View** ▷ *Set Crack View*.

5.2.3 Section View

In order to accommodate the capability for VecTor3 finite elements to displace in all three orthogonal directions, Section View features are available for all three planes. Regardless of model orientation and implement element type(s), the *XY Section*, *YZ Section* and *XZ Section* toolbar buttons exhibit equivalent functionality in producing relevant plane section views of the opened VecTor3 model at the requested coordinate. Refer to Subsection 3.6.7.1 for a full description of Section View functionality. Upon activation of Section View, the *Section Up* and *Section Down* toolbar buttons may be used to traverse through node coordinate intervals pertaining to the third out-of-plane axis of the current section view. Section View is fully compatible with Deformations and Crack Pattern modes for VecTor3 models.

5.2.3.1 Sectional Deformations

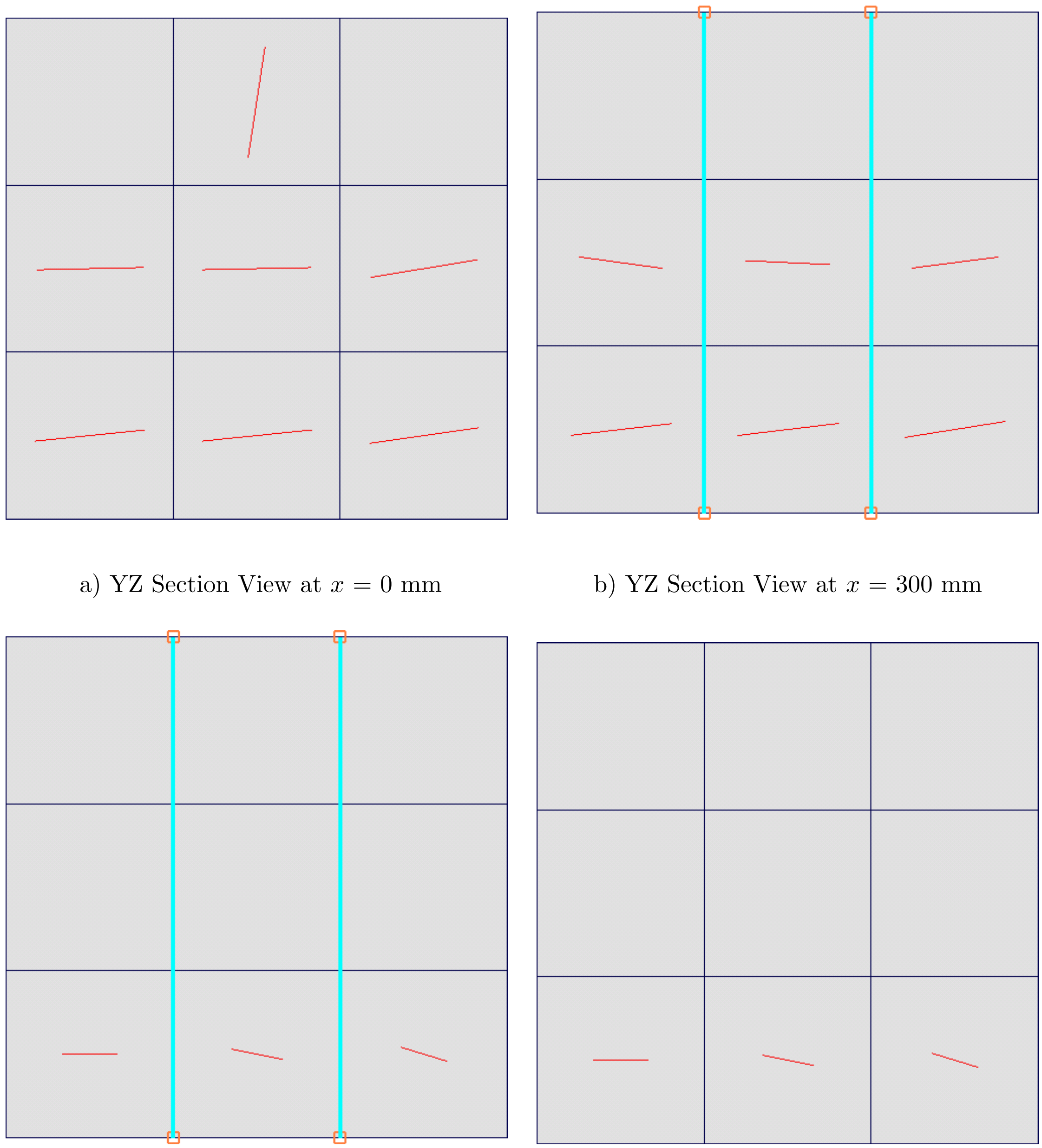
In the event that Section View and Deformations mode are enabled in tandem with a VecTor3 model, it is important to note that the resulting section view is produced based on nodal coordinate displacements in all dimensions. In addition to displaying planar deformations at the requested section coordinate, the out-of-plane coordinates of the model are affected by displacements as well. It should be noted that *Section Up* and *Section Down* options will be similarly affected by nodal displacements; section planes are drawn at each uniquely encountered displaced node coordinate value as the section input coordinate is incrementally increased or decreased.

5.2.3.2 Sectional Crack Patterns

Using the three-dimensional crack data provided in VecTor3 analysis output files, crack patterns are able to be displayed within Section View for VecTor3 models using the same planar projection methodology as the Global Model View. This feature is particularly useful for viewing the internal crack patterns within a model with multiple layers of elements, since the Janus Global Model View is only capable of displaying the crack lines on the exterior element faces for VecTor3 models. See Figure 5.5 for crack patterns displayed on a simple VecTor3 model using *YZ Section* view at various x-coordinate inputs.

5.2.3.3 Sectional Result Modes

Except for nodal displacement and reaction colour-gradient contour modes, all other result options found in the *Results* menu may be enabled within Section View for VecTor3 models in Janus. Equivalent functionality for Hotspot mode and displaying nodal features such as loads and restraints are also provided in Section View. Logically, only elements and nodal features which lie on or are intersected by the section plane will be available for displaying applicable result modes. Sectional result modes may used in combination with Deformations and Crack Pattern modes as well.



a) YZ Section View at x = 0 mm b) YZ Section View at x = 300 mm

c) YZ Section View at x = 600 mm d) YZ Section View at x = 900 mm

Figure 5.5: Sectional Crack Patterns of a Simple VecTor3 Model

Chapter 6

VecTor4 Models in Janus

In contrast to the conventional library of low-powered elements utilized in VecTor2, VecTor3, and VecTor6, VecTor4 employs high-powered nine-noded layered heterosis elements to represent plate and shell-related structures in a 3D environment. In addition to representing element stress-strain response characteristics on a structure-wide scale, the elements themselves are internally stratified into discrete layers of reinforced concrete and reinforcement - enabling the capability for presenting sectional analysis results. In anticipation of the additional post-processing needs attributed to this unique form of finite element, customized facilities are provided in Janus for the display and visualization of VecTor4 models and their associated analysis results.

6.1 Appearance

Within the context of visualization, the most marked distinction of VecTor4 lies in its exclusive use of a nine-noded degenerated heterosis element. Each element is comprised of eight side nodes (four corner nodes and four midpoints) as well as a ninth central node. VecTor4 shell elements are conceptually defined on a centreline basis, with the nine node points establishing the intermediate mid-depth surface of the shell element. For modelling purposes, each of the nine nodes is associated with an corresponding pair of top and bottom element vertices "above" and "below" the node in the v_3 direction by a constant depth of $\pm$(*element thickness/2*), where the element thickness is a user-defined RC material property. See Subsection 2.7.6 for a general overview of the nodal coordinate system in VecTor4 used for establishing the positions of top and bottom shell element vertices.

6.1.1 Axes

In a likewise manner to displaying 3D finite element models created in VecTor3, Janus portrays VecTor4 models in a simulated three-dimensional form. Actual nodal coordinates and element definitions are extracted from the VecTor4 expanded structure file and converted to equivalent standardized values for rendering in Janus model space. In order to provide consistent model rotational controls, VecTor4 model coordinates are transformed such that the resulting on-screen model in Janus is centred on the origin of the modelling axis system. Subsequent rotations of the model using the middle track wheel button or *Set Camera View* dialog will result in the VecTor4 model centrally pivoting about the specified axis or axes.

6.1.2 Elements

As emphasized at the beginning of this chapter, shell elements are the feature element of discussion in the visualization of VecTor4 models. The expanded structure files produced by VecTor4 explicitly declare shell element vertices using the basis of top and bottom node coordinate pairs. Since Janus utilizes data from the VecTor4 expanded structure file in order to render actual element and nodal coordinate information, each shell element is essentially modelled using data from a total of 18 node points - nine from each of the top and bottom vertex layers. For 3D representation of shell element faces in the Global Model View, compound nine-noded polygon shapes are rendered at top and bottom layers of nodes, with additional six-noded side faces used to connect top corner and midpoint nodes to their corresponding bottom nodes. In Global Model View, only the exterior-most top, bottom and side shell element faces will be rendered for display purposes.

Although VecTor4 shell elements are ideally intended to represent curved reinforced concrete surfaces, geometric edges and faces of VecTor4 shell elements are approximated using simplified linear relationships between nodes in Janus. Although not truly representative of the actual curves simulated by the shell element, the overall constitutive structural behaviour that can be captured through top and bottom node displacements and/or rotations is sufficient for the overarching majority of VecTor4 post-processing visualization purposes in Janus. Refer to Figure 6.1 below for a depiction of a simple VecTor4 model consisting of four shell elements undergoing nodal rotations and displacements as a result of pure moments applied to the extreme element edges.

Within each shell element, the out-of-plane depth is stratified into a series of RC and smeared reinforcement layers as specified by the user. The RC layers are contiguously defined throughout the depth of the element, where the respective layer thicknesses in the v_3 direction at each node are determined by

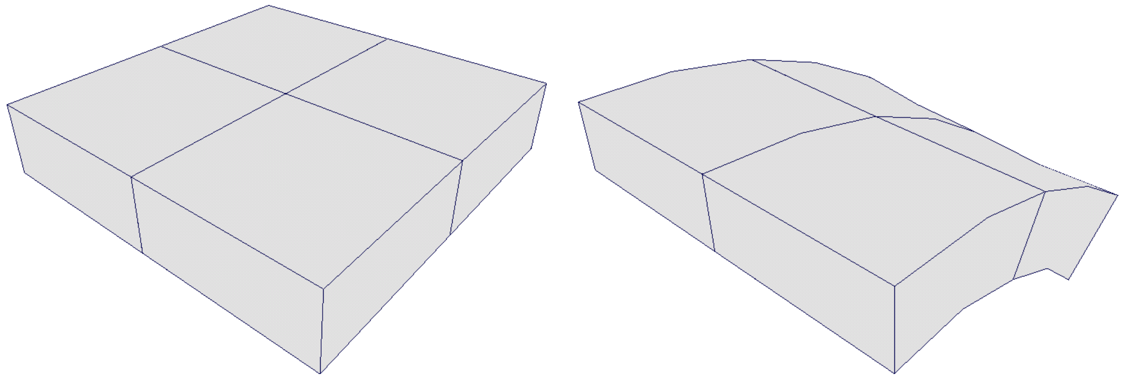


Figure 6.1: Simple VecTor4 Model in Deformations Mode

VecTor4 as a culmination of a variety of factors: the overall element thickness, specified concrete cover depth parameter, and the express requirement to present a whole number of evenly spaced layers. In contrast, the centroidal locations of smeared reinforcement layers may be established at any interstitial depth between the top and bottom faces of the element, independent of established RC layer boundaries. Instead of utilizing the nodal coordinate system for establishing the out-of-plane element thickness direction, the in-plane orientation of each smeared reinforcement layer utilizes the local coordinate system established at each Gauss Point. Using right-hand rule notation, the reinforcement layer orientation is specified as an angle rotated about the positive local z' axis, and relative to the local positive x' direction at each Gauss Point. Janus uses the local x' and z' vectors at the central shell element Gauss Point (denoted as Gauss Point 5) as the representative axes for displaying the relative in-plane direction of reinforcement for each smeared reinforcement layer. Refer to Subsection 2.7.6 for an overview of the local coordinate systems utilized in VecTor4.

It is important to note that these internally specified element RC and reinforcement layers are not visible from neither the Global Model View nor Section View. As subsequently explained in Subsection 6.2.4, the view of layers and the associated sectional performance of individual shell elements must be accessed through the *Layer View* toolbar button.

6.1.3 Restraint Symbols

As a 3D analysis program, VecTor4 is capable of assigning all conventional linear nodal restraints in the global x-, y-, and z-directions. These restraints are displayed in Janus in an analogous fashion to VecTor3, utilizing pairs of orthogonally oriented roller and/or pin restraints to represent restrained degree(s) of freedom for the respective axes that the symbols lie on. Although shell element vertices are defined on a top and bottom layer basis in Janus, nodal restraints in the global x-, y-, and z-directions are still rendered true to the actual node coordinate location for visual clarity and consistent restraint symbolism.

In addition to conventional linear displacement restraints, VecTor4 is capable of employing fixed support conditions which prevent a node from rotating in-plane about its local θ_1 and/or θ_2 rotation vectors. To illustrate that a node is being rotationally restrained, VecTor4 rotational restraints are presented as a pair of right-angled triangles drawn at the top and bottom element vertices associated with the restrained node. The triangles are both drawn with a edge flush with the element side face, suggesting that the entire shell element edge is rotationally supported at that node. To visually communicate the direction of resistance, each set of triangle symbols is rendered on the plane that rotation is being resisted. See Figure 6.2 below for an example of a simple VecTor4 model in Janus representing fixed shell element edges using a combination of 3D pin and fixed node restraint symbols.

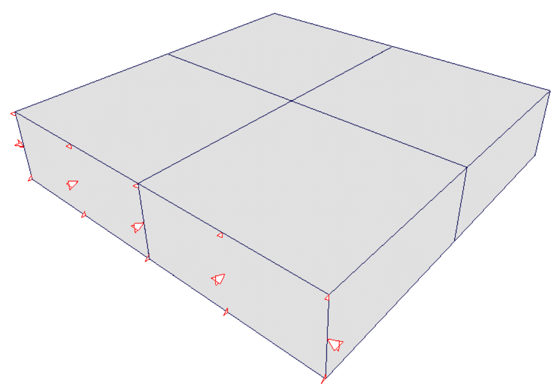


Figure 6.2: VecTor4 Restraints

6.1.4 Load/Moment and Displacement/Rotation Symbols

In an equivalent fashion to nodal restraints, VecTor4 analyses can support applied nodal loads and displacements in all three x-, y-, and z-directions. Janus displays such VecTor4 node features in a congruent manner to VecTor3, utilizing a load symbol consisting of a pair of orthogonally oriented single-headed arrows. The orientation of the load arrows is based on the prescribed direction of loading, as well as the numerical sense of the load value.

As well as having provisions for assigning linear nodal displacements in the x-, y-, and z-directions, VecTor4 also possesses the unique capability of determining the in-plane nodal rotations throughout each shell element. At each load stage, resulting rotation values are provided in the θ_1- and θ_2-directions, corresponding to planar rotations about node coordinate system axes v_2 and v_1, respectively. In order to accommodate for any arbitrary local node axis orientation - and, accordingly, rotation vector direction

- moment and rotation arrows are rendered in Janus as double-headed arrow symbols, using right-hand rule notation to signify the direction of rotation. At each applied nodal moment or rotation, double-headed arrow symbols are drawn in both orthogonal orientations in order to provide optimal visibility from any camera view in 3D space. For consistency in loads and rotations being assigned directly to a node, both single- and double-headed arrows are rendered at the exact mid-depth node location of the shell element. Figure 6.3 provides a visual depiction of 3D single-headed load and double-headed moment arrows applied to the shell element edges of a simple VecTor4 model.

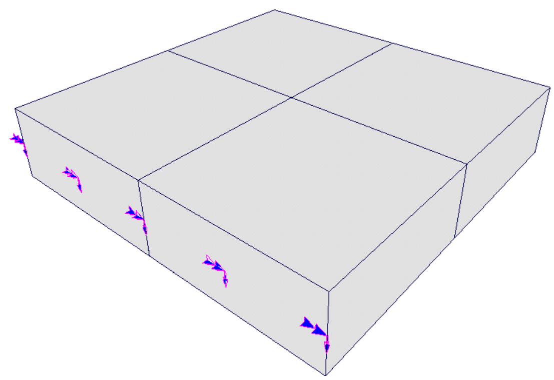


Figure 6.3: VecTor4 Load and Moment Arrows

6.2 Unique Features

6.2.1 Deformations

As previously described in Subsection 3.8.1.1, Janus is capable of displaying VecTor4 nodal rotation results directly on the rendered model as an enhancement of the Deformations mode (accessed via **Results** ▹ *Deformations*). Nodal rotations are represented as top and bottom element vertices rotating about their associated mid-depth node using a constant radius of (*element thickness/2*). The resulting rotational displacements are decomposed into corresponding x-, y-, and z-direction components, with relevant signs to represent positive and negative vertex rotations about the mid-depth node. The rotational displacement decomposition procedure is completed for both local in-plane rotation directions at each node. Accordingly, the orthogonal rotational displacement component values are universally magnified using the same deformation scale factor applied to linear nodal displacements. Lastly, the scaled components are correspondingly added to the similarly scaled linear displacements for each node, and the net displacement values are applied to the respective top and bottom element vertices. The deformation scale factor may be modified via **View** ▹ *Set Deform. Scale* ▹ *Select Scale Factor.*

6.2.2 Contour Mode

When viewing stress- or strain-related parameters for VecTor4 models, RC element stress and strain properties are specified on a layer and Gauss point basis. Within Global Model View, users must specify a layer number and Gauss point number via edit boxes in the VecTor4 *Legend* dialog, as shown in Figure 6.4 below. The + and - buttons immediately adjacent to the *Concrete/Reinf. Layer Select* edit box incrementally increase or decrease the current layer value between 1 and the maximum number of layers observed among all present RC material types. By convention, layer 1 is denoted as the topmost layer of the shell element, and layers are incrementally numbered with increasing depth through the section. Depending on the context of the variable selected in contour mode, the layer edit box may be used to specify either RC or smeared reinforcement layer numbers. The default Gauss Point value of 5 corresponds to the central mid-depth Gauss Point in each 3 × 3 array of Gauss points for the shell element. If a shell element is assigned a material type with a total number of RC or smeared reinforcement layers that is lower than the value requested in the VecTor4 *Legend* dialog, that element will be coloured black.

Smeared reinforcement contour modes for VecTor4 models are activated using options in **Results** ▷ *Reinforcement* ▷ *Direction 1* or *Direction 2*. While the average stress and strain values will be congruent for either direction, VecTor4 smeared reinforcement stress and strain values at the crack are differentiated using the distinct *Direction 1* and *Direction 2* options.

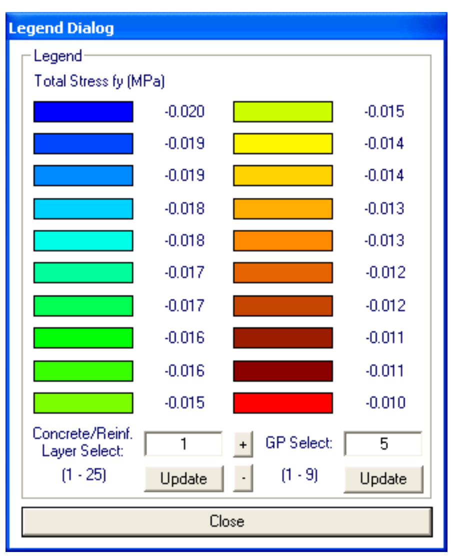


Figure 6.4: VecTor4 Legend Dialog

6.2.3 Section View

The available Section View options for 3D VecTor4 models are implemented in a congruent fashion as the VecTor3 Section View facilities - *Section XY*, *Section YZ*, and *Section XZ* toolbar buttons all produce relevant planar section views based on the requested out-of-plane section coordinate. As a 3D modelling program, VecTor4 shell elements may displace in all three directional axes. While Deformations mode is enabled, it is important to recognize that the resulting model coordinates are displaced in all three axes; the requested section coordinate will generate a section view based on deformed model geometry. Additionally, it is worthy to note that deformations associated with nodal rotations are capable of being displayed within relevant section views. Utilizing linear interpolation, Section View will present an analogous planar projection of all rotated and/or displaced shell element edges, faces, and vertices which intersect or lie on the section plane.

6.2.4 VecTor4 Layer View

Janus provides a unique feature for displaying the intermediate through-depth layers of selected VecTor4 shell elements, known as Layer View. Layer View is activated using the *Layer View* toolbar button, invoking an intermediate dialog which allows users to select the shell element of interest via drop-down menu list. The numerical listing of shell elements corresponds to the element numbers assigned to each shell element in the expanded structure file for the opened VecTor4 model.

Upon selection of a shell element, Layer View renders an isolated and magnified 3D version of the shell element in model space. The element is drawn true to the proportions specified within the structure file, but also enlarged to fill the extent of Janus model space that the entire VecTor4 model conventionally occupies. This modification is deemed necessary for the purpose of differentiating between the minute widths between RC layers, and is particularly helpful for viewing a single diminutive shell element from a large-scale model. For convenient and consistent rotation controls in Layer View, the sole Layer View shell element is also centred in Janus model space.

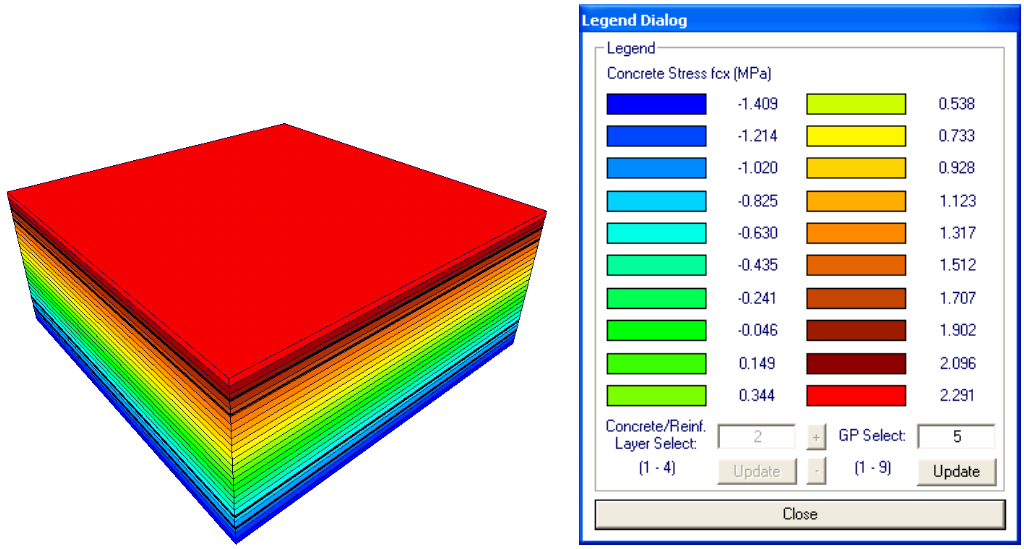
Using top and bottom vertex coordinate data extracted from the VecTor4 expanded structure file, discrete RC layer faces and lines are proportionally drawn across the side (i.e. through-depth) faces of the element. Accordingly, smeared reinforcement layers are established as a thick-lined perimeter line at the corresponding depth that the reinforcement layer centroid is designated at. In order to visually communicate the general reinforcement orientation for each smeared reinforcement layer, a thick line is drawn in the direction of reinforcement alignment. The line passes through the central depth of the element and spans between opposing ends of the reinforcement layer perimeter. By default, all RC layers are coloured dark green, and the thick lines representing smeared reinforcement layers are cyan. Similar to the display mechanics employed in 3D Global Model View, the isolated shell element is drawn as a hollow prismatic solid. To view smeared reinforcement orientation, users must utilize the track button scroll function to zoom in the camera view and bypass the exterior layer faces.

Within Layer View, the RC layer faces behave in an analogous manner to standard element faces for portraying RC-related result modes, whether showing as a colour gradient for contour modes, or red/brown/empty wireframe for applicable Hotspot mode parameters. Once a smeared reinforcement-related result mode is selected, RC layer faces are presented as hollow wireframes. Accordingly, the thick smeared reinforcement lines are assigned colours relevant to the selected result mode. By default, Layer View contour mode presents the layer-specific stress-strain value attributed to Gauss Point 5. Once activated, Layer View may be toggled on and off using the *Toggle 3D View* toolbar button, as previously described in Table 3.4. Upon resuming Layer View via *Toggle 3D View*, the most recent shell element selected for Layer View will be automatically recalled for user convenience. See Figure 6.5 and Figure 6.6 below for a complete demonstration of RC and smeared reinforcement contour mode and the associated Layer View of a specified element in a simple VecTor4 model.

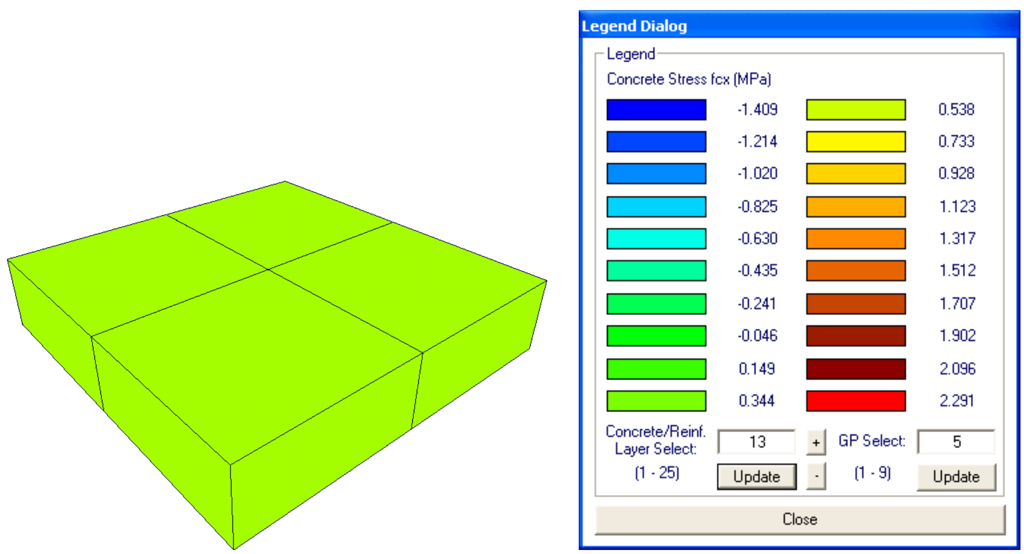
6.2.5 Gauss Points Mode

As initially described in Subsection 3.7.6, Janus hosts a dedicated function for displaying VecTor4 shell element Gauss Points, activated via **Structure** ▷ *Gauss Points (V4)*. Activation of the Gauss Points mode is dependent on successfully locating and reading the VecTor4-specific Gauss Point list (".GPL") file in the same local file directory that the opened job file is contained in. In the event that the Gauss Point List file is not found during the Janus file-opening procedure for VecTor4 models, a relevant disclaimer message will appear to notify the user that the Gauss Point List file is missing. As well, the **Structures** menu option to display Gauss Points will be disabled from being selected.

Each shell element utilizes a 3×3 array of Gauss Points for numerical integration purposes, positioned at the mid-depth surface through the shell element. Gauss Points are sequentially identified as Gauss Points 1 through 9, with Gauss Point 5 consistently representing the central Gauss Point of the element. Upon activation of the Gauss Points mode, all shell elements in the VecTor4 model are simplified into wireframe shapes. The positions of Gauss Points 1 through 9 for each shell element are displayed as a series of distinctly coloured points within each shell element wireframe. An instance of the *Legend* dialog appears for the purpose of attributing the numerical identifier per Gauss Point to the colour of the Gauss Point in Global Model View.

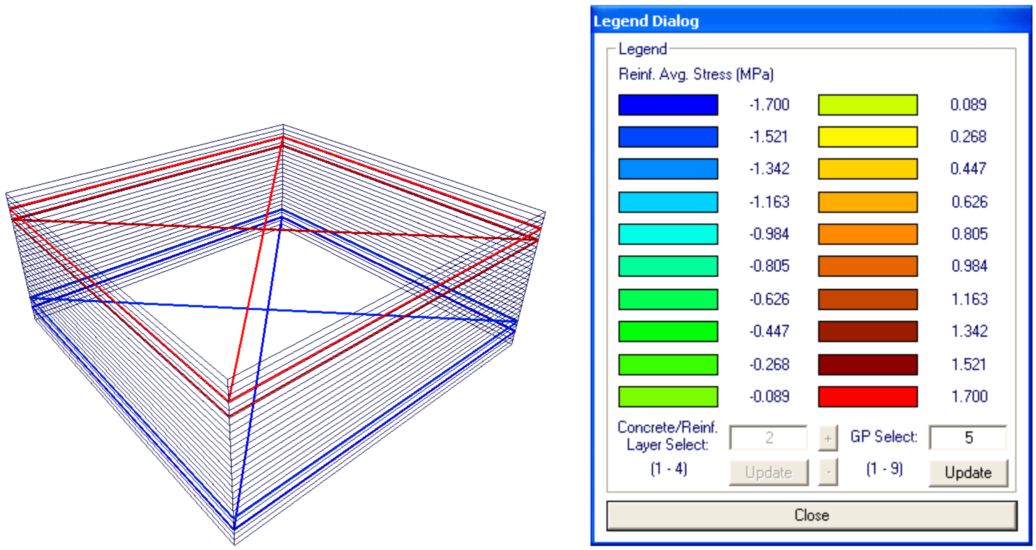


a) Layer View of Shell Element 1, RC Layer Contour Mode

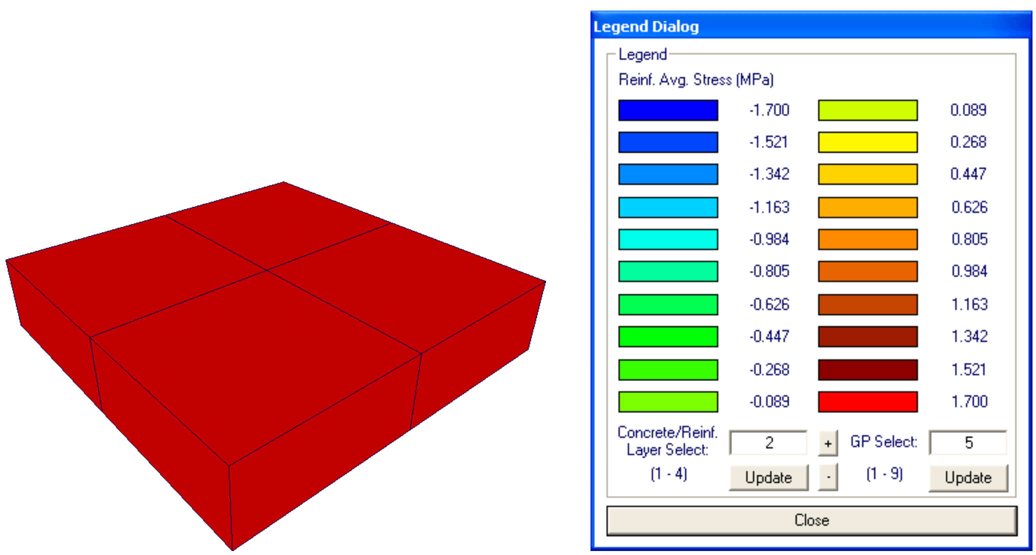


b) Global Model View of RC Layer 13, RC Element Contour Mode

Figure 6.5: RC Layer View and Global Model View Using a Simple VecTor4 Model



a) Layer View of Element 1, Smeared Reinforcement Contour Mode



b) Global Model View of Smeared Reinforcement Layer 2, Reinforcement Contour Mode

Figure 6.6: Smeared Reinforcement Layer View and Global Model View Using a Simple VecTor4 Model

Chapter 7

VecTor5 Models in Janus

VecTor5 is a 2D finite element analysis program designated for the analysis of RC planar frames with sectional analysis capabilities. In comparison to other contemporary VecTor programs, VecTor5 differs by representing frame structures as finite element models composed entirely of two-noded member elements. In addition, VecTor5 is able to output comprehensive sectional stress-strain response characteristics for user-specified member elements. Lastly, VecTor5 also distinguishes itself from other 2D-based VecTor programs (i.e. VecTor2 and VecTor6) by being able to accommodate the unique combination of in-plane nodal displacements as well as in-plane rotations in its analysis procedures. The following sections describe the ways in which Janus post-processing facilities are customized for visualizing VecTor5 models.

7.1 Appearance

VecTor5 is utilized for analyzing two-dimensional RC frame structures. As such, finite element frame models in VecTor5 are solely composed of two-noded member elements. In a likewise manner, Janus presents VecTor5 models as simple 2D frames, using thick width lines of constant thickness to represent each member element defined by the user. As such, the in-plane width of the member elements are not represented in the Global Model View.

7.1.1 Axes

As a two-dimensional analysis program, VecTor5 utilizes the x- and y-axes as the defining plane for establishing frame model parameters. Correspondingly, Janus exclusively uses the same x-y plane for all VecTor5 post-processing purposes. The x- and y-coordinates of member elements are directly interpreted by Janus and displayed in Global Model View. The VecTor5 Layer View also utilizes the same x-y plane for displaying all member element sections. Within VecTor5 result files, the sole use of the z-axis is attributed to specifying the out-of-plane direction of rotation-related analysis results as well as nodal restraints. Janus converts all z-axis related values and presents them in an analogous planar format on the x-y axes. As such, the z-axis is not utilized for any VecTor5-related visualization purposes in Janus.

7.1.2 Elements

VecTor5 frame models consist entirely of linear two-noded RC member elements, capable of accounting for end displacements in the x- and y-directions as well as in-plane rotations. Janus represents all VecTor5 member elements in Global Model View as thick line elements of constant width. However, each member element is internally characterized by constant sectional properties established by the user. Elements are stratified into a series of contiguous RC layers through the depth of the section, with each layer individually specified using constant thickness and width parameters. Ultimately, the RC components of the member element are represented as a series of discrete regular rectangular "slices" of varying widths and thicknesses, connected together as a unified section. Member element sectional properties are also capable of including specifications for reinforcement in all orthogonal directions. Instances of longitudinal reinforcement (i.e. reinforcement bars oriented parallel with the axial direction of the member element) are identified as discrete layers, and may be established at any interstitial depth between the top and bottom surfaces of the section. In contrast, transverse and out-of-plane reinforcement are treated as smeared reinforcement properties. As such, transverse and out-of-plane reinforcement are assigned to individual RC layers and perform as a function of the specified section dimensions.

Within the VecTor5 structure file (file extension ".S5R"), users may specify individual member elements as as part of the "Detailed Member Output List" for sectional analysis purposes. This list of output member element(s) instructs VecTor5 to include the comprehensive sectional stress-strain response of each identified member as part of the resulting analysis output file(s). Janus uses different coloured lines in order to distinguish output member elements from plain member elements. With no result modes enabled in Janus, plain member elements are coloured in cyan by default in Global Model View. In contrast, output member elements are coloured purple. All member elements are displayed with the same line thickness, whether or not they are selected by the user as an output member. Refer to Figure 7.1 for a depiction of a simple VecTor5 model with and without Deformations Mode enabled, with two member elements specified as output member elements.

a) Standard Mode

b) Deformations Mode

Figure 7.1: Simple VecTor5 Model in Deformations Mode

7.1.3 Restraint Symbols

Nodes in VecTor5 analyses are each capable of exhibiting three degrees of freedom: linear displacements in the x- and y-directions as well as in-plane rotations about the z-axis. Accordingly, nodal restraints may also be applied to each node for each associated degree of freedom. In an identical fashion to how Janus displays restraints for other 2D VecTor programs, linear restraints in the x- and/or y-directions are represented as planar roller and/or pin restraint symbols. See Subsection 4.1.3 for an explanation of how two-dimensional roller and pin symbols are implemented in Janus. In order to represent rotational restraint assignments for VecTor5 models, Janus renders a pair of right-angled triangles at each rotationally fixed node. See Figure 7.2 for a illustrated example of a rotational restraint symbol and roller support symbol superimposed upon a simple VecTor5 model in Janus. In the provided example, the combined rotational restraint and vertical roller restraint in the y-direction denote that the node is only capable of exhibiting displacements in the lateral x-direction.

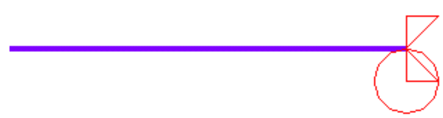


Figure 7.2: VecTor5 Restraints

7.1.4 Load/Moment and Displacement/Rotation Symbols

Amongst the current series of VecTor programs, VecTor5 is exclusive in its ability to accommodate for in-plane nodal rotations as a 2D analysis program. In addition to displaying conventional nodal forces and displacements using single-headed 2D arrows (akin to VecTor2 and VecTor6), users may also assign in-plane applied moments and rotations about the z-axis to frame nodes. Since VecTor5 models are exclusively presented about the x- and y-axes in Janus, special in-plane load case symbols are required to represent the moments and applied rotations assigned to nodes. As with any other VecTor program, load cases may be selected for viewing by choosing an active option in **Structure** ▷ *Load Cases*. See Figure 7.3 below for an example of a nodal rotation symbol superimposed upon a simple VecTor5 model with Deformations mode activated.

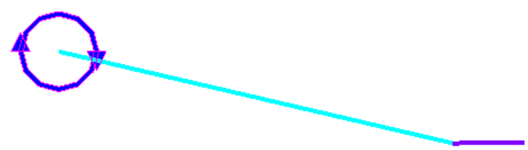


Figure 7.3: VecTor5 Rotation Load Arrow

7.2 Unique Features

In order to present the unique results format arising from the combined sectional and global analysis capabilities for VecTor5 frame structures, the post-processing features for VecTor5 models in Janus must be accordingly customized such that the analysis results may be visually demonstrated in a logical and comprehensive manner. The following subsections provide a general description of how the analysis results for VecTor5 models are displayed in Janus.

7.2.1 Member End Forces and Deformation Variables

As a frame analysis program, users define the frame model as a series of linear elements. At the successful resolution of each load stage, each VecTor5 member element subjected to loading exhibits a set of equal and opposite axial, shear and moment end forces as part of its analysis output. Janus provides the facilities for categorically displaying these member end forces in an analogous colour pattern to the contour mode result system utilized extensively throughout Janus. Upon selection of a member end force option via **Results** ▷ *Member Variables (V5)* ▷ *Forces*, all member elements will be coloured in a gradient corresponding to the numerical value range presented in the associated *Legend* dialog. Utilizing the same display mechanics attributed to contour mode in Janus, the *Toggle Face Fill* toolbar button may be used to toggle the result mode for member end forces on and off.

In addition to reporting member end forces, Janus also provides a similar contour mode feature for presenting general member element diagnostic variables reported as part of the global frame analysis solution. Variables include:

- Concrete axial strain at mid-depth of the cross section, ε_{cl} (ecl)
- Concrete shear strain at mid-depth of the cross section, γ_{xy} (gxy)
- Curvature, ϕ (phi)
- Maximum longitudinal reinforcement strain, $\varepsilon_{sl,max}$ (eslmax)
- Minimum longitudinal reinforcement strain, $\varepsilon_{sl,min}$ (eslmin)
- Maximum transverse reinforcement strain, $\varepsilon_{st,max}$ (estmax)
- Maximum crack width, w_{cr} (wcr)

The general-purpose *Legend* dialog will modify its entry ranges accordingly to represent the requested member element force or deformation parameter.

7.2.2 Output Member Contour Mode

Results attributed to output members provide an enhanced representation of member element behaviour, listing RC and smeared reinforcement structural response characteristics on a layered basis through the transverse depth of the member element section. In order to display section response parameters of output members in Global Model View, conventional stress-strain contour mode options in the **Results** drop-down menu for RC and reinforcement are adopted for activating a subset of contour modes specifically tailored for output member parameters. Within such contour modes, non-output member elements are presented using a black line colour. It is important to note that such contour modes are viewed on a single layer basis in Global Model View; output member elements are coloured in a solid colour corresponding to a specified layer number. By VecTor5 structure file notation, layer 1 is denoted as the topmost layer of the output member element section; subsequent layers are incrementally numbered through the depth of the section.

To modify the currently selected layer number, users must utilize the *Concrete/Reinf. Layer Select* controls provided in the VecTor5 *Legend* dialog to manipulate the selected RC or reinforcement layer for the current output member contour mode. Using the provided edit box, users may manually enter a number within the prescribed range of layers found. The layer selection is confirmed by pressing the *Update* button, or pressing return on the keyboard. Alternatively, the layer number may be incrementally increased or decreased from its current value using the + and - buttons, respectively. Refer to Figure 7.4

below for an example of an output member contour mode demonstrated on a simple VecTor5 model. Depending on the context of the selected contour mode, the layer edit box is used for the dual purpose of specifying both RC and longitudinal layer numbers. The range of applicable layer values will automatically modify based on whether selected contour mode pertains to RC or longitudinal reinforcement layers. If a valid layer number request exceeds the number of RC or longitudinal reinforcement layers that a particular section type possesses, output members corresponding to that section type will be coloured black. For further details concerning output member contour modes, refer to Subsection 7.2.3 below.

7.2.3 VecTor5 Layer View

Janus features a VecTor5-specific version of Layer View for demonstrating output member result values on a section-specific basis. Pressing the *Layer View* toolbar button, users may select an output member of interest using the provided drop-down menu list. Once an element is confirmed, Janus presents a planar section of the output member, magnified and centred in model space. RC layers are drawn to the scaled thickness and width proportions as specified in the associated VecTor5 expanded structure file. For each longitudinal steel layer, a representative solid circular shape is drawn centred at the specified depth from the top of the section. The circle is rendered with dimensions equivalent to the lumped area of steel prescribed for the layer. By default, RC layers are coloured in dark green and longitudinal reinforcement circle layers are coloured cyan.

Upon selection of a relevant output member contour mode from the **Results** menu, RC layers and

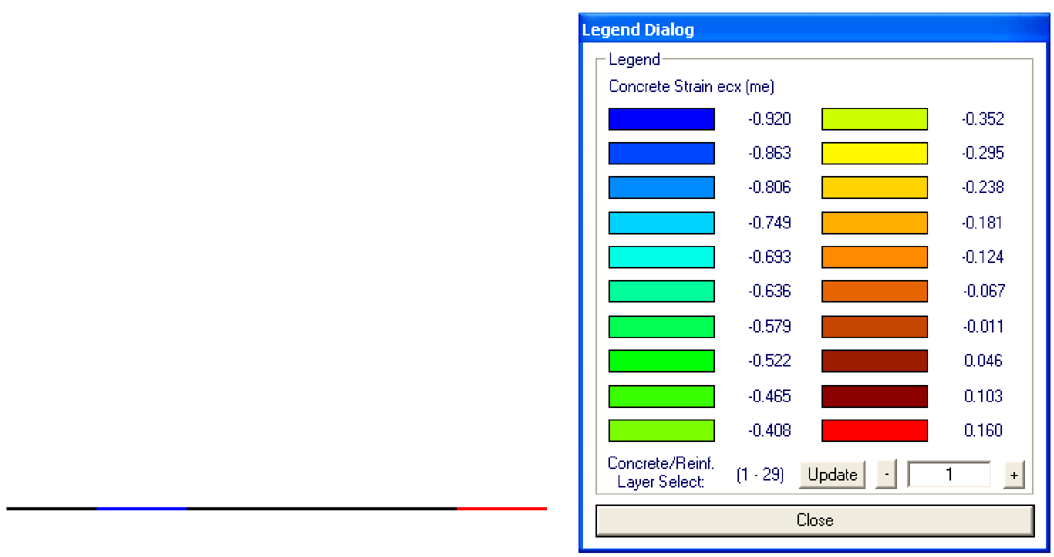


Figure 7.4: Output Member Contour Mode and Global Model View Using a Simple VecTor5 Model

longitudinal reinforcement layer circles for the section will be re-coloured in a standard whole colour pattern to reflect the number range presented in the VecTor5 *Legend* dialog. In a likewise fashion to result contour modes for other VecTor model types, selecting a variable relating to RC layers will correspondingly colour all of the RC layer polygon shapes, while longitudinal reinforcement layer circles will be coloured black. See Figure 7.5 for a demonstration of concrete strain contour mode in Layer View using a simple VecTor5 model.

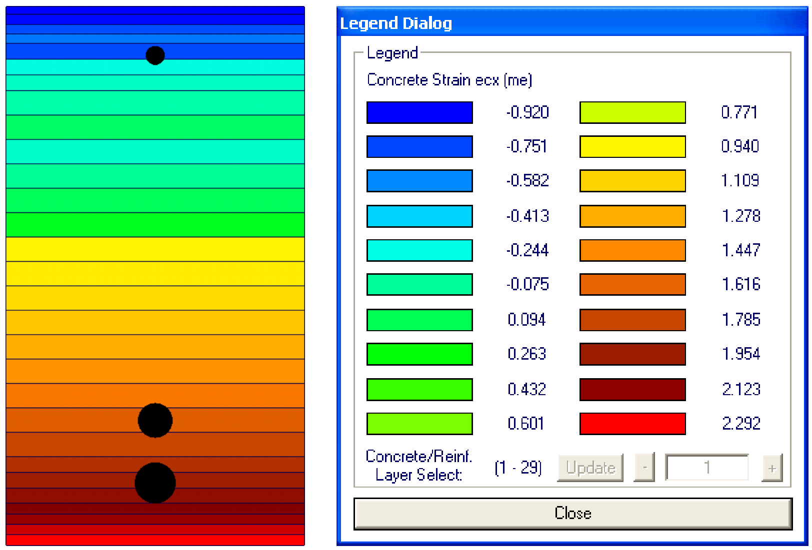


Figure 7.5: RC Layer View Using a Simple VecTor5 Model

In contrast, longitudinal reinforcement contour mode in Layer View will turn RC layers into empty wireframes, and the reinforcement circles are re-coloured to denote the corresponding contour value of the layer. Contour modes for longitudinal reinforcement variables are activated via **Results** ▷ *Reinforcement* ▷ *Direction 1* or *Direction 2*. See Figure 7.6 for a corresponding depiction of the described Layer View mode.

As previously mentioned in Subsection 7.1.2, transverse reinforcement is treated as a smeared reinforcement property in VecTor5, and is locally assigned to individual RC layers. As such, transverse reinforcement contour modes are displayed in Janus using coloured RC layer polygon shapes. VecTor5 transverse reinforcement contour modes are activated via **Results** ▷ *Reinforcement* ▷ *Direction 3*. RC layers that do not possess transverse reinforcement will be coloured black.

Hotspot mode may also be activated in VecTor5 Layer View. Utilizing the same symbolism for Hotspot mode in Global Model View, selected variables and parameters pertaining to RC layers will result in applicable RC layer faces to be either highlighted in solid red/brown colour, or wholly omitted as an empty wireframe. Hotspot mode for longitudinal reinforcement variables will highlight or omit the

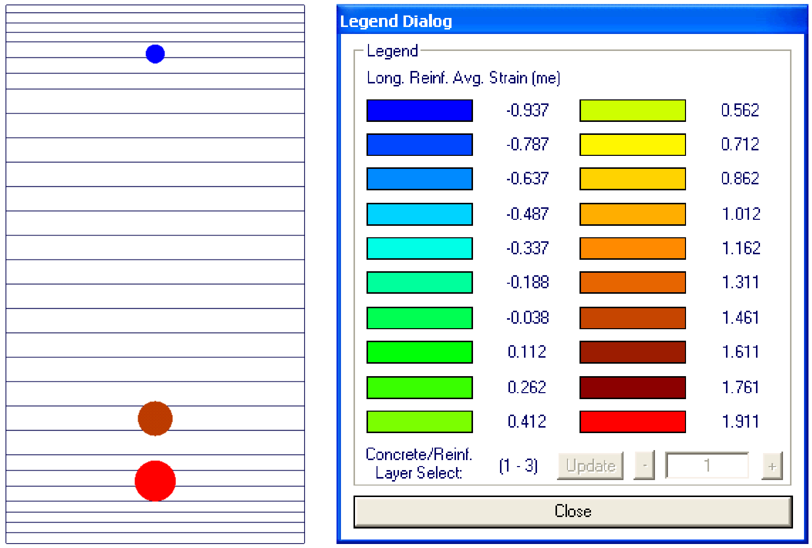


Figure 7.6: Longitudinal Reinforcement Layer View Using a Simple VecTor5 Model

representative reinforcement circles in a likewise manner, while RC layers are reduced to a wireframe.

Both Hotspot and result contour modes in VecTor5 Layer View may be toggled on and off using the *Toggle Face Fill* toolbar button. In a likewise fashion, VecTor5 Layer View may be enabled and disabled using the *Toggle 3D View* toolbar button.

Chapter 8

VecTor6 Models in Janus

The display features implemented for displaying VecTor6 models in Janus are discussed within this chapter. In particular, specific focus is provided in outlining the post-processing methodology used for presenting the geometry of VecTor6 axisymmetric models in a relevant planar manner.

8.1 Appearance

Although VecTor6 is designated as a two-dimensional finite element analysis program, one of its key defining analytical features is its capability to treat the modelled planar elements as annular shapes rotated about a common axis. In essence, the presented view of VecTor6 models in Janus is a simplified transverse cross section of the intended axisymmetric solid structure.

8.1.1 Axes

In contrast to the two-dimensional VecTor2 and VecTor5 finite element analysis programs which exclusively specify coordinates about the x-y plane, VecTor6 is unique in its use of the x-z plane as the primary modelling domain. Instead of representing actual elements, the x- and z-axis defined shapes in VecTor6 are intended to symbolize the transverse section of an annular axisymmetric solid RC structure. The z-axis is used as the common axis of rotation for all VecTor6 elements, resulting in circular cross sections in the perpendicular x-y plane which utilize the elements' distances from the z-axis as their respective radii. In order to provide visualization controls that are consistent with the original element specifications, VecTor6 models are also represented in a planar view about the x- and z-axes in Janus Global Model View. Section view about the x-y plane is used to represent the actual section of the annular shapes symbolically conveyed in VecTor6.

8.1.2 Elements

Models in VecTor6 consist of the following finite element types, with two linear degrees of freedom per node:

- three-noded triangular annular elements
- four-noded quadrilateral annular elements
- two-noded truss elements
- one-noded ring bar elements

Within Global Model View, VecTor6 annular elements are represented in Janus as two-dimensional planar shapes. RC triangular and quadrilateral elements are displayed as geometric polygons with coloured faces and black linear edges. By default, solid elements are coloured dark green. In terms of reinforcement, VecTor6 utilizes a combination of truss elements and ring bar elements to respectively represent in-plane and out-of-plane reinforcement. VecTor6 truss elements are displayed using typical conventions in Janus, with thick-width lines drawn between truss element nodes. As symbolic out-of-plane hoop reinforcement, ring bar elements are rendered as solid circular shapes to indicate reinforcement bar cross sections. By default, both truss and ring bar element shapes are coloured in cyan in Janus. The simple VecTor6 model represented in Figure 8.1 exemplifies a ring-beam with a rectangular cross section, with four ring bar elements representing hoop reinforcement.

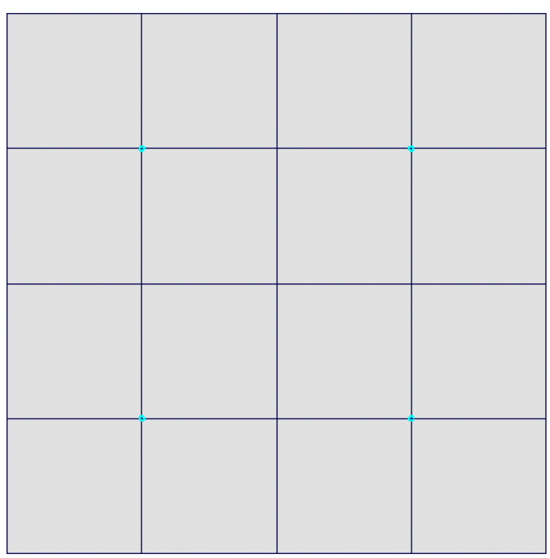


Figure 8.1: Simple VecTor6 Ring-Beam Model

8.1.3 Restraint Symbols

As a 2D analysis program, VecTor6 is capable of accounting for restrained orthogonal displacements in the in-plane x- and z-directions. In a typical fashion for displaying nodes with up to two restrained linear degrees of freedom, Janus utilizes a combination of planar pin and roller symbols to represent nodal restraints in VecTor6 models. Refer to Subsection 4.1.3 for an explanation of the planar roller and pin symbolism utilized in Janus. See Figure 8.2 for an example of a simple VecTor6 model with an edge fully supported in the vertical z-direction.

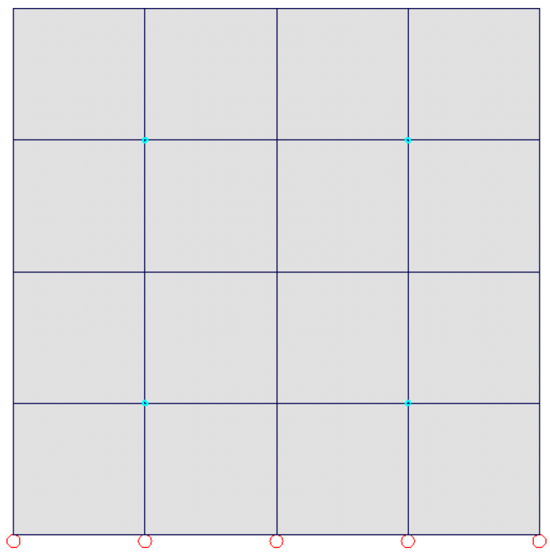


Figure 8.2: VecTor6 Restraints

8.1.4 Load and Displacement Symbols

Janus displays VecTor6 nodal loads and displacements as a series of single-headed 2D arrows rendered on the x-z plane. Load arrow symbols are oriented according to the numerical sign and direction of loading. For optimal visibility within the planar view of the model in Janus, load arrows may be interchangeably rendered with either head or tail end abutting the assigned node. Figure 8.3 below provides an illustrated depiction of nodal loads applied to a simple VecTor6 model.

8.2 Unique Features

For the purpose of displaying axisymmetric VecTor6 models in planar form, several post-processing features in Janus are adjusted to provide display model and analysis parameters in a relevant geometric context.

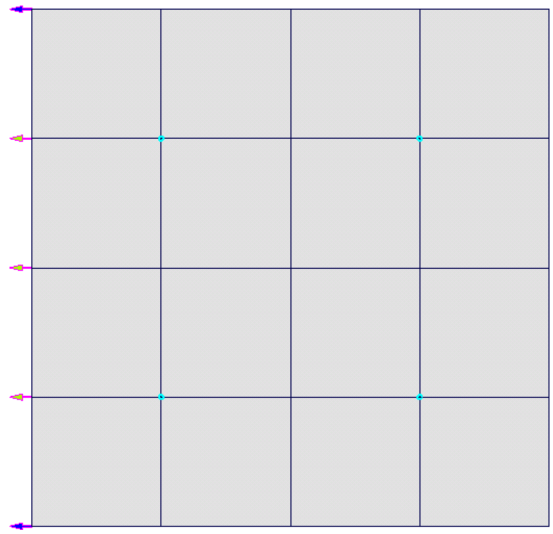


Figure 8.3: VecTor6 Load Arrows

8.2.1 Deformations

VecTor6 models in Janus are capable of exhibiting in-plane nodal displacements in the x- and/or z-directions. Upon activation of the Deformations mode, VecTor6 model elements are re-rendered with the prescribed nodal displacements as reported in the VecTor6 expanded analysis output file(s). As with other VecTor programs, linear nodal displacements for VecTor6 models in Janus are uniformly scaled by a numerical deformation scale factor. By default, the deformation scale factor is set to 30.

8.2.2 Crack Patterns

In a likewise manner to how cracks are presented within Crack Pattern mode for VecTor2, Janus projects VecTor6 crack patterns directly upon the planar faces of the annular 2D triangular and quadrilateral elements in Global Model View. See Subsection 3.8.2 for a detailed description of how crack patterns are visualized for VecTor6 models in Janus.

8.2.3 XY Section View

Due to the fact that VecTor6 models are only a representative transverse section of an entire axisymmetric RC structure, *Section XY* is the only Section View option which is wholly applicable for displaying VecTor6 models in Janus. The VecTor6-specific *XY Section* section view also provides a secondary function by visualizing the annular sectional shape which is representative the simulated axisymmetric structure. The RC element vertices which intersect or lie on the specified section z-coordinate are radially extruded about the x-y plane as circular shapes, utilizing their respective x-axis width and absolute orthogonal x-direction distance from the z-axis as the resulting radii. Truss elements, previously displayed in-plane in the Global Model View x- and z-axis orientation, are perpendicularly presented as single out-of-plane reinforcement bar truss sections located on the x-axis. If the input section z-coordinate coincides with the z-coordinate of a ring bar element, the ring bar element - previously represented as an out-of-plane reinforcement bar - is displayed in-plane with the section as a continuous circular reinforcement hoops. See Figure 8.4 below for a representation of the annular shapes resulting from an x-y plane section of a simple VecTor6 model, with magnified view of the intersected ring bar elements portrayed as hoop reinforcement.

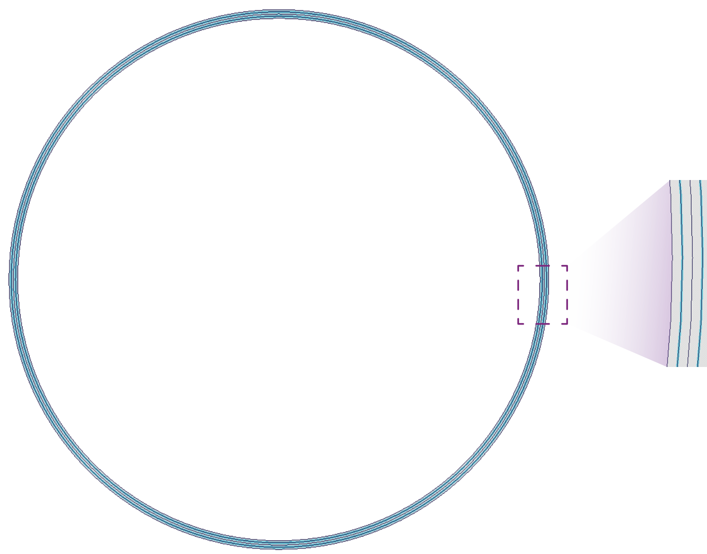


Figure 8.4: XY Section View of a Simple VecTor6 Ring-Beam Model

8.2.3.1 Sectional Deformations

In a consistent fashion with section view functionality for other VecTor programs, *XY Section* for VecTor6 may be activated in conjunction with Deformations Mode in Janus. However, instead of nodal displacements being directly represented through a deformed planar section, displacements in the x-direction are expressed as scaled radial dilations and/or contractions of the sectional circular node and/or element shapes. Accordingly, nodal displacements in the z-direction are reflected in the resulting position of the x-y plane section.

8.2.3.2 Sectional Result Modes

With the exception of node-related displacement and reaction contour modes, relevant RC and reinforcement-related contour mode options found in the *Results* menu may be activated within *XY Section* section view for VecTor6 models. Annular RC element shapes and/or reinforcement symbols will be re-coloured in a manner consistent with the applicable contour or Hotspot mode. For example, RC element-associated contour modes result in coloured RC annular shapes, with truss and ring bar elements coloured black. Reinforcement-related contour modes will result in the RC elements drawn as wireframe circular shapes, and reinforcement elements highlighted to reflect the numerical range presented in the *Legend* dialog that appears. Likewise, activating Hotspot mode within *XY Section* will predictably highlight relevant annular shapes in the typical red or brown ranges, and otherwise omit non-applicable ones as a wireframe.

Chapter 9

Using Janus

9.1 Introduction

The following sections provides an overview of facilities in Janus on an instructional basis. As previously mentioned, many of the post-processing features in Janus are dependent on the context of the VecTor model type being viewed; both the availability and functionality of Janus menu and toolbar controls will vary in order to present the finite element model of interest in a relevant and logical manner. Accordingly, subsequent sections within this chapter are organized by VecTor program in order to provide information in a succinct and applicable manner to the user.

In general, each of the forthcoming VecTor program-specific sections are organized in the following manner:

- An introductory section for preparing and opening data files specifically pertaining to the VecTor model type
- An overview of available view and mode options and their interactive relationships
- Demonstration of visualization features through the use of a relevant example model

9.2 Opening VecTor Job Files

As the first file initially read by Janus, information contained within the job file is vital for subsequent file reading procedures in Janus. Inputting data for these entries may be completed manually using a standard text editor, or through automation using the FormWorks pre-processor program. For visual reference to job file components described within this section, see Figure 9.1 below.

```
                                 * * * * * * * * * * * *
                                 *     V E C T O R     *
                                 *   J O B   D A T A   *
                                 * * * * * * * * * * * *

Job Title        (30 char. max.)         : Shear Wall SW22
Job File Name    ( 8 char. max.)         : SW22
Date             (30 char. max.)         : May 2011

STRUCTURE DATA
--------------
Structure Type                           : 2
File Name        ( 8 char. max.)         : SW22

LOADING DATA
------------
No. of Load Stages                       : 41
Starting Load Stage No.                  : 1
Load Series ID   ( 5 char. max.)         : SW22

Load    File Name                           Factors                                  Initial
Case    (8 char. max.)      Initial    Final    LS-Inc   Type  Reps    C-Inc      Load Stage
  1     SW22DL              0.0000   20.0000   0.5000    1     1     0.0000          1
  2     SW22VL              1.8200    1.8200   0.0000    1     1     0.0000          1
  3     NULL                0.0000    0.0000   0.0000    1     1     0.0000          1
  4     NULL                0.0000    0.0000   0.0000    1     1     0.0000          1
  5     NULL                0.0000    0.0000   0.0000    1     1     0.0000          1
```

VecTor model type

Expanded structure file name

Load stage properties

Expanded analysis output file name

Expanded load file name(s)

Figure 9.1: Sample Job File

9.2.1 Structure Data

Within the universal "STRUCTURE DATA" heading of each job file, users provide values describing the VecTor model type as well as the user-chosen structure file name. In order to fundamentally communicate the VecTor model type to Janus, an integer value between 2 to 6 (denoting VecTor2 to VecTor6, respectively) is expected at the entry "STRUCTURE DATA" ▷ "Structure Type". In addition to being the sole identifying variable for subsequently determining VecTor-specific functionality within Janus, the entered 'Structure Type" variable also establishes which file types are searched for when initially opening expanded structure, load and analysis output files. For example, the numerical "Structure Type" value of 2 corresponds with VecTor2 models; thus, Janus will only search the immediate file directory for VecTor2-associated file types (".S2E", ".L2E", and ".A2E").

Using the established structure file type and the string value retrieved from "STRUCTURE DATA" ▷ "File Name (8 char. max.)", Janus will seek to locate and read the contents of the structure file associated with the current VecTor job. At a minimum, only the job file and associated expanded structure file are required to successfully open and display a VecTor model. Janus functions for reading loading data and analysis output data will produce relevant warning messages and disable certain post-processing features if any expected files are invalid or missing, but it is essential that the expanded structure file is located and the model data is read correctly in order for the VecTor model to open without fault.

9.2.2 Loading Data

Within the job file, several essential variables are also listed under the "LOADING DATA" heading. The "Load Case" table provides entries for activating up to five load cases. If the load case file name entry is not "NULL", the load case is considered to be active. Janus will search the working directory for a file of the exact same name, ending with the expanded load file extension associated with the opened VecTor model type. If at least one expected load file is unsuccessfully found or the contents are presented in an incompatible or incomplete format, relevant disclaimer messages will appear in order to inform the user that not all of the load information will be available for visual display and interpretation.

9.2.3 Analysis Output Data

Following the procedure for reading expanded load files, Janus executes a function for reading VecTor-specific expanded analysis output files. General data associated with finding and reading analysis output files is also encapsulated under the "LOADING DATA" heading in the job file. Firstly, the integer variables corresponding to "No. of Load Stages" and "Starting Load Stage No." are used to establish the total range of load stages which may be specified for opening. According to the sample job file previewed in Figure 9.1, Janus would initialize expecting a maximum of 41 load stages, starting from load stage 1. Unless the VecTor-specific memory factor is exceeded and the user is prompted to specify a custom subset of the available load stage range using the *Load Stage Range* dialog (refer to Section 3.4), Janus proceeds to search for file names matching the entered five character string corresponding to "LOADING DATA" ▷ "Load Series ID (5 char. max.)", appended with an underscore ("_") and serialized with the applicable load stage number. Referring again to Figure 9.1, Janus would begin its reading function with a search for a file named "SW22_01.A2E".

Upon successfully reading an expanded analysis output file, Janus will increment the load stage number and search for the next file. If an expected output file is invalid or missing, Janus will terminate the file reading loop and present the result data currently available for viewing. All load stage-related interface features and variables in Janus will be accordingly modified for the limited range of analysis output files encountered during file reading. As well, disclaimer messages will appear to inform the user that not all expected analysis output files could be found or correctly loaded as requested.

9.3 Viewing VecTor2 Models

9.3.1 Opening a VecTor2 Model

As with any VecTor model, the file opening procedure for viewing VecTor2 models in Janus begins with opening a VecTor2 job file. Since all VecTor job files utilize a common ".JOB" file extension, it is imperative that VecTor2 models are explicitly specified within the job file by declaring an integer value of "2" under the entry prompt for "STRUCTURE DATA" ▹ "Structure Type". Upon recognizing that a VecTor2 model type is being opened, Janus searches the immediate file directory for files corresponding to the file naming convention listed in the job file and matching VecTor2-specific file extensions. The specified "No. of Load Stages" and "Starting Load Stage No." parameters serve as default values for the specified load stage range used when reading expanded analysis output files. See Table 9.1 for an overview of VecTor2 file types and corresponding naming entries.

9.3.2 Feature Overview

As previously mentioned in Section 3.8, a variety of view and mode options may be generally employed in combination for improved versatility in post-processing purposes. However, in the context of viewing VecTor2 model types, several specific Janus visualization features may not be applicable or available for implementation. Table 9.2 below provides a tabular overview of the general post-processing features available for viewing in Janus, as well as the interactive relationships between different view and mode options. A check mark (✓) signifies that the mode and/or views may be operated in tandem, while a cross (✗) denotes that only one of the two features may be enabled at one time.

File Type	**File Extension**	**File Name Entry**
Expanded Structure File	S2E	"STRUCTURE DATA" ▹ "File Name (8 char. max.)"
Expanded Load File	L2E	"LOADING DATA" ▹ "Load Case" (1-5) ▹ "File Name (8 char. max.)"
Expanded Analysis Output File	A2E	"LOADING DATA" ▹ "Load Series ID (5 char. max.)" note: each expanded analysis output file name is also serialized using the following format: "(Load Series ID)_i.A2E", where i is the load stage number.

Table 9.1: VecTor2 File Types

	Load Cases	Restraints	Materials	Deformations	Crack Pattern	Nodal Contour	RC Element Contour	Truss Contour	Bond Contour	Hotspot	XY Section	YZ Section	XZ Section
Load Cases													
Restraints	✗												
Materials	✓	✓											
Deformations	✓	✓	✓										
Crack Pattern	✓	✓	✓	✓									
Nodal Contour	✓	✓	✗	✓	✗								
RC Element Contour	✓	✓	✗	✓	✓	✗							
Truss Contour	✓	✓	✗	✓	✗	✗	✗						
Bond Contour	✓	✓	✗	✓	✗	✗	✗	✗					
Hotspot	✗	✗	✗	✓	✗	✗	✗	✗	✗				
XY Section	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓			
YZ Section	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✗		
XZ Section	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✗	✗	

Table 9.2: Janus Feature Overview for VecTor2 Models

9.3.3 Example Model

The following subsection provides a brief presentation of the available visualization features in Janus available for VecTor2, using a previously developed example model. The finite element model selected for VecTor2 post-processing demonstrations is entitled Shear Wall SW22, developed by Vecchio based on experimental specimens previously designed and tested by Lefas et al. (1990). Due to the planar nature of the shear wall and applied loading, VecTor2 is deemed to be an appropriate analysis program for producing an analogous finite element membrane structure.

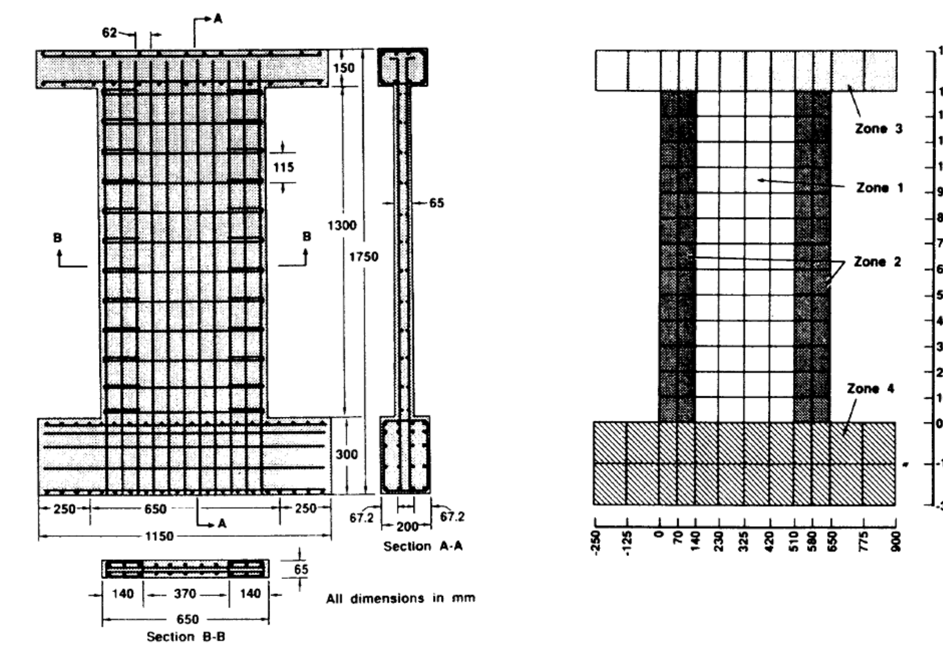
9.3.3.1 Description

SW22 is an experimental reinforced concrete shear wall specimen originally designed and tested by Lefas et al. (1990). In terms of gross dimensions, the wall consisted of a regular rectangular cross section with a 65 mm thickness and width of 650 mm, and exhibited an overall height to width ratio of 2:1. For the purpose of load transfer, top and bottom loading beams of 200 mm thickness were monolithically cast with the wall. In order to simulate a rigid ground foundation, the bottom base beam was heavily reinforced and anchored to the testing floor. See Figure 9.2 a) for the structural details of experimental specimen SW22. For illustration purposes, the equivalent VecTor2 finite element model of SW22 is comparatively shown in Figure 9.2 b). In contrast to the original model specifications of 140 rectangular elements displayed in Figure 9.2 b), the SW22 VecTor2 example model presented in this manual utilizes a finer model mesh consisting of 496 rectangular elements. However, the gross specimen dimensions and material specifications are congruent to the model implemented in the original analysis.

9.3.3.2 Structure

As specified by Lefas et al., the reinforced concrete shear wall specimen SW22 consisted of 45 MPa concrete, and contained an arrangement of vertical and horizontal reinforcement with additional horizontal steel ties specified at the wall edges. The vertical reinforcement was designated as two layers of 8 mm diameter deformed bars at 62 mm spacing, while the horizontal reinforcement were sets of 6.25 mm diameter bars spaced at 115 mm through the height of the wall. Lastly, the closed ties at the wall edges were formed from 4 mm diameter mild-steel bars, establishing internal reinforcement cages of 140 mm width at each edge of the wall (Vecchio, 1992).

Based on the experimental concrete and steel reinforcement specifications of SW22, the resulting VecTor2 model utilizes RC membrane elements with smeared reinforcement components to represent the



a) Shear Wall Specimen Structural Details (adapted from Lefas et al., 1990)

b) Finite Element Model Details (adapted from Vecchio, 1992)

Figure 9.2: SW22 Experimental Specimen and Equivalent Finite Element Model Details

concrete shear wall integrated with each distinct type/orientation of steel reinforcement. For simplification of the expressed amounts of steel reinforcement and ease of conversion into smeared reinforcement properties, quantities are presented as equivalent percentages of the gross concrete area in a defined region and/or orientation. For finite element modelling purposes, three distinct RC material regions are identified: the top and bottom loading beams, the shear wall edges (due to the presence of closed ties), and central shear wall web. See Table 9.3 below for an overview of the material specifications used in the VecTor2 example model, and Figure 9.3 for the corresponding material colours displayed in Janus. The model materials may be viewed by enabling the menu option **Structure** ▷ *Materials*.

Alternatively, the material properties of individual RC membrane elements may be recalled using the *Element Attributes* dialog as previously described in Section 3.10. In addition to providing material property characteristics, the *Element Attributes* dialog also presents the load stage-specific stress-strain state of the selected element.

Material Type		Concrete Properties						Steel Properties						
No.	Colour	f'_c (MPa)	f'_t (MPa)	E_c (MPa)	ε_0 (me)	Thickness (mm)	Other	Orientation[3] (deg)	ρ (%)	f_y (Mpa)	f_u (MPa)	E_s (MPa)	ε_{sh} (me)	ε_u (me)
1		43	2.16	32,800	2.0	65	default	0	0.82	520	650	200,000	10	75
								90	2.09	470	650	200,000	10	100
2		43	2.16	32,800	2.0	65	default	0	0.82	520	650	200,000	10	75
								0	0.82	520	650	200,000	10	125
								90	3.312	470	650	200,000	10	100
								361[4]	0.9	420	650	200,000	10	125
3		43	2.16	32,800	2.0	200	default	0	2.5	420	650	200,000	10	125
								90	2.5	420	650	200,000	10	125
								361	2.5	420	650	200,000	10	125

[3] with respect to the x-axis [4] 361 = out-of-plane reinforcement

Table 9.3: VecTor2 Example Model Material Specifications

9.3.3.3 Restraints

As previously mentioned, the reinforced bottom base beam for experimental specimen SW22 was noted to be fixed to the laboratory floor during testing. Accordingly, the VecTor2 finite element model assigned nodal restraints along the entire bottom edge of the bottom beam in both the x- and y-directions. In Janus, restraining both the x- and y-direction degrees of freedom results in a pin restraint symbol, as demonstrated in Figure 9.4 below. The view of restraints may be activated using the menu option **Structure** ▷ *Restraints*.

9.3.3.4 Loads

In addition to a horizontal shear load incrementally applied to failure, the loading protocol for the original SW22 test specimen also prescribed a constant axial compressive load to simulate representative multi-storey building loads bearing down upon the shear wall. In an analogous manner to the loads applied during the experimental test procedure, equivalent nodal loads are applied to the SW22 finite element model through the top loading beam as a combination of two active load cases.

The first load case, "SW22DL", is designated as the shear load-inducing load case, applying a lateral displacement in the positive x-direction to a central top beam node. The lateral displacement in Load Case 1 is monotonically applied to a single node at the top loading beam in 0.5 mm increments, up to a maximum of 20 mm. Load Case 2, denoted in the SW22 job file as "SW22VL", consists of a constant 182 kN vertical axial load applied in the negative y-direction to the shear wall. In order to avoid localized stress concentrations, the constant axial load is symmetrically distributed among 13 central nodes spread

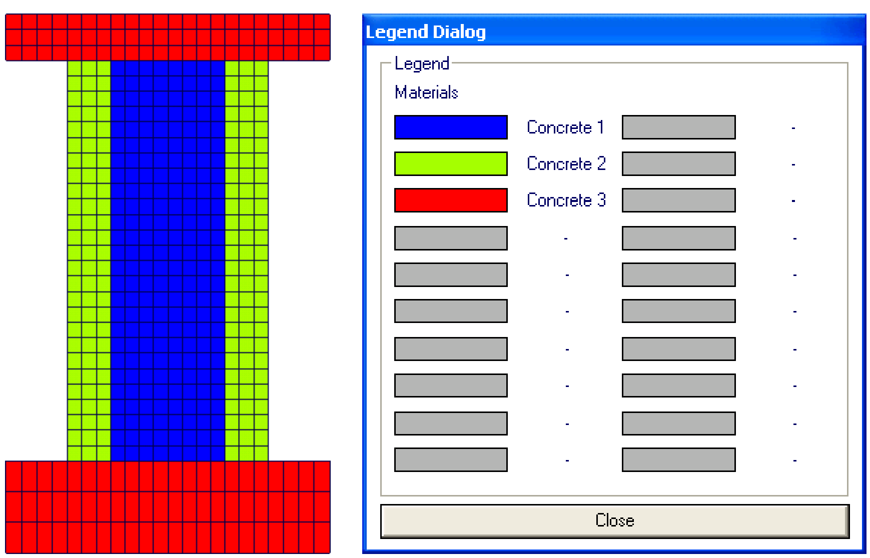


Figure 9.3: VecTor2 Example Model in Material Mode and Legend Dialog

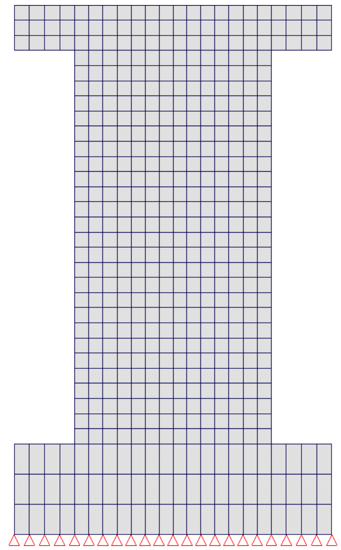


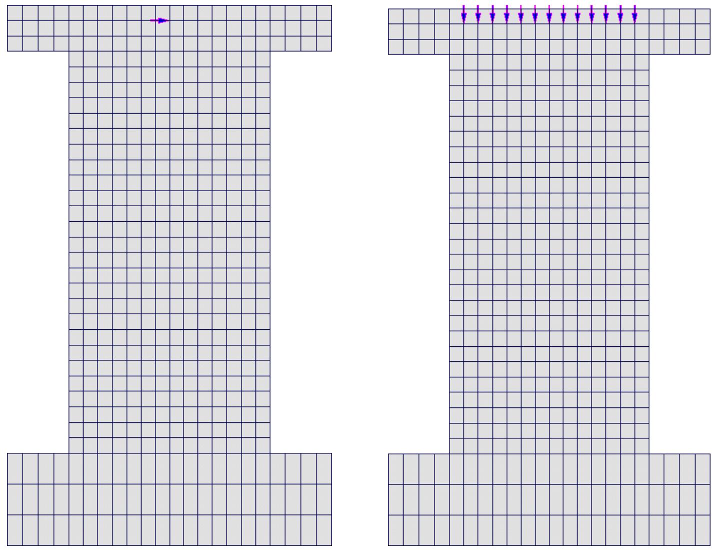
Figure 9.4: VecTor2 Example Model Restraints

across the top loading beam.

Observing the load increment, initial and final load factor values for Load Case 1, 41 load stages are required in total to monotonically increase the applied horizontal displacement from 0 mm to 20 mm in 0.5 mm load stage increments. Since Load Case 2 represents a constant vertical axial load, the characteristics of Load Case 1 are the determining factor for the total number of load stages prescribed in the VecTor2 analysis of SW22. For a complete description of various load stage-related specifications for monotonic as well as cyclic and reverse-cyclic load types, refer to the VecTor2 and FormWorks software literature (Wong et al., 2012). See Figure 9.5 a) and b) for illustrated load case arrows of Load Cases 1 and 2 as depicted by Janus. Each active load case may be viewed by selecting the applicable load case option via **Structure** ▷ *Loads*.

9.3.3.5 Results

A demonstration of VecTor2 post-processing features in Janus is provided in the following subsection, using the output data produced by VecTor2 as a result of the SW22 shear wall finite element model analysis. All analysis results are viewed on a basis of the currently selected load stage. In order to view different load stages, the user must modify the current load stage selection. As previously described in Section 3.3, the load stage may be incrementally increased or decreased by clicking on the appropriate buttons in the *Load Stage Controls* area of the *Navigation* dialog, or pressing the page up and page down keys. Users may specify an exact load stage number by entering a valid integer value into the *Current* edit box in the *Load Stage Controls* area of the *Navigation* dialog. The typical file size of SW22



a) Load Case 1 - Monotonically Increasing Lateral Displacement

b) Load Case 2 - Constant Vertical Axial Load

Figure 9.5: VecTor2 Example Model Load Cases

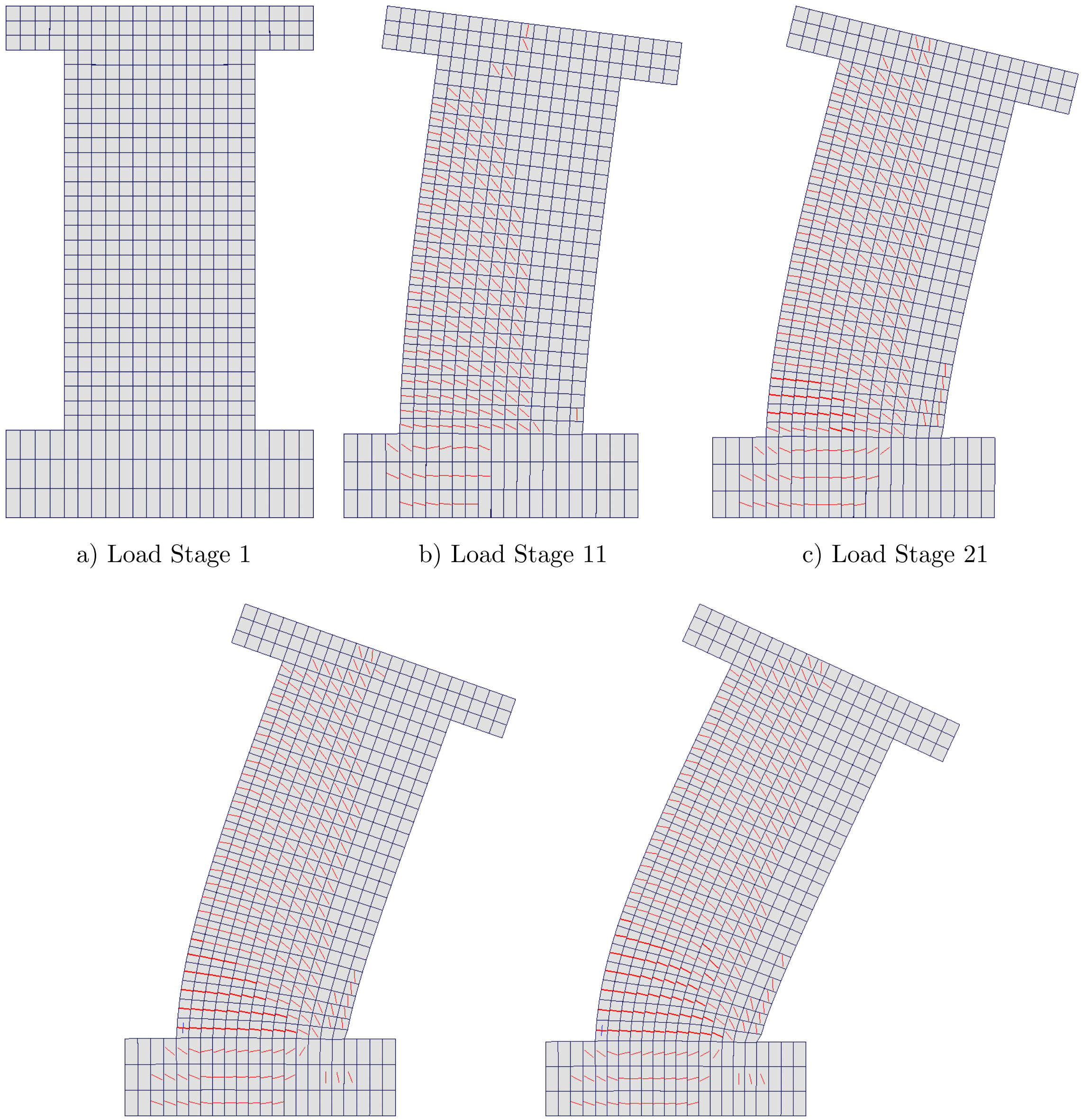
expanded analysis output files is approximately 400 kB.

Progressive model deformations and crack patterns may be observed through load stages by concurrently activating the menu options **Results** ▷ *Deformations* and **Results** ▷ *Crack Pattern*. Refer to Figure 9.6 below for the resulting deformations and crack patterns of the VecTor2 SW22 model from load stages 1 through 41 in load stage increments of 10.

Additionally, it may be of interest to the user to examine the overall load-deformation response of SW22. The *Data Platform* dialog may be used to extract the necessary numerical data from the expanded analysis output files. For instance, the total load versus displacement curve for a central mid-width node at the top of the shear wall, specified in the model as Node 479, provides an indication of the overall structural response of the experimental specimen as the prescribed lateral displacement is incrementally applied.

The load and displacement variables may be specified in the general variable selection dialog by selecting "Total X Restraint Force" and "dx - Displacement x" in two separate variable input columns. Once the variables are confirmed, the two entry columns are respectively identified as "Force X" and "dx" in the *Data Platform* dialog (see Figure 9.7 below). The two entry columns must both be activated by selecting the appropriate boxes corresponding to the "Check box to include data" label.

Using the *Create Excel File* option in the *Data Platform* dialog, data are exported as an external file in a column-separated format for facilitating subsequent data manipulation and visualization needs.



a) Load Stage 1 b) Load Stage 11 c) Load Stage 21

d) Load Stage 31 e) Load Stage 41

Figure 9.6: VecTor2 Example Model Deformations and Crack Pattern

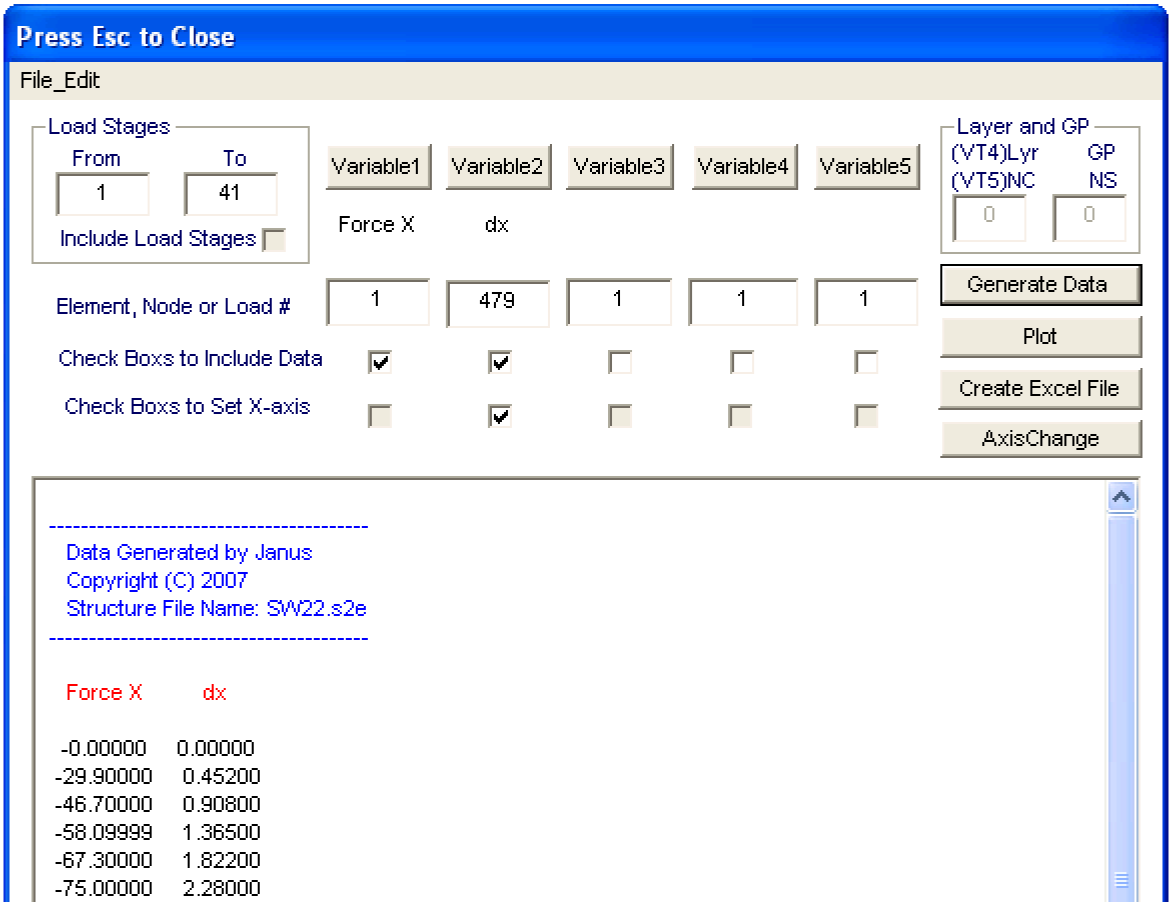


Figure 9.7: VecTor2 Example Model Data Platform Dialog

The resulting load-deformation response of the VecTor2 SW22 model is displayed in Figure 9.8 below.

Since the VecTor2 SW22 model consists entirely of RC membrane elements integrated with smeared reinforcement components, applicable contour mode variables may be selected for post-processing purposes. Stress and strain variables may be invoked using **Results** ▷ *Total/Concrete Stresses*, and **Results** ▷ *Total/Concrete Strains*, while the stress-strain contour modes for different smeared reinforcement directions may be selected using applicable options found in **Results** ▷ *Reinforcement*. See Figure 9.9 for a sample presentation of the contour mode for the distribution of RC element stresses in the y-direction.

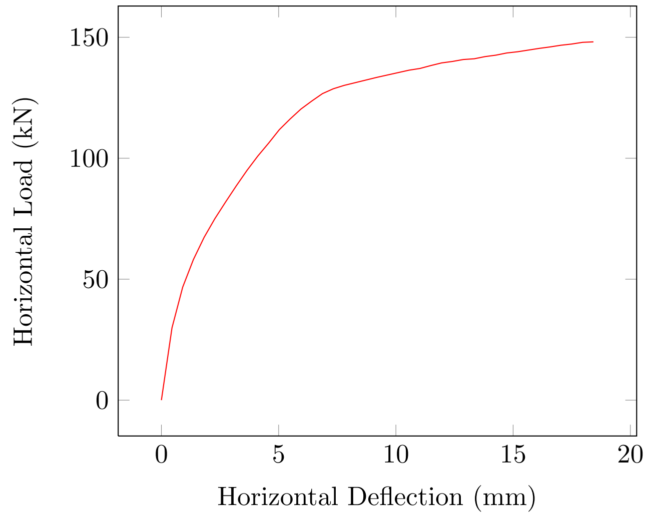


Figure 9.8: VecTor2 Example Model Load-Deformation Response

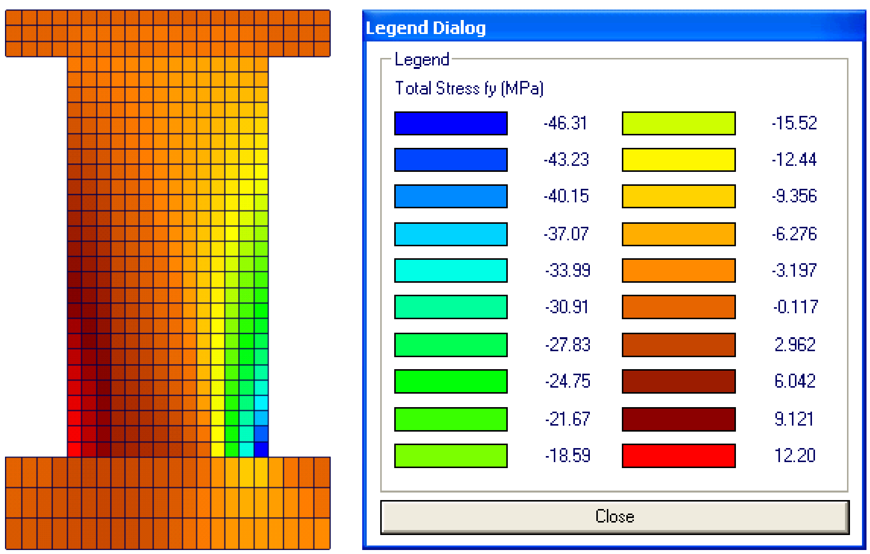


Figure 9.9: VecTor2 Example Model in RC Element Contour Mode

9.4 Viewing VecTor3 Models

9.4.1 Opening a VecTor3 Model

The procedure for opening VecTor3 models in Janus is performed in a congruent manner to opening any other VecTor job. Within the job file, VecTor3 models are explicitly declared using an integer value of "3" in the entry prompt for "STRUCTURE DATA" ▷ "Structure Type". Once Janus recognizes that a VecTor3 job file is being opened, the post-processor program will search the immediate file directory for matching file names and file types corresponding to entry strings in the job file and matching VecTor3-specific file extensions. The universally specified "No. of Load Stages" and "Starting Load Stage No."

parameters are used within the Janus file reading protocol to establish the default load stage range for successively searching and reading expanded analysis output files. See Table 9.4 for an overview of VecTor3 file types and corresponding naming entries.

9.4.2 Feature Overview

As previously described in Chapter 5, VecTor3 is a finite element analysis program developed for the purpose of analyzing RC solid elements in 3D space. Amongst the five VecTor programs, VecTor3 possesses one of the most diverse combinations of element geometry and analysis result conditions from a post-processing visualization perspective. The unique demands of rendering VecTor3 elements and displaying results in 3D space are highly influential in defining the fundamental design of various mode and view features in Janus; much of the display functionality implemented for other VecTor model types is derived from features originally developed for VecTor3.

A variety of view and mode options may be activated in tandem for improved versatility in the post-processing of VecTor3 models. However, based on the available analysis result parameters and contextual display mechanics for VecTor3 models, several specific Janus visualization features may not be compatible for combined use with others. A tabular overview of available VecTor3 post-processing features is provided in Table 9.5 below, with check marks and crosses denoting the interactivity between different view and mode options. Check marks (✓) between two functions signify that the two modes and/or views may be simultaneously enabled, while crosses (✗) denotes that selecting one mode will alternatively toggle off the other.

File Type	File Extension	File Name Entry
Expanded Structure File	S3E	"STRUCTURE DATA" ▷ "File Name (8 char. max.)"
Expanded Load File	L3E	"LOADING DATA" ▷ "Load Case" (1-5) ▷ "File Name (8 char. max.)"
Expanded Analysis Output File	A3E	"LOADING DATA" ▷ "Load Series ID (5 char. max.)" note: each expanded analysis output file name is also serialized using the following format: "(Load Series ID)_i.A3E", where i is the load stage number.

Table 9.4: VecTor3 File Types

	Load Cases	Restraints	Materials	Deformations	Crack Pattern	Nodal Contour	RC Element Contour	Truss Contour	Bond Contour	Hotspot	XY Section	YZ Section	XZ Section
Load Cases													
Restraints	✗												
Materials	✓	✓											
Deformations	✓	✓	✓										
Crack Pattern	✓	✓	✓	✓									
Nodal Contour	✓	✓	✗	✓	✗								
RC Element Contour	✓	✓	✗	✓	✓	✗							
Truss Contour	✓	✓	✗	✓	✗	✗	✗						
Bond Contour	✓	✓	✗	✓	✗	✗	✗	✗					
Hotspot	✗	✗	✗	✓	✗	✗	✗	✗	✗				
XY Section	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓			
YZ Section	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗		
XZ Section	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗	

Table 9.5: Janus Feature Overview for VecTor3 Models

9.4.3 Example Model

The following subsection presents a selection of the post-processing display capabilities in Janus available for VecTor3 models. A VecTor3 model previously developed by the author will be used for demonstration purposes. The finite element model, designated as Wind Turbine Foundation KWF, is a wind turbine foundation subjected to combined axial, shear and moment loads applied to a central loading pedestal. The design intent of KWF is to corroborate and supplement prior results obtained from a previous 2D VecTor2 analysis of the same wind turbine foundation; the VecTor3 version of the model actively accounts for out-of-plane structural mechanisms not fully captured in VecTor2 due to the inherently planar nature of 2D analyses.

9.4.3.1 Description

KWF is a 3D finite element model of a reinforced concrete wind turbine foundation created using VecTor3 analysis software. The wind turbine foundation structure is a spread footing foundation, consisting of a regular octagonal base pad which tapers upward to a truncated square apex topped with a circular loading pedestal and steel tower flange. The extreme outer dimensions of the foundation are 19 m by

19 m in plan, with the central widths measuring 7.88 m on all sides. The octagonal base pad has a minimum height of 0.4 m at the outer edges, and increases linearly to a maximum of 2.0 m at the top surface of the pad. The central circular loading pedestal is approximately 5.5 m in diameter, extending 1.1 m in height from the top surface of the base pad. See Figure 9.10 for an isometric representation of wind turbine foundation structure dimensions.

Based on the symmetry of loading and the physical dimensions of the model, the VecTor3 model KWF is designed as an equivalent half-model with appropriately modified restraint, material, and load conditions along the established plane of symmetry. The half-model is composed of approximately 2448 regular hexahedral elements and 533 truss elements. Due to limitations in the available element geometry for VecTor3, the circular loading pedestal is represented using a loading pedestal with an equivalent rectangular cross section. Similarly, the circular steel tower flange is substituted with an analogous steel loading plate. Aside from 13 truss elements used to represent discrete vertical reinforcement bars and post-tensioned anchor bolts cast within the loading pedestal, the majority of truss elements established in KWF are used as compression-only reinforcement elements to analogously represent the approximate stiffness of the supporting soil substrate beneath the foundation. See Figure 9.11 for an isometric conceptual illustration of the wind turbine foundation half-model and the conceived VecTor3 finite element model.

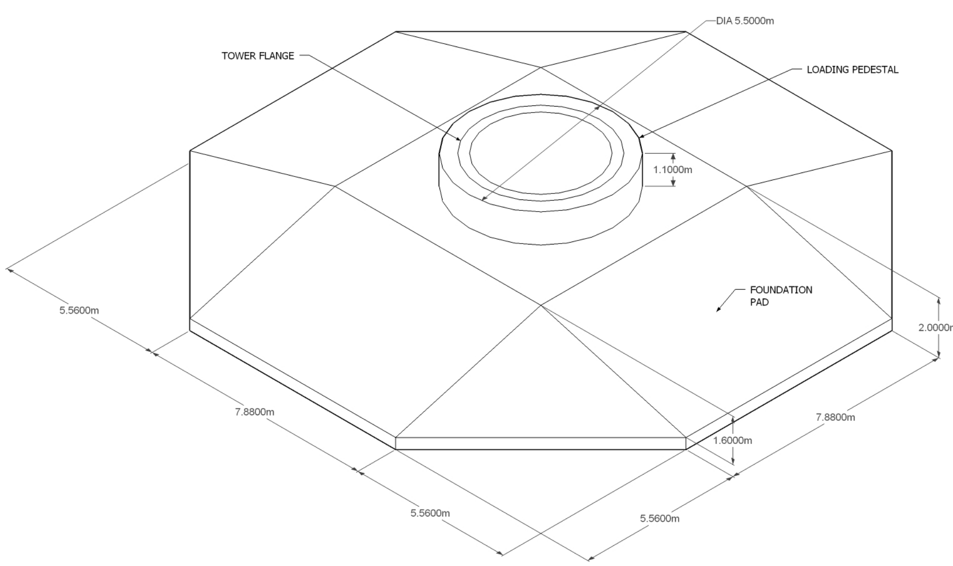
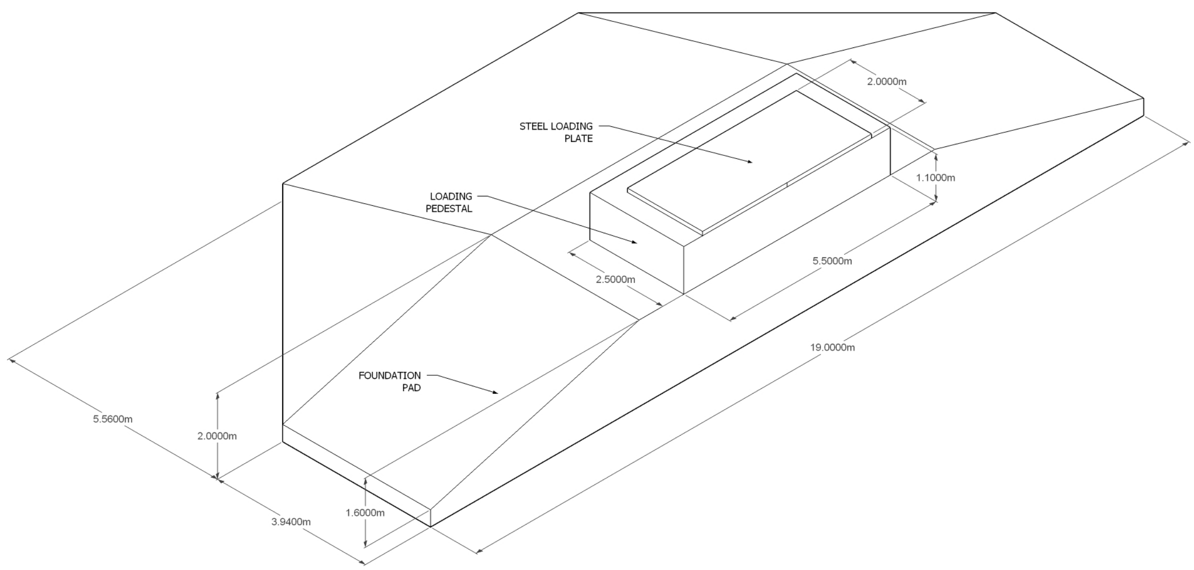
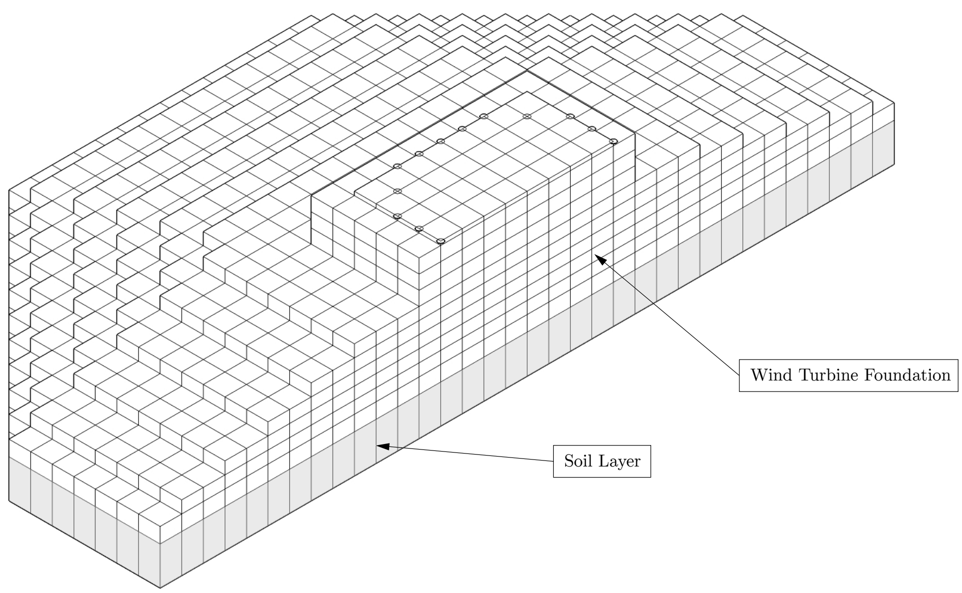


Figure 9.10: KWF Wind Turbine Foundation Dimensions



a) Wind Turbine Foundation Conceptualized Half-Model



b) Finite Element Model Details

Figure 9.11: KWF Finite Element Model

9.4.3.2 Structure

As specified in the structural detailing, the wind turbine foundation utilizes 30 MPa concrete, with layers of longitudinal reinforcement placed in both directions along the top and bottom surfaces of the foundation base pad. At the top reinforcement mat, reinforcement bars are angled to conform to the sloping geometry of the pad as the thickness tapers linearly downward away from the truncated square apex. The top mat of longitudinal reinforcement is composed of #25M bars spaced at 200 mm in both directions. The bottom mat of reinforcement consists of #20M bars spaced at 200 mm, with an additional reinforcement layer of #30M bars placed within a central 12 m by 12 m square region. The loading pedestal vertical reinforcement consists of 80-#25M 'Z' bars radially spaced about the perimeter, with a gross steel area of 40,000 mm^2. Additionally, post-tensioned 160 1-3/8" diameter steel bolts are arranged about the load pedestal, attaching the steel tower flange to the loading pedestal. Lastly, the top surface of the loading pedestal is reinforced with a mat of longitudinally oriented #20M reinforcement bars spaced at 200 mm in both directions.

In the corresponding KWF finite element model, the concrete components of the wind turbine foundation are entirely represented using regular hexahedral elements. In order to integrally represent the complex geometry of the angled longitudinal reinforcement, distinct concrete material types with equivalent amounts of smeared reinforcement are assigned to elements corresponding to longitudinally reinforced regions of the foundation. For ease of implementation and calculation, smeared reinforcement components are established by the aggregate percentage of steel oriented in each orthogonal direction. In contrast, the vertical reinforcement 'Z' bars and post-tensioned anchor bolts positioned in the loading pedestal are discretely represented using truss elements. As well, the supporting soil substrate beneath the foundation base pad is represented using truss elements exhibiting equivalent compression-only stiffness per unit area of soil. General overviews of the prescribed KWF reinforced concrete and discrete reinforcement material properties are provided in Table 9.6 and Table 9.7, respectively. Corresponding isometric views of the KWF VecTor3 model are provided in Figure 9.12. Finally, see Appendix A for a complete series of section views of KWF displayed using the pre-processor program FormWorks.

Mat. No.	Colour	Concrete Properties						Smeared Reinforcement Properties									Description
		Type	f'c	f't	Ec	ν	a	Reinf. Ratio, ρ			db	fy	fu	Es	εsh	εu	
								k (x-dir)	l (y-dir)	m (z-dir)							
			MPa	MPa	MPa	-	mm	%			mm	MPa	MPa	MPa	me	me	
1								no smeared reinforcement									· plain concrete
2								1.25	1.25	0	25						· bottom mat long. reinforcement · #20M + #30M @ 200 mm each way
3								0.375	0.375	0	20						· bottom mat long. reinforcement · #20M @ 200 mm each way
4								0.75	0.781	0.211	25						· top mat long. reinforcement · #25M @ 200 mm each way · bars angled in x-direction
5								0.75	0.75	0.422	25						· top mat long. reinforcement · #25M @ 200 mm each way · bars angled both x- and y- directions
6								0.781	0.75	0.211	25						· top mat long. reinforcement · #25M @ 200 mm each way · bars angled in y-direction
7		Reinforced Concrete	30	1.81	25,000	0.15	20	0.781	0.781	0	25	400	600	200,000	10	145	· top mat long. reinforcement · #25M @ 200 mm each way
8								0.409	0.409	0	20						· loading pedestal long. reinforcement · #20M @ 200 mm each way
12								0.975	1.0	0.211	25						· intersection between top and bottom mat long. reinforcement mats · #20M + #25M each way · top bars angled in x-direction
13								1.0	0.975	0.211	25						· intersection between top and bottom mat long. reinforcement mats · #20M + #25M each way · top bars angled in y-direction
9				0.01				0.409	0.409	0	20						· loading pedestal long. reinforcement · #20M @ 200 mm each way · concrete tensile strength ~0 MPa
10			0.007	*	*	*	*	no smeared reinforcement									· hexahedral elements between soil compression-only truss bar elements
11		Structural Steel	400	*	*	*	*	no smeared reinforcement									

Table 9.6: VecTor3 Example Model RC Material Specifications

Reinf. No.	Colour	Type	Area	D_b	f_y	f_u	E_s	ε_{sh}	ε_u	Description
			mm^2	mm	MPa	MPa	MPa	me	me	
1		Compression-only Reinforcement	16.66	0.1	9,999	10,000	100,000	150	200	• soil truss elements with equivalent stiffness as VecTor2 model
2		Ductile Steel Reinforcement	3,333	25	400	600	200,000	10	145	• Vertical 'Z' bar long. reinforcement truss bars • Total area of 20,000 mm^2 over 6 elements
3		Ductile Steel Reinforcement	13,333	35	950	1,050	200,000	5	35	• High-strength post-tensioned anchor bolt truss bars • Total area of 66,667 mm^2 over 5 elements • 2.1 x 10^{-3} pre-strain
4		Ductile Steel Reinforcement	6,667	35	950	1,050	200,000	5	35	• High-strength post-tensioned anchor bolt truss bars, half-area • Total area of 13,333 mm^2 over 2 elements • 2.1 x 10^{-3} pre-strain

Table 9.7: VecTor3 Example Model Discrete Reinforcement Material Specifications

9.4.3.3 Restraints

A variety of nodal restraints are applied to the VecTor3 model in order to ensure static equilibrium throughout all stages of analysis, as well as enforce strictly symmetrical behaviour about the plane of symmetry while still allowing the foundation half-model to realistically displace under loading. For example, all nodes positioned the axis of symmetry (user-established as the x-z plane) are assigned restraints in the y-direction. The lowest-depth layer of nodes in the z-axis, corresponding to the bottom ends of the compression-only truss elements representing soil, are fixed in all x-, y-, and z-directions. Lastly, a single node on the extreme x-axis edge of the foundation model is restrained in the x-direction. VecTor3 example model nodal restraints are presented in Janus as depicted in Figure 9.13 below.

9.4.3.4 Loads

All prescribed design loads are applied to the VecTor3 model as series of nodal loads. In total, four distinct load cases are prescribed for KWF: Load Cases 1 to 3 represent monotonically increasing axial, shear and in-plane moment loads applied to the foundation pad through the tower flange and loading pedestal, while Load Case 4 represents constant gravity loads resulting from foundation self-weight and the overlying soil. Load Cases 1 to 3 are applied in 0.05 factor increments, while the load factor for Load Case 4 is held constant at 1.0 throughout all load stages.

For accurate simulation of true load conditions, several notable design considerations and load modifications are employed. For example, the magnitude of the prescribed design loads are accordingly halved to accommodate for a simulated half-model. In order to realistically simulate the expected path of loading, nodal loads for Load Cases 1 to 3 are applied to the representative anchor bolt truss elements presumed to integrally connect the modelled RC foundation to the base of the wind turbine itself. Additionally, to accommodate the corresponding opposite half of the foundation excluded from the VecTor3

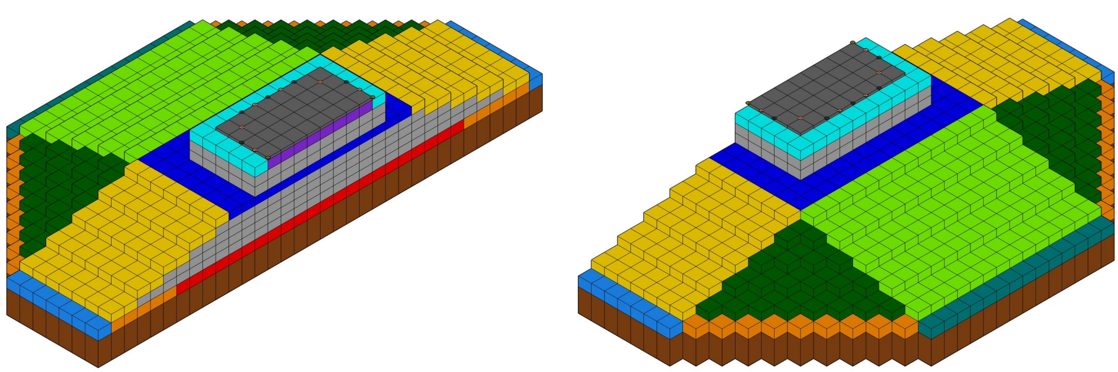


Figure 9.12: VecTor3 Example Model Element Materials in Isometric View

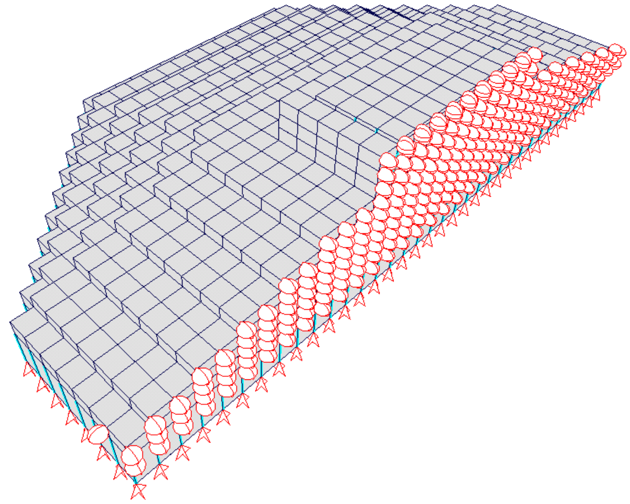


Figure 9.13: VecTor3 Example Model Restraints

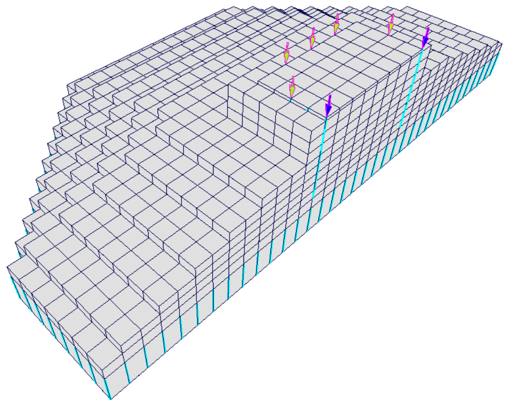
model, nodal loads applied at plane of symmetry are also halved in magnitude. See Table 9.8 for a tabular summary of nodal loads applied to VecTor3 model KWF. Similarly, see Figure 9.14 for Load Cases 1 to 4 as visualized by Janus.

9.4.3.5 Results

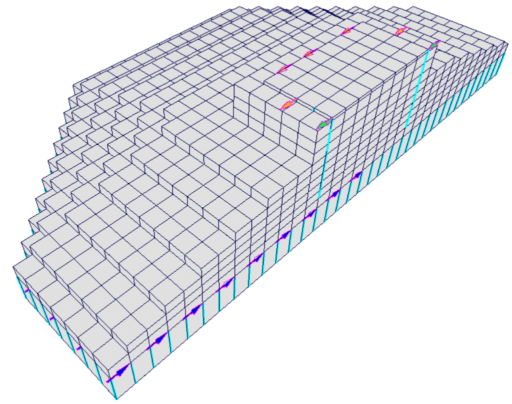
Using results obtained from the analysis of VecTor3 model KWF, a variety of post-processing features in Janus may be demonstrated. Using the available facilities for manipulating the current load stage (refer to Section 3.3 for descriptions of load stage controls), result modes such as Deformations, Crack Pattern, Hotspot mode and contour mode will update to reflect the current stress-strain state of the model. For reference purposes, each of the KWF expanded analysis output files are noted to be approximately 2,800 kB in size. Appendix B presents images of model KWF displayed in Janus, with progressive structural deformations and crack pattern generated on the model plane of symmetry, serving as an effective cross-

Load Case	**1**	**2**	**3**	**4**
Description	Axial	Shear	Moment	Gravity
Loads	(241.67 kN · 5 nodes) + (120.83 kN · 2 nodes)	(75.83 kN · 5 nodes) + (37.92 kN · 2 nodes)	5110.9 kN · 3.5 nodes · 1.906 m	(43.33 kN · 9 nodes) + (134.78 kN · 34 nodes)
Total	1,450 kN	455 kN	34,100 kNm	390 kN (self-weight) + 4582.5 kN (soil)

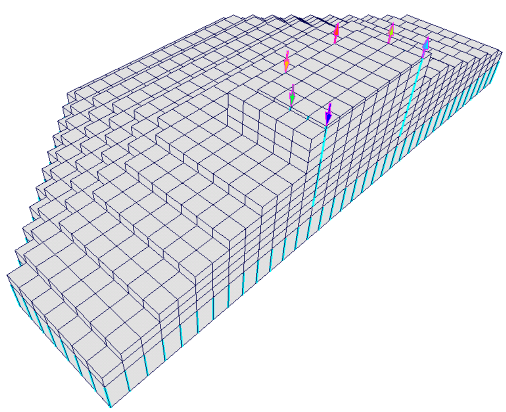
Table 9.8: VecTor3 Example Model Load Case Overview



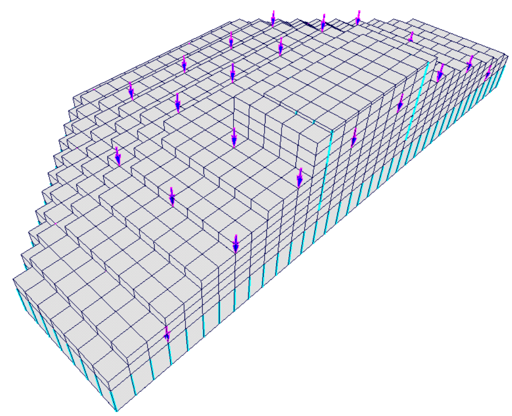
a) Load Case 1 - Axial



b) Load Case 2 - Shear



c) Load Case 3 - Moment



d) Load Case 4 - Gravity

Figure 9.14: VecTor3 Example Model Load Cases

sectional view through the foundation centreline. The structural deformations and crack pattern results are provided in five load stages intervals, corresponding to 0.25 load factor increments.

Additionally, internal crack patterns and applicable contour modes may be simultaneously activated in Section View by enabling the following mode and view options:

- **Results** ▷ *Crack Pattern*,
- A non node-related[1] contour mode (e.g. ε_x), and
- *Section XY*, *Section YZ*, or *Section XZ* section view

See Appendix C for intermediate *XZ Section* section views of model KWF crack patterns and total x-direction strains at regular y-coordinate intervals through the structure. The crack pattern and contour mode values are captured at load stage 26, corresponding to an incremental load factor of 1.30.

[1]node-related colour gradient contour modes are not applicable for activation within Section View

Observing the ultimate deformed shape of the foundation model, KWF exhibits a predominant rotation about the x-z axes, with one end of the base pad lifting off from the supporting soil elements. Utilizing the *Data Platform* dialog, the centreline rotation of the VecTor3 model about the x- and z-axes may be determined using the x- and z-displacements per load stage for top and bottom central nodes and calculating the resulting rotation using small-angle approximation and standard trigonometric relationships. The resulting load factor versus x-z plane rotation of model KWF is depicted in Table 9.15 below.

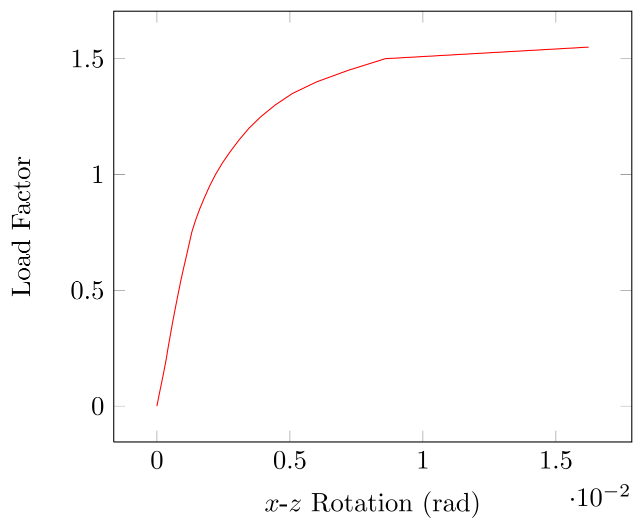


Figure 9.15: VecTor3 Example Model Load-Rotation Response

9.5 Viewing VecTor4 Models

9.5.1 Opening a VecTor4 Model

As with any VecTor model type, VecTor4 models are opened in Janus by submitting a relevant job file specified with a “4” in the entry prompt for “STRUCTURE DATA” ▹ “Structure Type”. Upon recognition of a VecTor4 job file, Janus begins the standard procedures for reading expanded structure, load and analysis output files. However, Janus also executes additional file opening procedures for the VecTor4-specific files previously described in Subsection 2.7.6. All VecTor4-specific files prescribed for opening in Janus are expected to have file names congruent to the specified structure file name correspondingly listed under “STRUCTURE DATA” ▹ “File Name (8 char. max.)” in the job file. See Table 9.9 for an overview of file types associated with VecTor4 as well as corresponding file name entries in the job file.

File Type	File Extension	File Name Entry
Expanded Structure File	S4E	"STRUCTURE DATA" ▷ "File Name (8 char. max.)"
Expanded Load File	L4E	"LOADING DATA" ▷ "Load Case" (1-5) ▷ "File Name (8 char. max.)"
Expanded Analysis Output File	A4E	"LOADING DATA" ▷ "Load Series ID (5 char. max.)" note: each expanded analysis output file name is also serialized using the following format: "(Load Series ID)_i.A4E", where i is the load stage number.
Gauss Point List File	GPL	"STRUCTURE DATA" ▷ "File Name (8 char. max.)"
Node Directional Cosine File	DCN	"STRUCTURE DATA" ▷ "File Name (8 char. max.)"
Gauss Point Directional Cosine File	DCG	"STRUCTURE DATA" ▷ "File Name (8 char. max.)"

Table 9.9: VecTor4 File Types

9.5.2 Feature Overview

As previously described in Chapter 6, VecTor4 is a finite element analysis program used to analyze shell and plate structures in 3D space. VecTor4 uses a special layered heterosis element for representing RC structural components, notable for its capacity for accommodating in-plane rotations as well as exhibiting sectional response characteristics. The distinct requirements associated with displaying results for such a unique element type in 3D space are addressed through VecTor4-customized view and result mode features in Janus.

When viewing VecTor4 models, several different view and mode options may be simultaneously activated for improved versatility in post-processing activities. Accordingly, due to limited display capabilities and/or incompatible display functionality, some specific Janus visualization features are not compatible for combined use with others. A tabular overview of available VecTor4 post-processing features is provided in Table 9.10 below. Check marks (✓) denote that the two intersecting mode and/or view features may be used at the same time, while crosses (✗) conversely demonstrate that the modes cannot both be active at once.

	Load Cases	Restraints	Materials	Deformations	Nodal Contour	RC Element Contour	Truss Contour	Hotspot	XY Section	YZ Section	XZ Section	Layer View
Load Cases												
Restraints	✗											
Materials	✓	✓										
Deformations	✓	✓	✓									
Nodal Contour	✓	✓	✗	✓								
RC Element Contour	✓	✓	✗	✓	✗							
Truss Contour	✓	✓	✗	✓	✗	✗						
Hotspot	✗	✗	✗	✓	✗	✗	✗					
XY Section	✓	✓	✓	✓	✗	✓	✓	✓				
YZ Section	✓	✓	✓	✓	✗	✓	✓	✓	✗			
XZ Section	✓	✓	✓	✓	✗	✓	✓	✓	✗	✗		
Layer View	✗	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗	

Table 9.10: Janus Feature Overview for VecTor4 Models

9.5.3 Example Model

This subsection is dedicated to presenting post-processing features for VecTor4 models in Janus using a previously developed example model. The structure of interest is a two-pack cluster of cylindrical reinforced concrete silos used for the purpose of cement storage. The resulting VecTor4 model is an single silo of the pair, with the integrally adjoining companion silo represented using an isolated 60° arc segment of the silo walls with appropriately assigned restraint conditions. The curved reinforced concrete surfaces which constitute the majority of the silo geometry present an ideal application for the use of VecTor4 as a shell element analysis program.

9.5.3.1 Description

Each of the storage silos are approximately 56.1 m in height, with an inner diameter of 18.3 m, and a typical wall thickness of approximately 279 mm. The common wall shared by the two silos is typically 610 mm in thickness. The silos discharge stored material using an inverted cone hopper, with the silo walls at the base of the cone fortified using a reinforced concrete ring-beam. Beneath the ring-beam, the typical silo wall width increases to 635 mm, and the common wall width increases to 1,321 mm in thickness. Throughout the silo, longitudinal reinforcement is prescribed in both the vertical and circumferential (hoop) directions. The structural component of the ring-beam is the sole region of the storage silo which possesses transverse (out-of-plane) steel reinforcement for shear strength enhancement purposes. See Figure 9.16 below for a conceptual representation of the described storage silo model.

The resulting dimensions for VecTor4 silo model M2E are true to the actual structural specifications, using a total of 588 shell elements to represent the whole single silo as well as an integrated 60° section of the adjoining silo. A centreline radius of 9.3 m is adopted for all established shell element mid-depth node coordinates. All reinforcement prescribed for the silo are constitutively represented within the model using smeared reinforcement components. Within model M2E, three distinct types of silo wall/shell element regions are recognized: "common" walls represent the conjoined wall region between two silos, "typical" walls represent the singular wall of the silo of interest, and "adjoining" walls represent the free wall section edges of the partially included adjacent silo. An overall presentation of the VecTor4 finite element model plan and cross-sectional details is provided in Figure 9.17 below.

9.5.3.2 Structure

Regarding structural provisions for the two-pack silos, the concrete strength of the silo walls is prescribed to vary by height. For 15.2 m and above, the concrete strength is typically specified as 24.1 MPa. The strength of concrete for the remaining base section of the silo is specified to be approximately 27.6 MPa. In accordance with in-situ compressive strength results of core samples extracted from the actual structure, the vertical wall region between the heights of 10.7 m and 15.2 m is assigned a reduced concrete strength of 26.1 MPa. All reinforcement steel utilized in the silo is assigned a minimum yield stress of 414 MPa.

The VecTor4 model M2E exclusively employs shell elements to represent both the concrete and reinforcement components of the cement storage silo. Overall, a total of 33 RC material types are utilized in model M2E; at 11 distinct intervals over the height of the silo, each of the three material types associated with the "common", "typical", and "adjoining" wall types are accordingly modified to express unique combinations of material properties, such as: concrete strength, smeared steel reinforcement quantities/orientations, transitions in wall thickness, etc. A complete tabular overview of the VecTor4 example model RC material types is provided in Table 9.11, corresponding to the model details specified in Figure 9.17. As depicted in the conceptual plan view of model M2E, certain "typical" and "adjoining" wall type shell elements appear to overlap and occupy conflicting volumes in space. Recall that VecTor4 shell elements are declared on a centreline basis; apparent volume overlap occurs due to shell elements sharing a common row of nodes (as well as top and bottom element vertices) on one edge, with the opposing edge nodes orienting the connected elements in similar yet distinct directions.

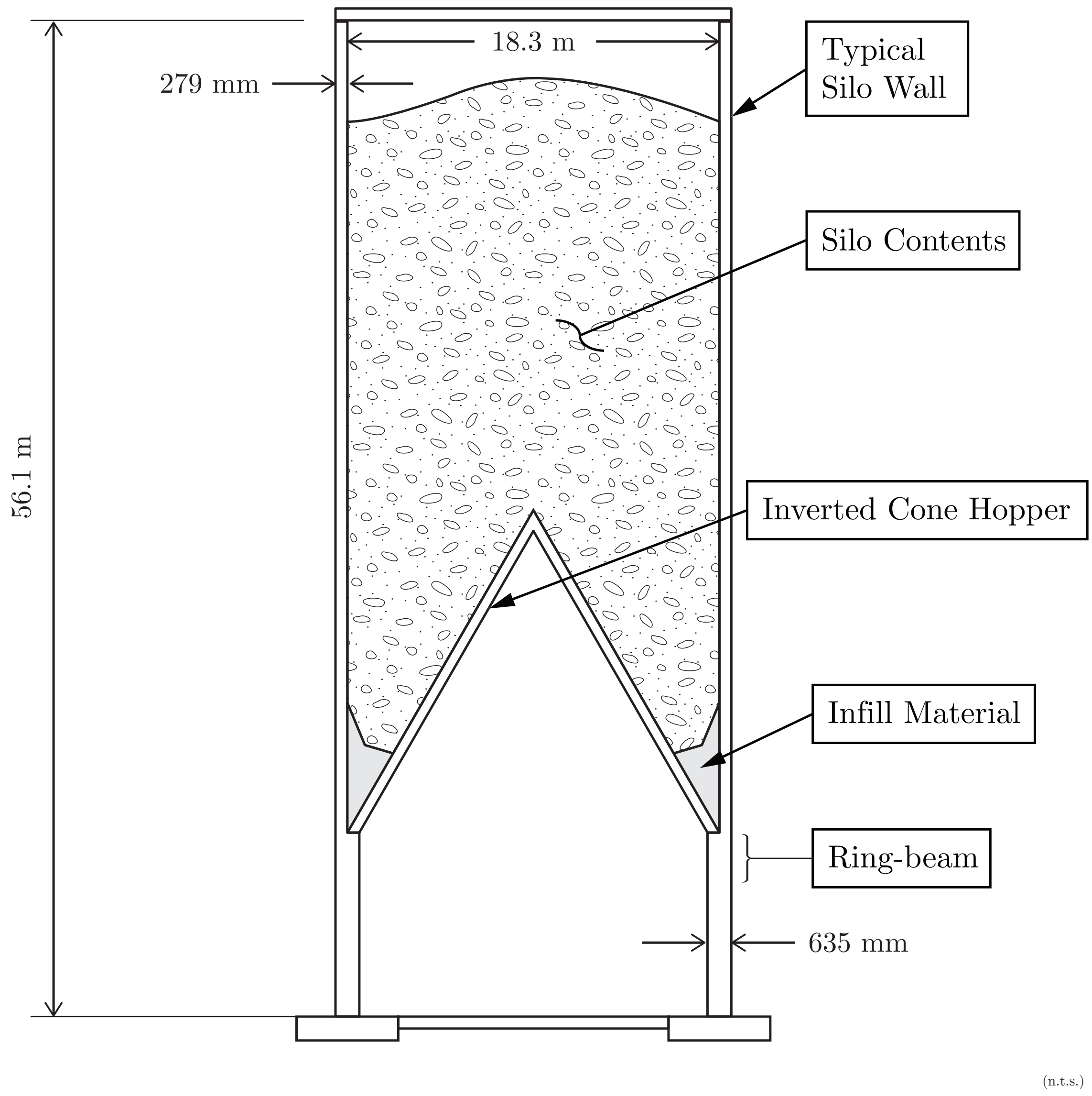


Figure 9.16: Storage Silo Details and Dimensions

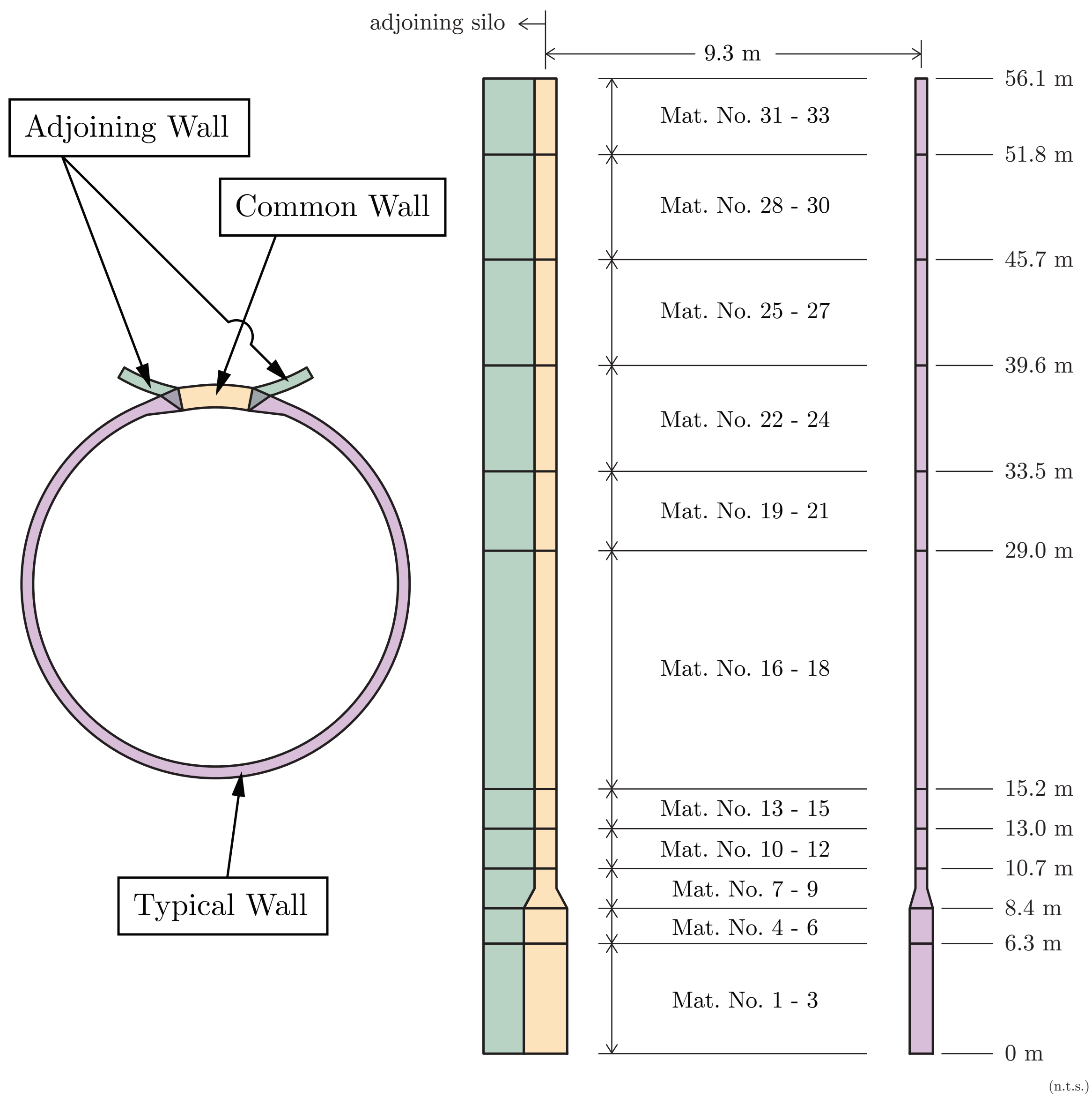


Figure 9.17: VecTor4 Example Model Plan and Cross-Sectional Details

Mat. No.	Wall Type	Location (Height – Height)	Concrete Properties f'_c	f'_t	E_c	ε_0	Smeared Reinforcement Properties: Vertical Reinf. Outside	Vertical Reinf. Inside	Hoop Reinf. Outside	Hoop Reinf. Inside	Trans. Reinf. ρ_t	f_y	f_u	E_s	ε_{sh}	ε_u
		m – m	MPa	MPa	MPa	me	mm^2 / m	mm^2 / m	mm^2 / m	mm^2 / m	%	MPa	MPa	MPa	me	me
1	common															
2	typical	0 – 6.3							801.5	801.5	n/a					
3	adjoining															
4	common							623.4								
5	typical	6.3 – 8.4	27.6	1.73	27,504	2.007			962.1	962.1						
6	adjoining															
7	common															
8	typical	8.4 – 10.7														
9	adjoining								498.7	498.7						
10	common															
11	typical	10.7 – 13.0														
12	adjoining		26.1	1.69	26,156	1.996										
13	common								4987.3	3989.8						
14	typical	13.0 – 15.2							4987.3	3989.8						
15	adjoining								3989.8	4987.3						
16	common								4987.3	3989.8						
17	typical	15.2 – 29.0					623.4		4987.3	3989.8		414	517.5	200,000	10	200
18	adjoining								3989.8	4987.3						
19	common								3989.8	3324.9						
20	typical	29.0 – 33.5						277.1	3989.8	3324.9	n/a					
21	adjoining								3324.9	3989.8						
22	common								3324.9	2659.9						
23	typical	33.5 – 39.6							3324.9	2659.9						
24	adjoining		24.1	1.62	24,334	1.981			2659.9	3324.9						
25	common								2545.6	2036.5						
26	typical	39.6 – 45.7							2545.6	2036.5						
27	adjoining								2036.5	2545.6						
28	common								1870.2	1320.2						
29	typical	45.7 – 51.8							1870.2	1320.2						
30	adjoining								1320.2	1870.2						
31	common								767.3	498.7						
32	typical	51.8 – 56.1							767.3	498.7						
33	adjoining								498.7	767.3						

Table 9.11: VecTor4 Example Model Material Specifications

9.5.3.3 Restraints

In order to analogously represent the structural contributions of the omitted silo walls, all nodes along the free edges of the "adjoining" wall segments are restrained from moving laterally in the y- and z-directions. However, these edge nodes may still translate vertically (user-established as the global x-direction) as well as exhibit in-plane rotations. As equivalently pinned foundation supports, the bottom-most nodes of the silo model are restrained in all orthogonal x-, y-, and z-directions, but are free to rotate in both in-plane directions. See Figure 9.18 for a Janus-rendered depiction of nodal restraints for model M2E.

9.5.3.4 Loads

The loading scenario applied to finite element model M2E is discretized into three distinct load cases. The combined loads are anticipated to represent the eccentric loading that occurs while the contents of the silo are unloaded through a designated location at the base of the inverted cone hopper. While dispensing out from the base of the inverted cone hopper, the stored granular material exerts a combination of lateral and downward frictional forces upon the silo walls, and follows an eccentric pressure distribution - conceptually presented in plan view in Figure 9.19 below. Loads associated with the silo contents are accordingly reduced at the base of the silo as a result of the modified volumetric geometry due to presence of the inverted cone hopper and infill material lining the bottom of the cone.

The first load case, Load Case 1, is identified as the friction pressure of the stored material bearing vertically down upon the interior surface of the silo. The frictional pressure is represented as a series of equivalent point loads applied in the negative x-direction at mid-depth node locations about the

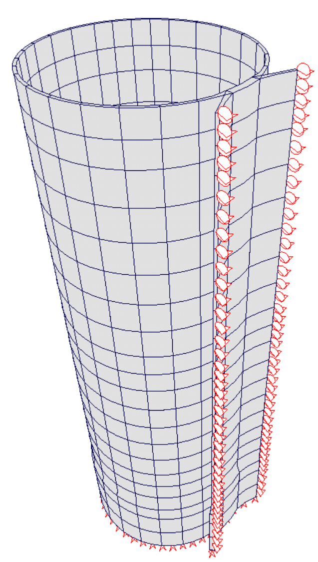


Figure 9.18: VecTor4 Example Model Restraints

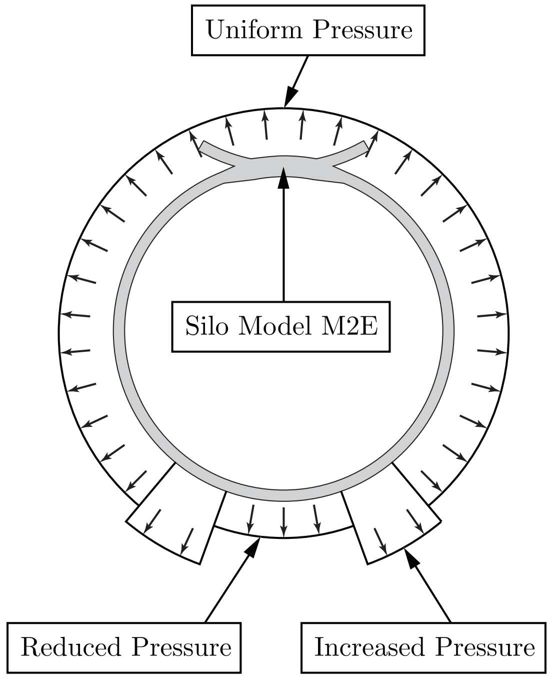


Figure 9.19: VecTor4 Example Model Pressure Distribution

circumference and through the loaded height of the silo. The point load magnitudes are assigned based on the prescribed pressure distribution, depth of granular material, as well as the tributary inner surface area of the shell elements that the stored material acts upon. See Figure 9.20 for a depiction of Load Case 1 point load arrows superimposed on VecTor4 model M2E in Janus.

The second load case embodies the outward lateral pressure attributed to the contents of the storage

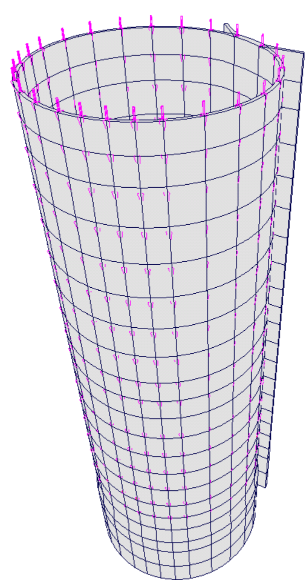


Figure 9.20: VecTor4 Example Model Load Case 1

silo, accordingly modified to match the eccentric loading scheme. Load Case 2 assigns lateral pressures per element as a uniformly distributed stress applied across the planar surface of the shell element. See Figure 9.21 for a graphical plot representing of the lateral pressure distribution of Load Case 2 over the loaded height of the silo model. Lastly, the Load Case 3 represents the silo self-weight. Based on provided material density properties, equivalent self-weight values are assigned per element. Load Cases 1 and 2 are incrementally applied in 0.25 load factor intervals per load stage, while Load Case 3 is held constant at 1.0. Currently, Janus does not provide features for visualizing element-based load cases as part of its post-processing facilities.

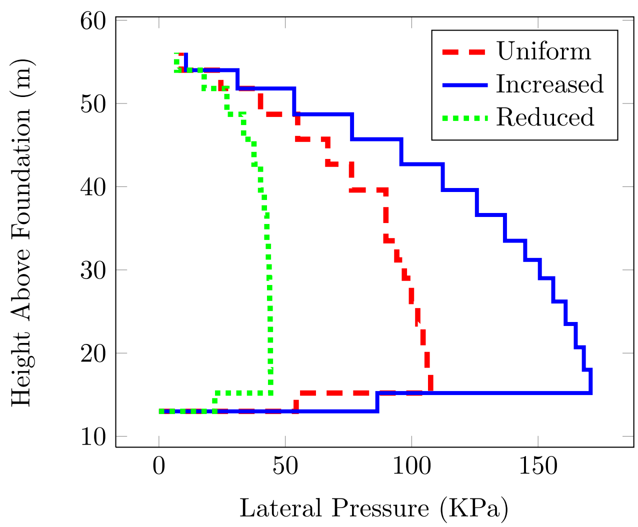


Figure 9.21: VecTor4 Example Model Load Case 2 Lateral Loads vs. Height

9.5.3.5 Results

Utilizing the analysis output files from model M2E, a variety of VecTor4 post-processing features in Janus may be demonstrated. It is worthy to note that the expanded analysis result files of VecTor4 model M2E are each 40 MB in size, corresponding to approximately 40 million characters listed along 415,000 lines of data. As with any other finite element analysis, users may view progressive structural deformations as the silo structure is subjected to a monotonically increasing eccentric load scenario. Appendix D presents a series of figures displaying renderings of model M2E in Janus with both Deformations mode and contour mode for nodal displacements in the lateral y-direction concurrently enabled. Each successive figure presents the silo model as the load factor is cumulatively increased in 0.50 intervals, resulting in an ultimate load factor of approximately 2.0 for Load Case 1 and 2. For all load stages, Load Case 3 is held constant at 1.0.

Figure 9.22 demonstrates the VecTor4 example model in a combined Deformations mode and contour mode for in-plane concrete strains. Accordingly, Layer View may be utilized to further investigate the layer-by-layer distribution of concrete stress- and strain-related properties. Figure 9.23 depicts Layer View of a shell element located in the enhanced pressure region, with the the same in-plane shear stress contour mode enabled in Janus. Layer View is activated using the *Layer View* toolbar button and selecting the shell element number of interest. Shell element numbers provided in the dialog drop-down list for Layer View correspond with the numerical identifiers assigned to each shell element in the VecTor4 expanded structure file.

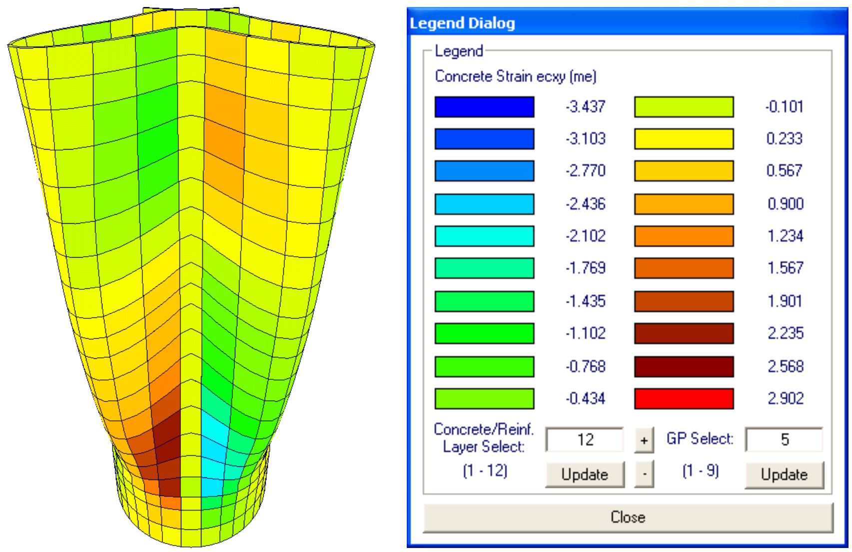


Figure 9.22: VecTor4 Example Model in Deformations Mode and RC Element Contour Mode

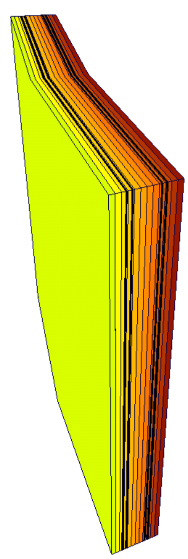


Figure 9.23: VecTor4 Example Model in Layer View and RC Element Contour Mode

9.6 Viewing VecTor5 Models

9.6.1 Opening a VecTor5 Model

In the Janus file opening procedures, VecTor5 models are identified by an integer value of "5" in the job file entry prompt corresponding to "STRUCTURE DATA" ▷ "Structure Type". Using the provided expanded structure, load, and analysis output file naming conventions provided in the job file, Janus proceeds to search and open VecTor5-specific file types for the active model, outlined in Table 9.12 below.

It is important to note that post-processing visualization features for VecTor5 output member elements are only available in Janus if the output data is generated by VecTor5. Output member elements must be initially identified prior to the onset of analysis by listing the members under the "Detailed Member Output List" heading in the original VecTor5 structure file (file extension ".S5R"). Based on the list of output member elements in the structure file, VecTor5 analysis output files include additional data describing sectional analysis parameters for each attributed output member element. In the event that no output member element data are encountered while reading VecTor5 analysis output files, applicable Janus results features such as Layer View, Hotspot mode, and certain *Results* menu options will be disabled from activation.

9.6.2 Feature Overview

VecTor5 provides analysis capabilities for two-dimensional RC planar frame structures, producing analytical solutions for member forces and deformations on a global scale as well as local sectional performance characteristics of specific member elements. The unique result format established by VecTor5 necessitates an equivalently distinct array of visualization features for frame analyses in Janus. Table 9.13

File Type	**File Extension**	**File Name Entry**
Expanded Structure File	S5E	"STRUCTURE DATA" ▷ "File Name (8 char. max.)"
Expanded Load File	L5E	"LOADING DATA" ▷ "Load Case" (1-5) ▷ "File Name (8 char. max.)"
Expanded Analysis Output File	A5E	"LOADING DATA" ▷ "Load Series ID (5 char. max.)" note: each expanded analysis output file name is also serialized using the following format: "(Load Series ID)_i.A5E", where i is the load stage number.

Table 9.12: VecTor5 File Types

provides a comprehensive overview of VecTor5-specific post-processing functions in Janus as well as their inter-compatibility with each other. Check marks (✓) denote that the two intersecting mode and/or view features may be concurrently enabled, while crosses (✗) conversely demonstrate that the enabling one mode will disable the other.

	Load Cases	**Restraints**	**Materials**	**Deformations**	**Nodal Contour**	**RC Element Contour**	**Long. Reinf. Contour**	**Member Forces**	**Member Deform.**	**Hotspot**	**Layer View**
Load Cases											
Restraints	✗										
Materials	✓	✓									
Deformations	✓	✓	✓								
Nodal Contour	✓	✓	✗	✓							
RC Element Contour	✓	✓	✗	✓	✗						
Long. Reinf. Contour	✓	✓	✗	✓	✗	✗					
Member Forces	✓	✓	✗	✓	✗	✗	✗				
Member Deform	✓	✓	✗	✓	✗	✗	✗	✗			
Hotspot	✗	✗	✗	✓	✗	✗	✗	✗	✗		
Layer View	✗	✗	✓	✗	✗	✓	✓	✗	✗	✓	

Table 9.13: Janus Feature Overview for VecTor5 Models

9.6.3 Example Model

The following subsection demonstrates several VecTor5 post-processing features in Janus using a previously developed VecTor5 example model. The designated VecTor5 finite element frame model is named EMARA, in reference to the original experimental specimen tested by Vecchio and Emara (1992). Due to the planar characteristics of the experimental frame specimen and load application, VecTor5 is regarded as a wholly applicable and relevant program for verifying the original results obtained from testing the experimental specimen.

9.6.3.1 Description

The large-scale test frame developed by Vecchio and Emara was a single-span, two-story reinforced concrete plane frame. Each span was measured as 3500 mm from centre to centre, with each storey set at a height of 2000 mm. Including the integrated base section, the gross height of the test frame was 4600 mm. All frame members utilized a regular rectangular cross section, prescribed as 300 mm in width and 400 mm in depth. To simulate a rigid ground foundation, the base section was heavily reinforced and anchored to the floor of the testing location. See Figure 9.24 below for an overview of the structural details of the experimental test frame. For comparison, the equivalent finite element model EMARA (as presented in Janus) is displayed in Figure 9.25 below. Overall, EMARA is composed of 92 member elements, utilizing congruent centreline measurements to represent an analogous whole model of the actual test frame specimen.

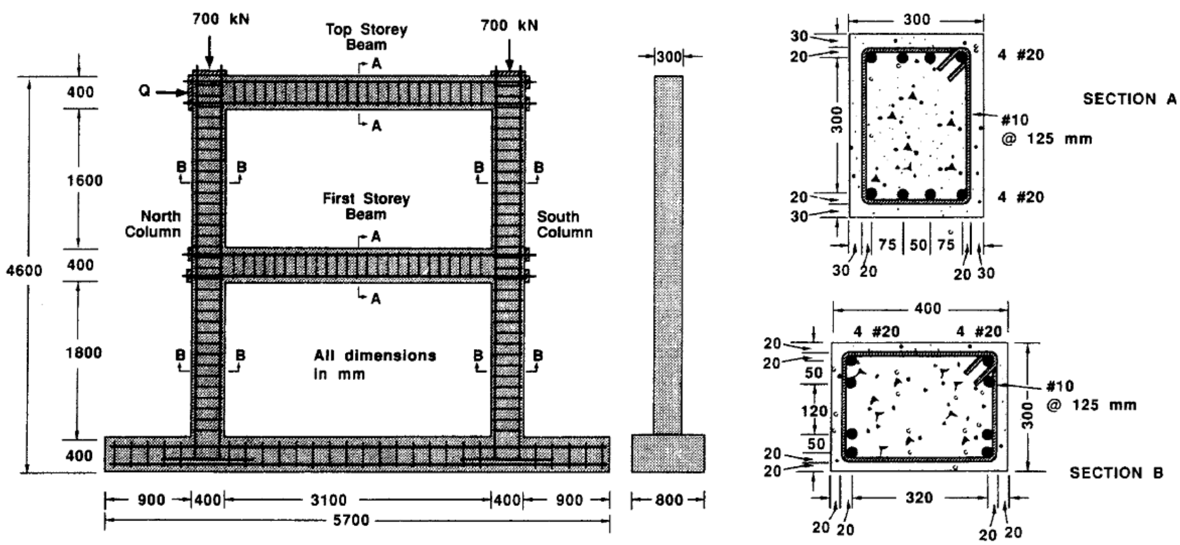


Figure 9.24: Test Frame Specimen Structural Details (adapted from Vecchio and Emara, 1992)

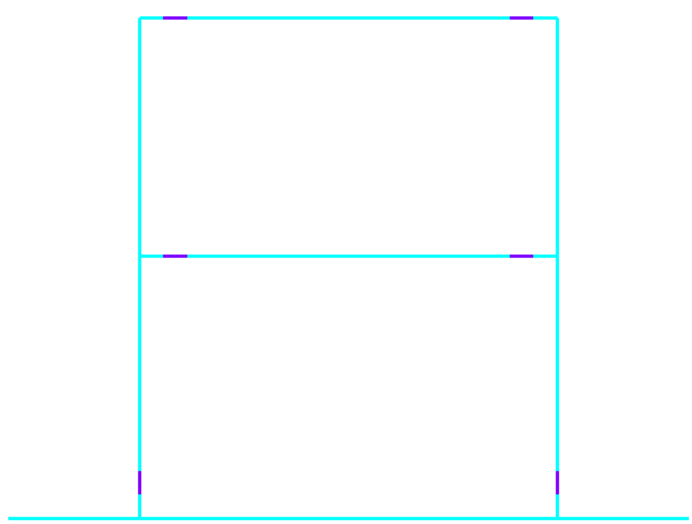


Figure 9.25: EMARA Finite Element Model

9.6.3.2 Structure

As designed by Vecchio and Emara, the test frame specimen was composed entirely of 30 MPa concrete. All members of the frame were longitudinally reinforced using single layers of four No. 20 bars at both top and bottom locations. A series of No. 10 closed stirrups spaced at 125 mm was prescribed for shear reinforcement purposes. Based on tested material samples, the longitudinal reinforcement steel was determined to exhibit yield and ultimate strengths of 418 MPa and 596 MPa, respectively. The shear reinforcement stirrups were tested to have a yield strength of 454 MPa and ultimate strength of 640 MPa. For integral connection of longitudinal reinforcement within the frame, all longitudinal reinforcement bars were extended through the beam-column joints and anchored to stiff bearing plates at the exterior faces of the joints.

Accordingly, material types used in the VecTor5 model EMARA are designated in such a way that all structural components of concrete and reinforcement are essentially represented in the ensuing finite element analysis. All member element sections are stratified into 40 layers, each with a uniform thickness of approximately 10 mm and width of 300 mm. In conformance with the modelling capabilities of VecTor5, the top and bottom longitudinal reinforcement bars are represented as distinct layers specified to act at the respective centroidal depth assigned to the reinforcement layer. The closed steel stirrups are represented as combination of smeared transverse and out-of-plane reinforcement components within individual RC layers. See Table 9.14, Table 9.15, and Table 9.16 below for comprehensive tabular summaries of general and sectional material properties for the VecTor5 example model. Figure 9.26 provides a corresponding overview of EMARA frame member element material types as displayed in Janus.

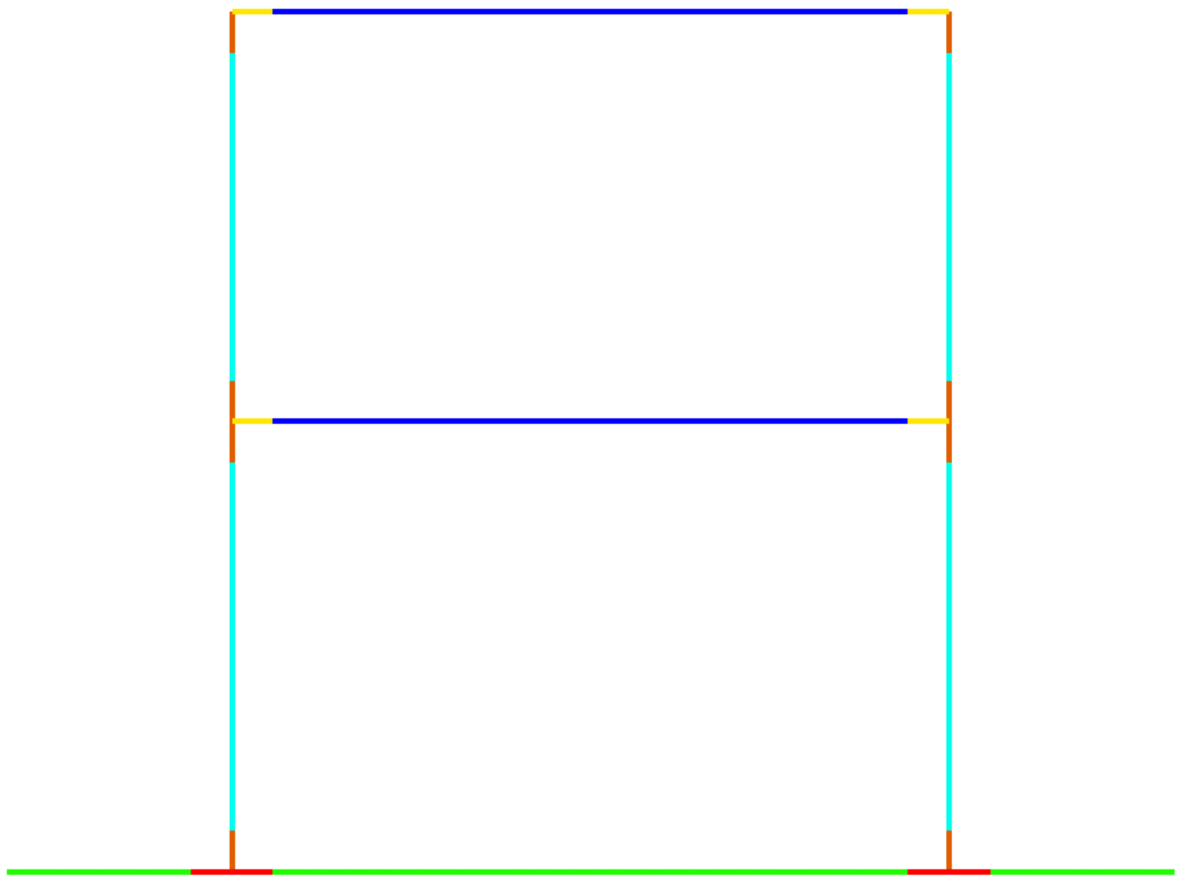


Figure 9.26: VecTor5 Example Model Material Specifications

Mat. No.	Colour	Concrete Properties				Smeared Transverse Reinforcement Properties						
		f'_c	E_c	e_0	Other	S_t	$d_{b,t}$	f_y	f_u	E_s	ε_{sh}	ε_u
		MPa	MPa	me		mm	mm	MPa	MPa	MPa	me	me
1		30	27,386	2.03	default	125	11.3	454	640	200,000	10	100
2												
4												
5												
3		30	27,386	2.03	default	200	11.3	454	640	200,000	10	100
6												

Table 9.14: VecTor5 Example Model General Material Specifications

Mat. No.	Colour	Comp. No.	D_c mm	W_c mm	ρ_t %	ρ_z %	N_x
1		1	10	300	0	0.800	3
		2			0.533	0.800	2
		3			0.533	0.800	5
		4			0.533	0	20
		5			0.533	0.800	5
		6			0.533	0.800	2
		7			0	0.800	3
2		1	10	300	0	0.800	2
		2			0.533	0.800	2
		3			0.533	0.800	5
		4			0.533	0	22
		5			0.533	0.800	5
		6			0.533	0.800	2
		7			0	0.800	2
4		1	10	300	0	1.600	3
		2			1.066	1.600	2
		3			1.066	1.600	5
		4			1.066	0	20
		5			1.066	1.600	5
		6			1.066	1.600	2
		7			0	1.600	3

Mat. No.	Colour	Comp. No.	D_c mm	W_c mm	ρ_t %	ρ_z %	N_x
5		1	10	300	0	1.78	3
		2			1.066	1.78	2
		3			1.066	1.78	5
		4			1.066	0	20
		5			1.066	1.78	5
		6			1.066	1.78	2
		7			0	1.78	3
3		1	10	800	0	0.460	4
		2			0.125	0.460	2
		3			0.125	0.460	5
		4			0.125	0	18
		5			0.125	0.460	5
		6			0.125	0.460	2
		7			0	0.460	4
6		1	10	800	0	0.920	4
		2			0.25	0.920	2
		3			0.25	0.920	5
		4			0.25	0	18
		5			0.25	0.920	5
		6			0.25	0.920	2
		7			0	0.920	4

Table 9.15: VecTor5 Example Model RC Material Specifications (from top of section)

Mat. No.	Colour	Comp. No.	Y_s mm	A_s mm^2	d_b mm	f_y MPa	f_u MPa	E_s MPa	ε_{sh} me	ε_u me
1		1	50	1200	19.5	418	596	200,000	10	100
		2	350							
2		1	40	1200	19.5	418	596	200,000	10	100
		2	360							
4		1	50	2400	19.5	418	596	200,000	10	100
		2	350							
5		1	40	2400	19.5	418	596	200,000	10	100
		2	360							
3		1	60	3000	19.5	418	596	200,000	10	100
		2	340	3300						
6		1	60	6000	19.5	418	596	200,000	10	100
		2	340	6600						

Table 9.16: VecTor5 Example Model Longitudinal Reinforcement Material Specifications (from top of section)

9.6.3.3 Restraints

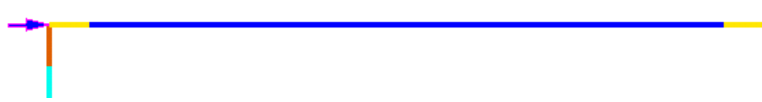
At the time of testing, the base of the experimental test frame was bolted to the test floor, effectively simulating a fully fixed condition. In order to simulate equivalent support conditions in the finite element frame model, a series of bottom nodes of EMARA are specified as pinned restraints, preventing linear displacement in both x- and y-direction degrees of freedom. The equivalent EMARA model base restraints are displayed in Figure 9.27 below.

Figure 9.27: VecTor5 Example Model Restraints

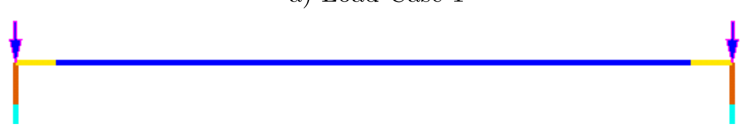
9.6.3.4 Loads

The loading protocol for the experimental test frame prescribes a combined load scenario of axial and shear loads. Firstly, constant axial loads were applied to the columns in order to simulate representative above-story loads bearing down upon the building frame. Secondly, a monotonically increasing horizontal load was applied at the second storey beam-column joint until failure occurred.

The load scenario applied to the test frame is equivalently represented as two distinct load cases in the VecTor5 model EMARA. The lateral component of experimental loading is represented in Load Case 1 as a monotonically increasing nodal displacement in the x-direction. Load Case 2 embodies the constant axial loads as a pair of vertical point loads acting in the negative y-direction, each applying 700 kN in force. Load Case 1 is applied in 5 mm increments, while Load Case 2 is held at a constant value of 1.0. See Figure 9.28 below for depictions of load arrows for Load Cases 1 and 2 superimposed upon the EMARA model in Janus.



a) Load Case 1



b) Load Case 2

Figure 9.28: VecTor5 Example Model Load Cases

9.6.3.5 Results

Based on the applied load scenario and nodal restraints prescribed to the EMARA frame, a variety of VecTor5 numerical results may be displayed in Janus from the 150 kB-sized expanded analysis output files. As per typical Janus post-processing functionality, scale structural deformations may be activated via **Results** ▷ *Deformations*. See Figure 9.29 for a combined demonstration of Deformations mode and x-direction displacement contour mode features at the ultimate load stage. Alternatively, the member deformation parameter of average concrete shear strains is also displayed in Figure 9.30 below.

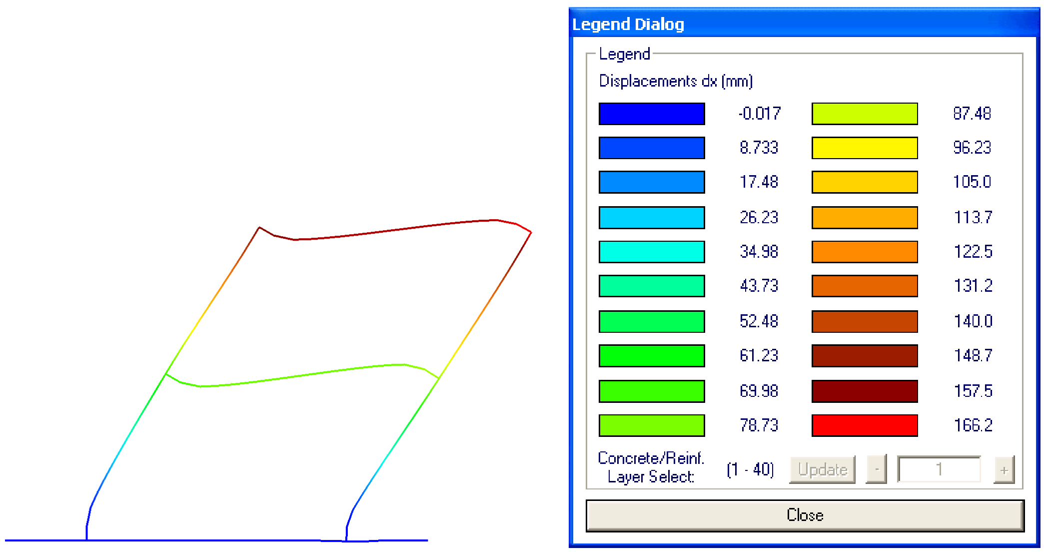


Figure 9.29: VecTor5 Example Model in Deformations Mode and Nodal Displacements Contour Mode

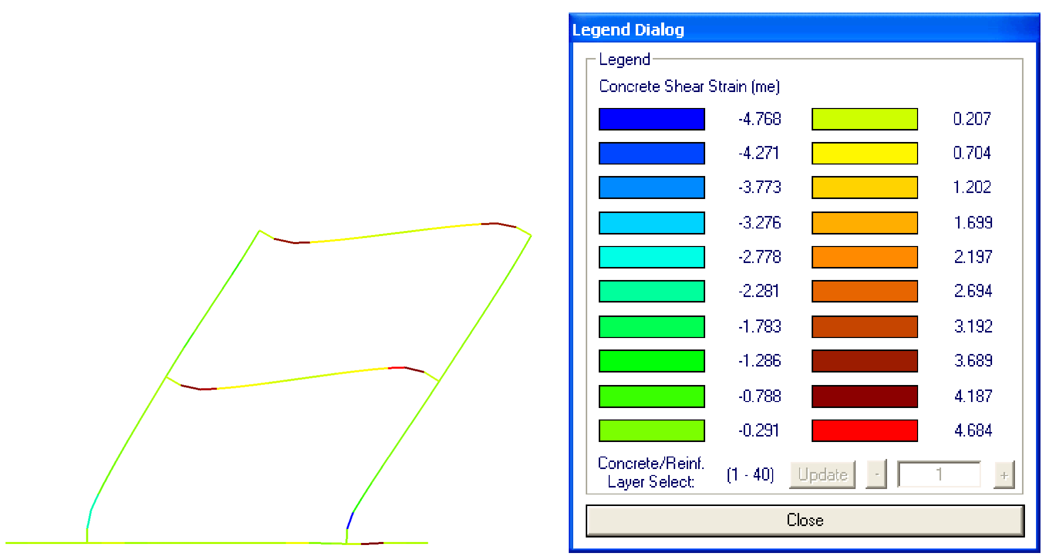


Figure 9.30: VecTor5 Example Model in Deformations Mode and Shear Strain Contour Mode

As previously mentioned, VecTor5 Layer View may be utilized to display sectional performance characteristics of member elements specified in the "Detailed Member Output List" of the VecTor5 structure file ending with the extension ".S5R". An example of VecTor5 Layer View with in-plane shear strain contour mode activated for a first storey horizontal member element is provided in Figure 9.31 below.

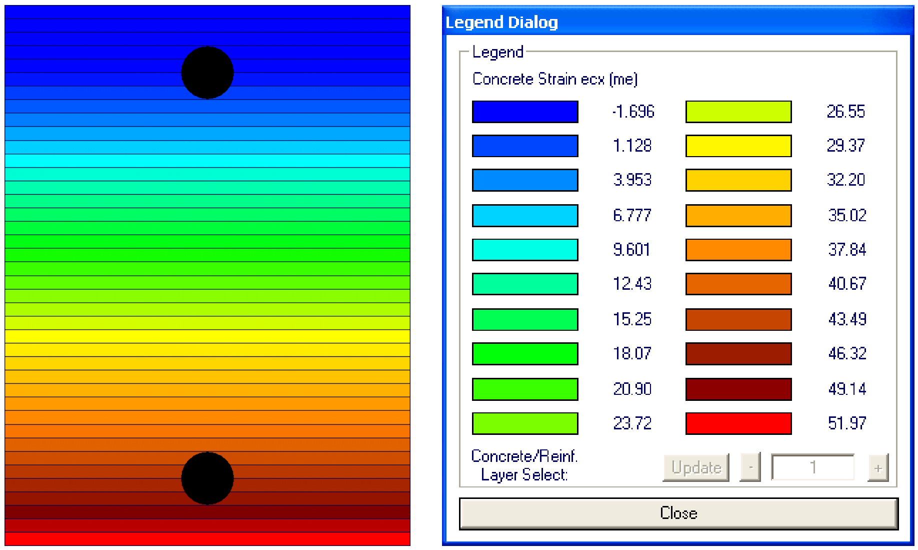


Figure 9.31: VecTor5 Example Model in Layer View and Shear Strain Contour Mode

9.7 Viewing VecTor6 Models

9.7.1 Opening a VecTor6 Model

Janus executes file-reading procedures specific to VecTor6 models upon reading an integer value of "6" in the entry prompt corresponding to "STRUCTURE DATA" ▷ "Structure Type" in the VecTor job file. Janus attempts to locate and read VecTor6-specific file types corresponding to file names listed in the job file. A general description of the VecTor6 file types and file name designations expected by Janus are outlined in Table 9.17 below.

9.7.2 Feature Overview

VecTor6 is a finite element analysis program capable of analyzing axisymmetric structures as a simplified transverse section using representative planar RC and reinforcement element components. Accordingly, several post-processing features in Janus are not applicable while displaying VecTor6 models in a similarly planar manner. Table 9.18 below describes the interactive relationships among the various mode and

File Type	File Extension	File Name Entry
Expanded Structure File	S6E	"STRUCTURE DATA" ▷ "File Name (8 char. max.)"
Expanded Load File	L6E	"LOADING DATA" ▷ "Load Case" (1-5) ▷ "File Name (8 char. max.)"
Expanded Analysis Output File	A6E	"LOADING DATA" ▷ "Load Series ID (5 char. max.)" note: each expanded analysis output file name is also serialized using the following format: "(Load Series ID)_i.A6E", where i is the load stage number.

Table 9.17: VecTor6 File Types

view features enabled in Janus for the purpose of displaying VecTor6 models. A check mark (✓) at a row and column intersection denotes that the two mode and/or view features are compatible for simultaneous activation during post-processing operations. Conversely, crosses (✗) signify that the two features may not be utilized in tandem.

	Load Cases	**Restraints**	**Materials**	**Deformations**	**Nodal Contour**	**RC Element Contour**	**Truss Contour**	**Hotspot**	**XY Section**
Load Cases									
Restraints	✗								
Materials	✓	✓							
Deformations	✓	✓	✓						
Nodal Contour	✓	✓	✗	✓					
RC Element Contour	✓	✓	✗	✓	✗				
Truss Contour	✓	✓	✗	✓	✗	✗			
Hotspot	✗	✗	✗	✓	✗	✗	✗		
XY Section	✗	✗	✓	✓	✗	✓	✓	✓	

Table 9.18: Janus Feature Overview for VecTor6 Models

9.7.3 Example Model

In order to demonstrate Janus' post-processing capabilities for VecTor6 models, a previously developed example model is examined in the following subsections. The VecTor6 example model, designated as

RB2, is a vertical section of a cylindrical reinforced concrete silo wall with an integrated ring-beam.

9.7.3.1 Description

Model RB2 is a representative transverse section of a cylindrical reinforced concrete wall, specified with an outer radius of 12.46 m. The axisymmetric VecTor6 model represents a 5.30 m vertical segment of the wall which encompasses the transition point between the typical wall and integrated ring-beam regions. The x-direction geometry of wall is varied to represent the wall region containing the integrally cast cylindrical ring-beam. The transition point from the typical wall to the combined wall and ring-beam occurs at the approximately 1.83 m from the bottom of model RB2, with the x-direction wall width immediately decreasing from 813 mm to 457 mm in thickness.

9.7.3.2 Structure

The VecTor6 example model is comprised of 1204 quadrilateral elements and 60 ring bar elements. Model RB2 utilizes a combination of 13 RC material types to represent unique quantities of longitudinal/transverse reinforcement components within the the cylindrical wall and ring-beam, each with a compressive strength of approximately 41.4 MPa. Within the typical silo wall, in-plane and out-of-plane reinforcement are represented as a combination of smeared reinforcement and discrete ring bar elements. In contrast, the integrated ring-beam region exclusively utilizes smeared reinforcement material assignments to represent the averaged reinforcement properties of all orientations. The silo wall uses both conventional steel bars and high-strength prestressing strands for circumferential reinforcement. All discrete ring bar element steel is specified with a yield strength of 518 MPa, and an ultimate strength of 650 MPa. The smeared prestressed steel, established on the inner and outer silo wall surfaces, is assigned a yield strength of 1680 MPa, ultimate strength of 1860 MPa, and prestrain value of $4.0{\times}10^3$. Smeared transverse and longitudinal steel components are prescribed with yield and ultimate strengths of 414 MPa and 650 MPa, respectively. See Table 9.19 and Table 9.20 below for a comprehensive overview of RC and steel material types utilized in VecTor6 model RB2. The corresponding colour-coded RC and reinforcement material assignments are depicted in Figure 9.32 below.

9.7.3.3 Restraints

In order to provide support conditions which readily reflect the intermediate cylindrical wall section modelled by RB2, all bottom nodes are assigned as vertical roller supports in the z-direction. See Figure 9.33 for a depiction of model RB2 with nodal restraint symbols superimposed upon the finite element model.

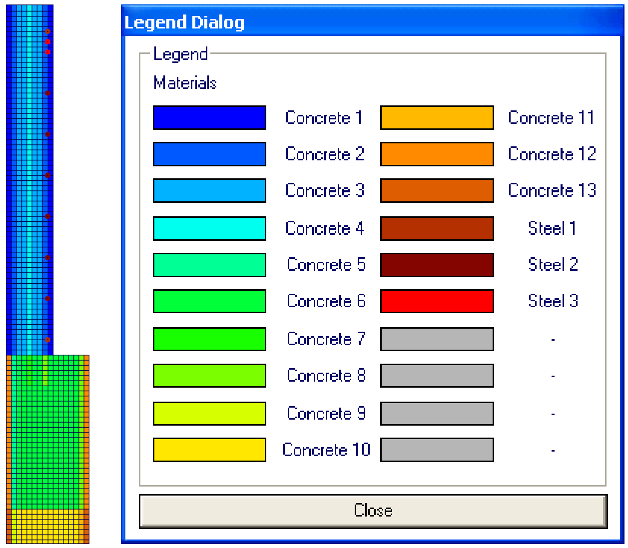


Figure 9.32: VecTor6 Example Model in Material Mode and Legend Dialog

9.7.3.4 Loads

The loading scenario applied to RB2 consists of a comprehensive array of nodal loads which subjects the silo wall model to a combination of shear, moment and axial forces. The loads are intended to represent an inverted cone hopper seated on the ring beam, as well as the lateral/downward load components of stored granular material bearing upon the silo wall. See Figure 9.34 for an illustration of the combined loading applied to VecTor6 model RB2 in Janus. All loads are cumulatively applied in 0.05 factor increments.

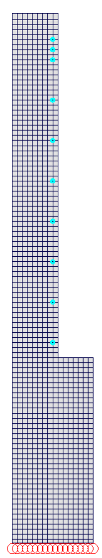


Figure 9.33: VecTor6 Example Model Restraints

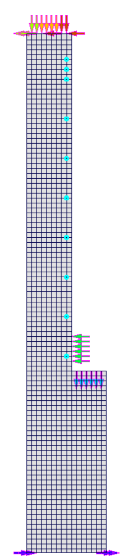


Figure 9.34: VecTor6 Example Model Loads

Mat. No.	Colour	Concrete Properties			Smeared Reinforcement Properties								
		f'_c	f'_t	E_c	Reinf. Ratio, ρ			f_y	f_u	E_s	ε_{sh}	ε_u	Dep
					k (x-dir)	l (y-dir)	m (x-dir)						
		MPa	MPa	MPa	%			MPa	MPa	MPa	me	me	me
1		41.4	2.12	34,500	0	1.764	0	518	650	200,000	10	142	0
					0	1.243	0	1680	1860			82	4.0
2		41.4	2.12	34,500	0.108	0	1.833	414	650	200,000	10	142	0
					0	1.764	0	518	650			142	0
					0	1.243	0	1680	1860			82	4.0
3		41.4	2.12	34,500	0.108	0	0	414	650	200,000	10	142	0
					0	1.764	0	518	650			142	0
					0	1.243	0	1680	1860			82	4.0
4		41.4	2.12	34,500	0.108	0	3.292	414	650	200,000	10	142	0
					0	1.764	0	518	650			142	0
					0	1.243	0	1680	1860			82	4.0
5		41.4	2.12	34,500	0.153	0	1.833	414	650	200,000	10	142	0
					0	0.992	0	518	650			142	0
					0	0.559	0	1680	1860			82	4.0
6		41.4	2.12	34,500	0.153	0	0	414	650	200,000	10	142	0
					0	0.992	0	518	650			142	0
					0	0.559	0	1680	1860			82	4.0
7		41.4	2.12	34,500	0.153	0	3.292	414	650	200,000	10	142	0
					0	0.992	0	518	650			142	0
					0	0.559	0	1680	1860			82	4.0
8		41.4	2.12	34,500	0.153	0	1.833	414	650	200,000	10	142	0
					0	0.992	0	518	650			142	0
					0	0.559	0	1680	1860			82	4.0
9		41.4	2.12	34,500	0	0	1.833	414	650	200,000	10	142	0
					0	0.992	0	518	650			142	0
10		41.4	2.12	34,500	0	0.992	0	518	650	200,000	10	142	0
11		41.4	2.12	34,500	0	0	1.833	414	650	200,000	10	142	0
					0	0.992	0	518	650			142	0
12		41.4	2.12	34,500	0	0.992	0	518	650	200,000	10	142	0
					0	0.559	0	1680	1860			82	4.0
13		41.4	2.12	34,500	0	0.992	0	518	650	200,000	10	142	0

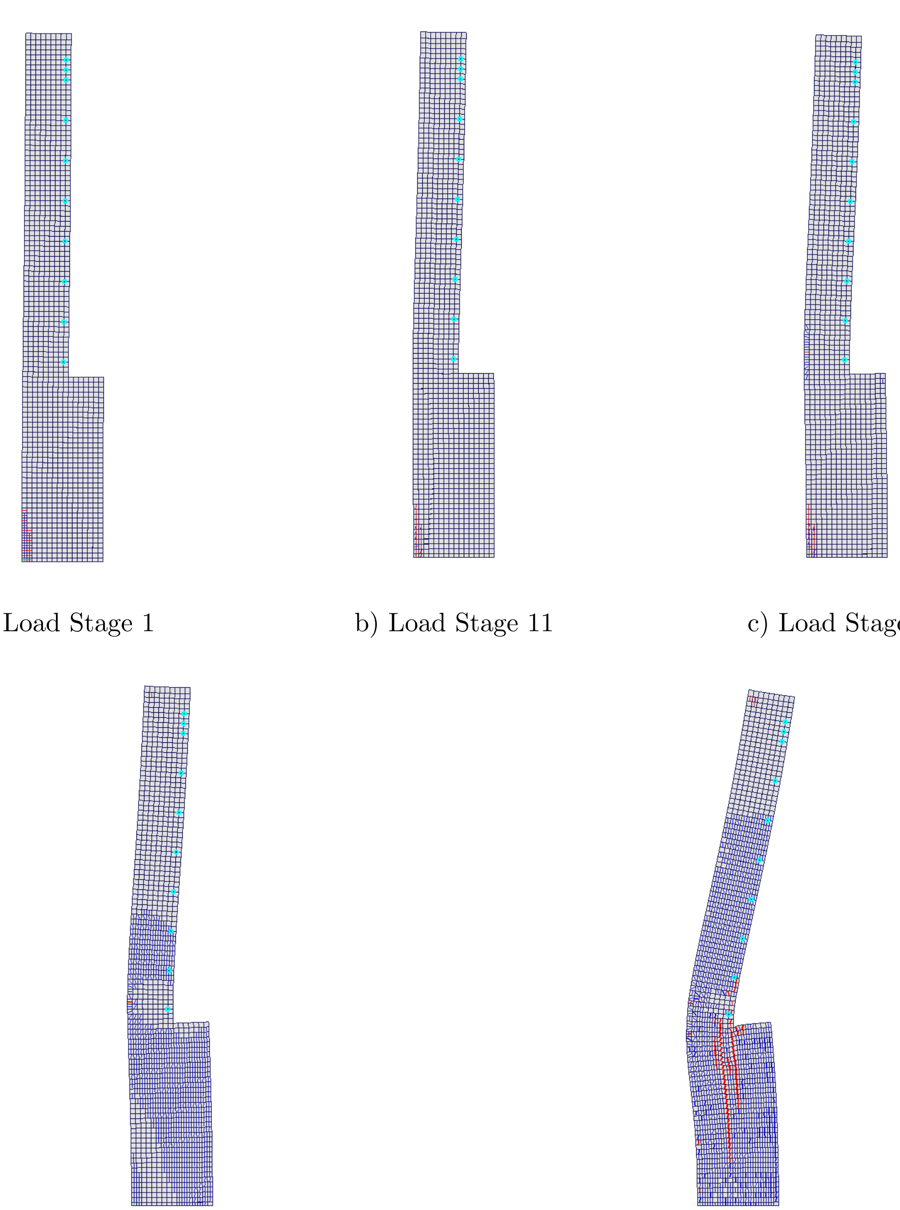
Table 9.19: VecTor6 Example Model RC Material Specifications

Reinf. No.	Colour	Area	D_b	f_y	f_u	E_s	ε_{sh}	ε_u
		mm^2	mm	MPa	MPa	MPa	me	me
1		819	32	414	650	200,000	10	167
2		1638						
3		7371						

Table 9.20: VecTor6 Example Model Discrete Reinforcement Material Specifications

9.7.3.5 Results

Utilizing the numerical data obtained from the analysis output files for model RB2, several of the post-processing features available in Janus for VecTor6 models may be demonstrated. Each of the expanded analysis output files for RB2 are noted to be approximately 1,400 kB in size. For example, see Figure 9.35 for model RB2 depicted in Janus at 0.50 load increment intervals, with Deformations and Crack Pattern modes simultaneously enabled. Additionally, refer to Figure 9.36 for the distribution of in-plane concrete shear strains at each annular element within RB2 at Load Stage 31. Lastly, Figure 9.37 displays a graphic plot of the lateral deflection of the silo wall with respect to the monotonically increasing load factor.



a) Load Stage 1 b) Load Stage 11 c) Load Stage 21

d) Load Stage 31 e) Load Stage 41

Figure 9.35: VecTor6 Example Model Deformations and Crack Pattern

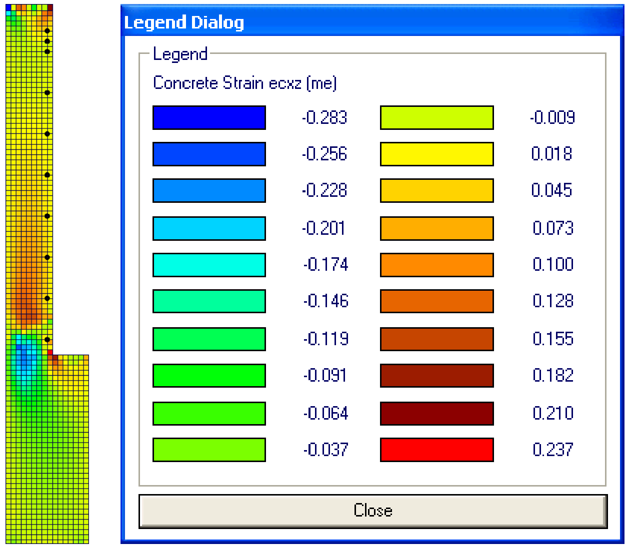


Figure 9.36: VecTor6 Example Model in Shear Strain Contour Mode

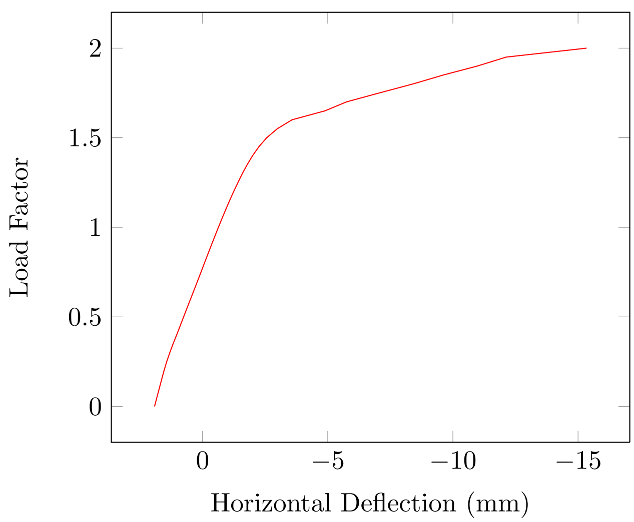


Figure 9.37: VecTor6 Example Model Load-Deflection Response

Chapter 10

Summary and Recommendations

10.1 Summary

The VecTor suite of finite element analysis programs is designed for the purpose of modelling the nonlinear response of reinforced concrete structures under a variety of loading conditions. Due to the complex and comprehensive nature of the generated analysis output data, a versatile post-processor program is necessary in order to assist users in viewing and disseminating the results of their VecTor-based analyses in a user-friendly and intuitive manner. To that end, the post-processor program Janus is developed for the dual purpose of interpreting VecTor output files as well as graphically displaying the resulting structural model and its associated structural performance characteristics.

Janus is a graphical post-processor program developed using the C++ programming language, utilizing the MFC class library and OpenGL API in order to provide versatile structural analysis rendering capabilities through an intuitive and readily implemented user interface. Janus is designed as a standalone executable program, capable of reading analysis output files for all five VecTor programs currently available for use. At its most fundamental level, Janus is designed to display relevant analysis results such as: nodal loads, nodal restraints, structural deformations, and elemental stress-strain parameters. Depending on the analysis capabilities supported by the VecTor program source, additional post-processing result modes and view facilities may be enabled or disabled when viewing a particular VecTor model type in Janus.

For optimal versatility in post-processing capabilities, Janus is integrated with a variety of functions with regards to file loading. Upon opening a VecTor model listed with a significant number of prescribed load stages and/or elements, Janus will present users with an option to load a custom subset of the entire load stage range. As a further measure of memory conservation, users may also opt for Janus to selectively load only the node-related result values.

In order to comprehensively display both two-dimensional and three-dimensional model types of varying geometry and scale, Janus hosts a variety of customizable view features within its capabilities as a post-processing program. For example, depending on the VecTor model type, relevant Section View and/or Layer View features are provided in order to allow users to inspect intermediate sections or internal elements of a structure that may not be accessible from the general exterior view. Additionally, the scale of visual features such as model deformations and crack patterns may be veritably fine-tuned by the user, depending on the contextual presentation of the resulting values. Lastly, a variety of toolbar buttons are provided for toggling various enabled views, result modes and nodal features on and off.

In general, numerical result parameters are capable of being presented in Janus using a variety of display mechanics. Activating the contour mode for a node- or element-related variable correspondingly colours nodes or elements using a colour gradient, with each shade representing a different interval between the local maximum and minimum values found for the current load stage. Element type(s) that are not applicable to a particular contour mode are rendered as simple wireframes or coloured black for visual clarity. The Hotspot mode may be used to further isolate the node(s) and/or element(s) which exhibit a result value which falls within or exceeds a user-specified range. Using number line relationships, model nodes or elements that exceed the specified range are highlighted as red, while ones that fall within the range are highlighted as brown. All other non-applicable elements are omitted or rendered as a wireframe. It is worthy to note that most visual result features such as contour and Hotspot modes are compatible for application with alternative view modes such as Section View and Layer View. Lastly, the data platform may be utilized to pinpoint numerical values for a set range of load stages, specifying up to five independent variables and elements/nodes at a time. Users can collect the data of interest as formatted text, preview the values on-screen as a graphic plot, or even export the information as column-separated data in an external file.

In summary, Janus is a post-processor program that is capable of graphically displaying the analysis results generated by the entire suite of VecTor programs. A comprehensive array of view and mode features are provided in order to present both node- and element-related result variables in an intuitive yet versatile manner. Janus serves as a support program and concluding platform for the VecTor analysis procedure, allowing users to visualize the numerical results they have obtained on a model-wide scale as well as effectively demonstrate them to others.

10.2 Recommendations

Based on the current capabilities and functionality of Janus as a post-processor program, the following list presents additional features and options that could potentially improve the overall performance of the program as well as its ability to display results:

- Currently, user-specified view and mode options in Janus are provided on a transient basis, where enabled features only apply to the model being viewed. Upon opening a new model, all result modes and view variables are reset to default values. Since users may be using the post-processor to repeatedly inspect a local area of interest or a particular subset of response characteristics for a series of similar models, it may be useful to allow users to save and recall their preferred view and result mode settings in Janus.
- The *Element Attributes* dialog may only be invoked by using the right mouse button in Janus' Global Model View, which in itself only presents the exterior-most element faces for viewing purposes. Universally implementing equivalent right mouse click functionality for element/layer attributes within Section View and/or Layer View would significant improve overall program utility, permitting users to view element/layer information at intermediate locations throughout the model.
- At present, structural parameters such as node numbers, element numbers, and nodal coordinates are only accessible on a single-element basis by invoking the *Element Attributes* dialog. Instituting the capability for users to categorically hide or reveal node and/or element attributes on a model-wide scale in Janus should be investigated.
- In its current form, Janus only supports the display of applied nodal force and displacement load cases. Further efforts could be made in order to implement the visualization of alternative node and element-related load types available for assignment in the VecTor programs.
- As previously explained, Janus approximates VecTor4 shell elements as a series of shapes rendered on a piecewise linear basis between all adjoining nodal vertices. In actuality, differential rotations/displacements shell element nodes are intended to represent smoothly curved surfaces in three dimensions as dictated by the quadratic shape functions which govern its behaviour. In the pursuit of accurately representing curved shell element surfaces in Janus, further exploration into modern graphics rendering technology for curved surfaces could be warranted.
- Janus is currently limited to portraying VecTor5 member end forces using the conventional solid colour contour mode in conjunction with the number ranges presented in the *Legend* dialog. Visualization of VecTor5 member forces would be more readily apparent to the user by providing the

option to selectively enable axial, shear and bending moment diagrams superimposed upon the planar frame structure, with local maxima/minima labels provided at appropriate locations.

- To assist users in visualizing VecTor6 axisymmetric models on a planar basis, Janus should be capable of graphically displaying the position of the z-axis in Global Model View. This would serve as a means of denoting the actual location of the rotational axis that VecTor6 models are conceptually expressed to be radially extruded about.
- The development of an integrated electronic help manual and function description system within Janus itself would improve its overall accessibility as a post-processor program, particularly in the event that external supporting documents such as this manual are unavailable for reference.
- In foresight of maintaining compatibility for future implementation purposes, the Janus source code should be upgraded in order to operate without fault in contemporary operating systems such as Windows 7 and Windows 8 as well as 64-bit memory environments.

Appendix A

VecTor3 Example Model Section Views in FormWorks

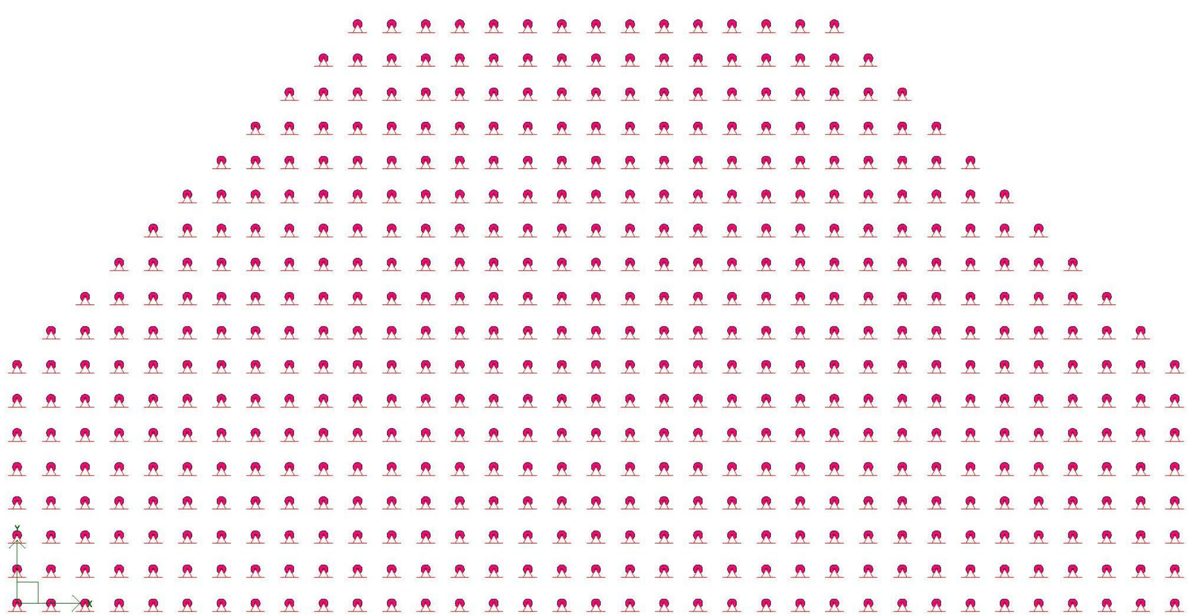


Figure A.1: XY Section View at z = -1000 mm

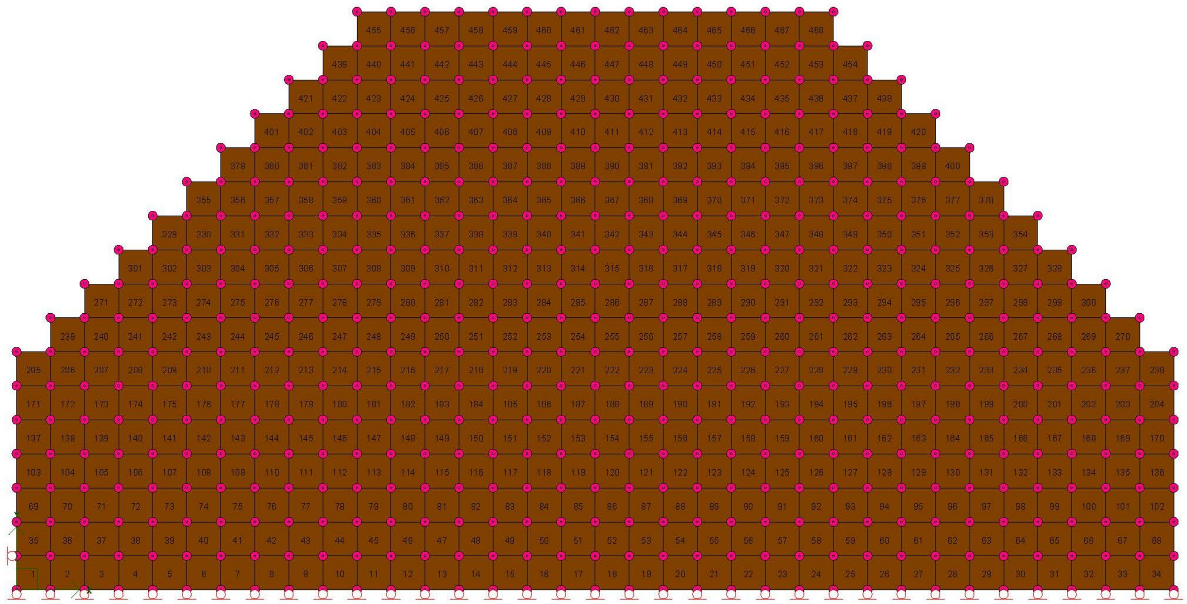


Figure A.2: XY Section View at $z = 0$ mm

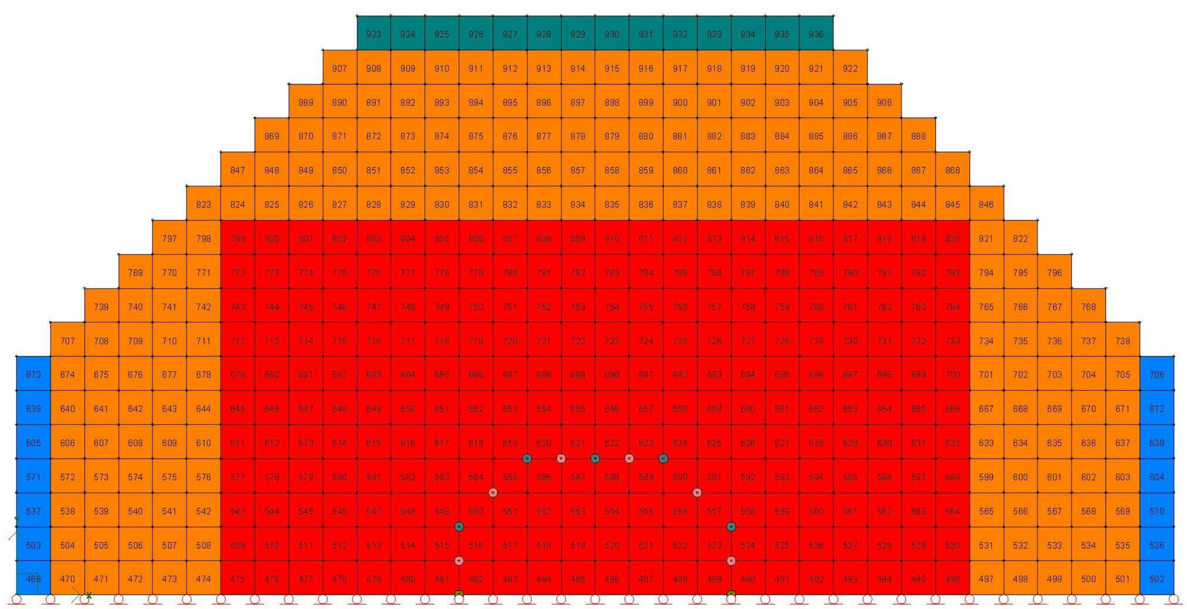


Figure A.3: XY Section View at $z = 400$ mm

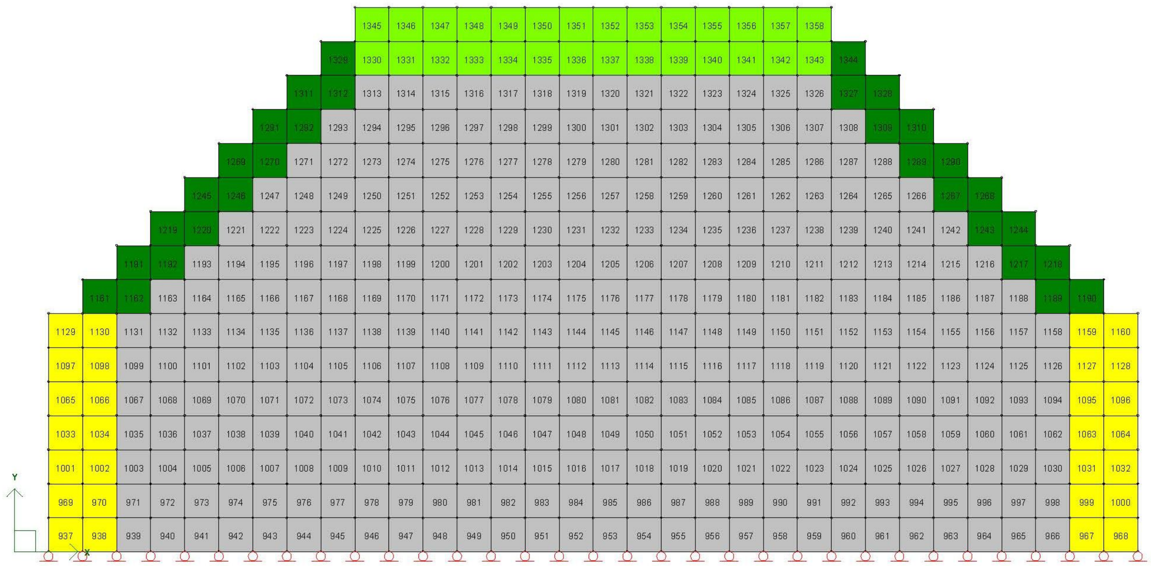


Figure A.4: XY Section View at z = 720 mm

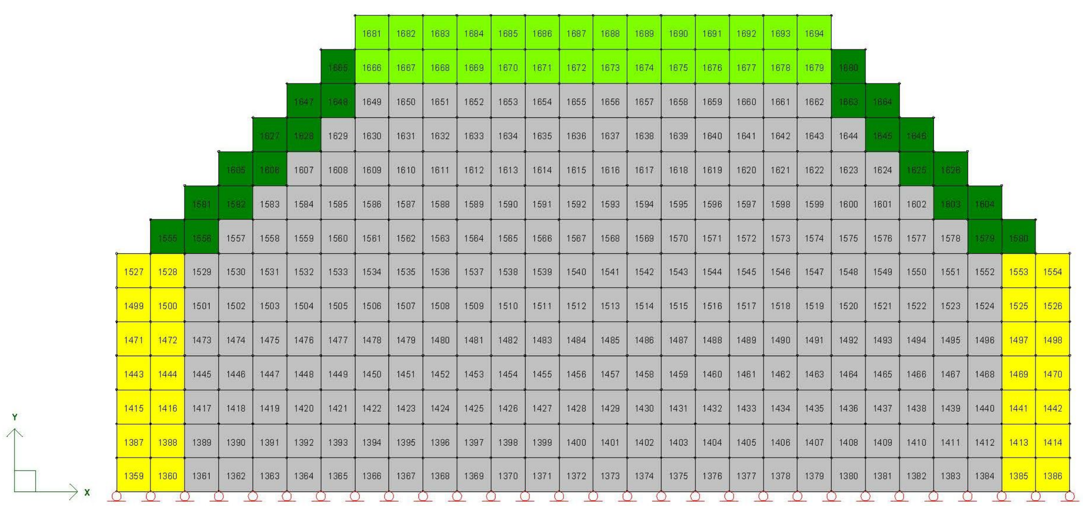


Figure A.5: XY Section View at z = 1040 mm

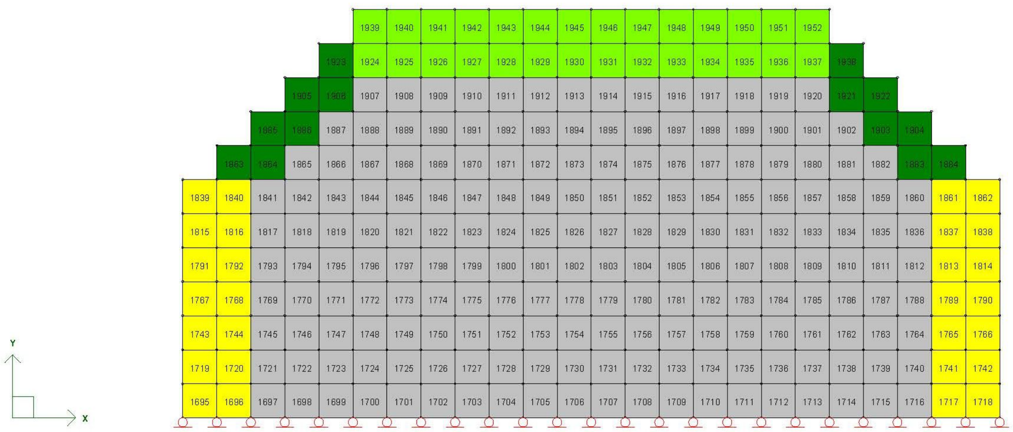


Figure A.6: XY Section View at z = 1360 mm

Figure A.7: XY Section View at z = 1680 mm

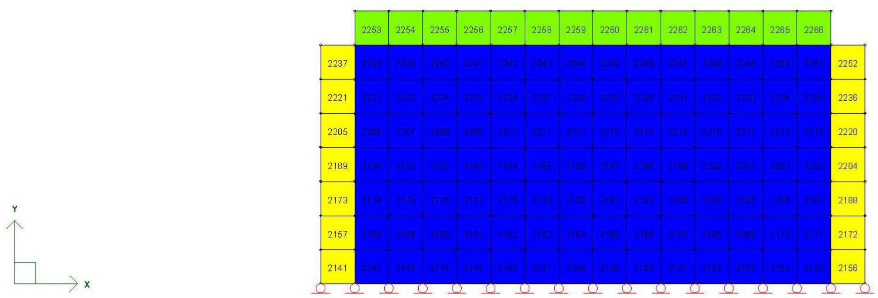


Figure A.8: XY Section View at $z = 2000$ mm

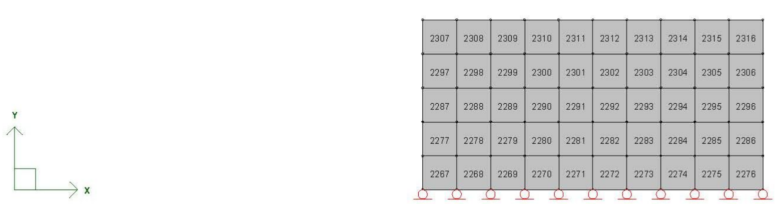


Figure A.9: XY Section View at $z = 2367$ mm

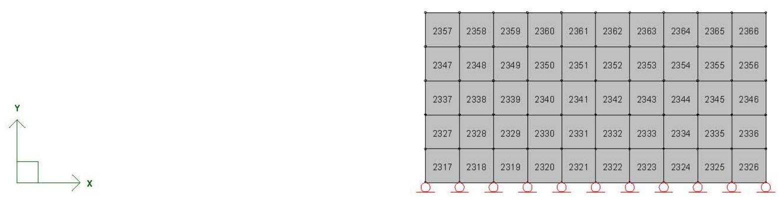


Figure A.10: XY Section View at z = 2734 mm

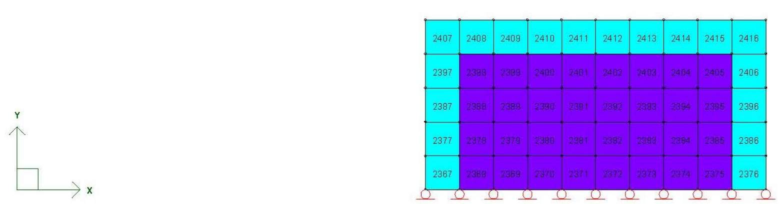


Figure A.11: XY Section View at z = 3100 mm

Figure A.12: XY Section View at z = 3200 mm

Appendix B

VecTor3 Example Model Deformations and Crack Pattern

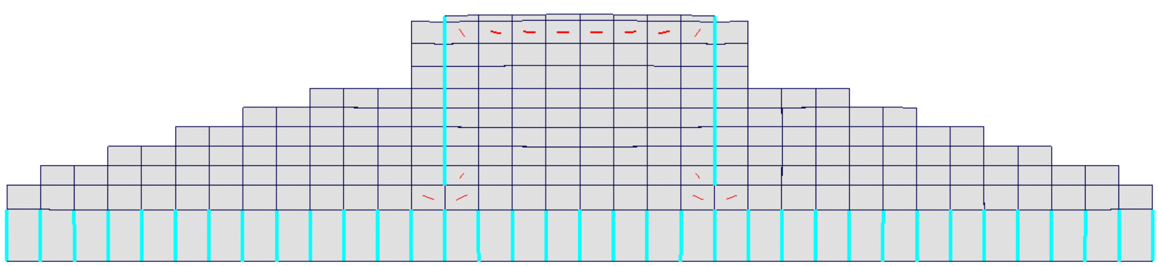


Figure B.1: XZ Plane View at Load Stage 1

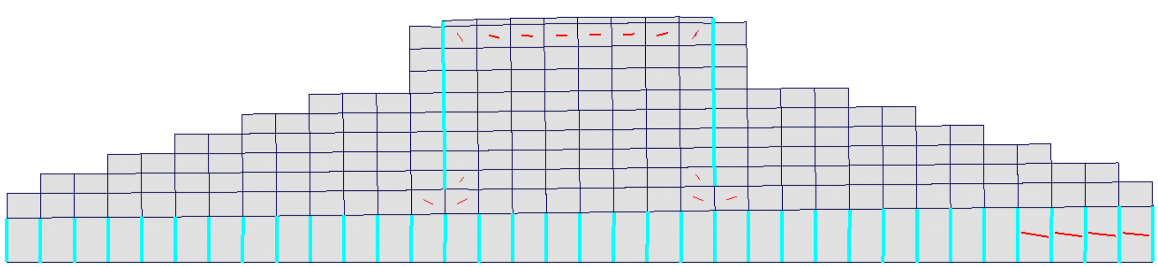


Figure B.2: XZ Plane View at Load Stage 6

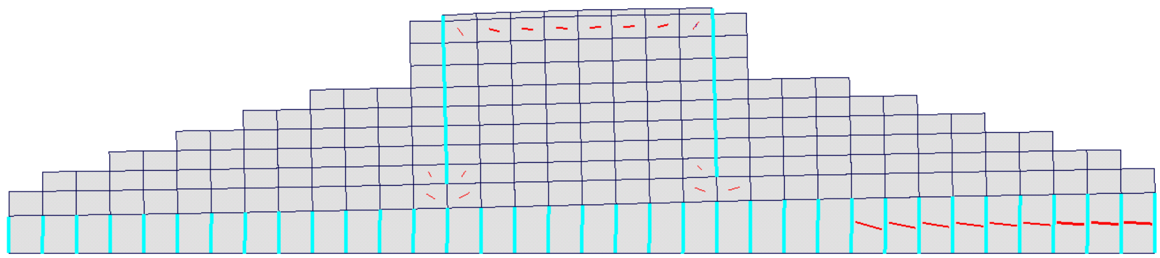


Figure B.3: XZ Plane View at Load Stage 11

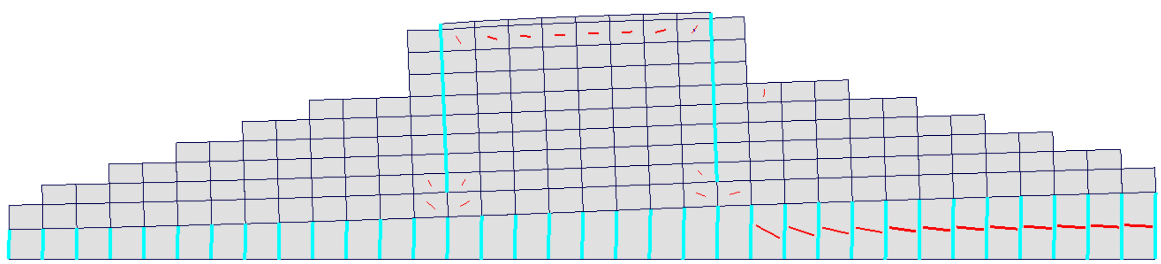


Figure B.4: XZ Plane View at Load Stage 16

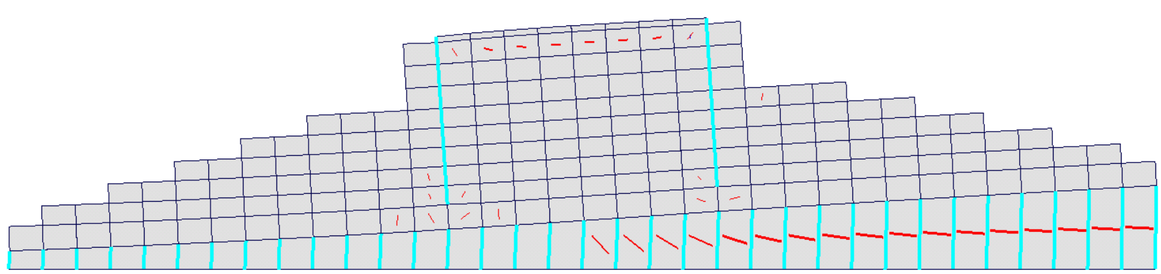


Figure B.5: XZ Plane View at Load Stage 21

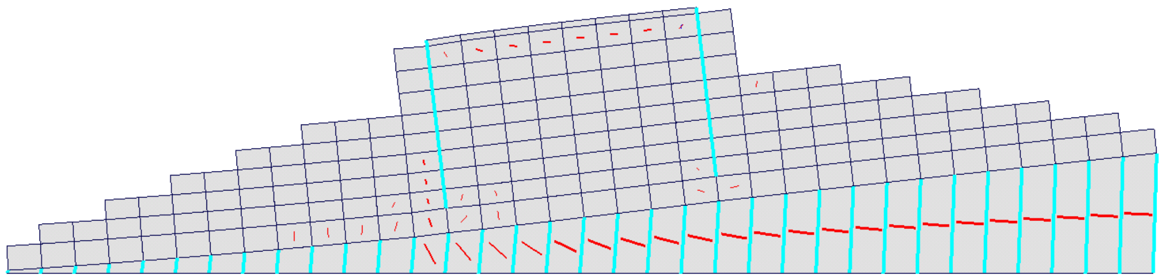


Figure B.6: XZ Plane View at Load Stage 26

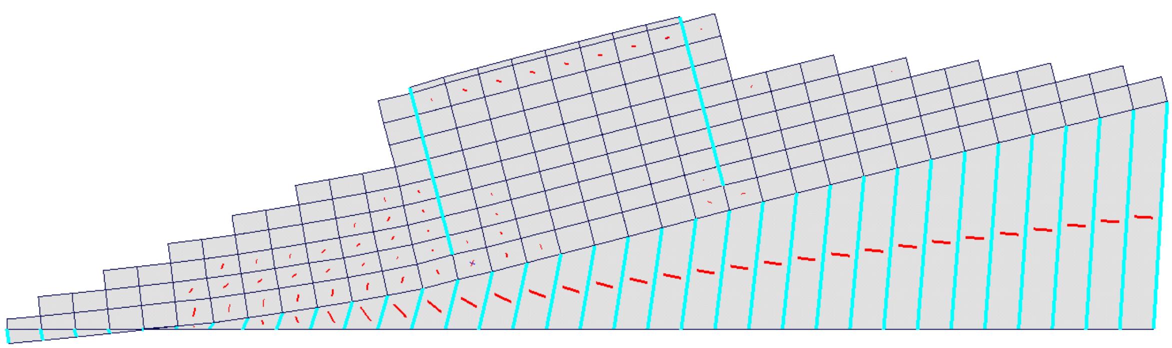


Figure B.7: XZ Plane View at Load Stage 31

Appendix C

VecTor3 Example Model Sectional Crack Pattern and Contour Mode

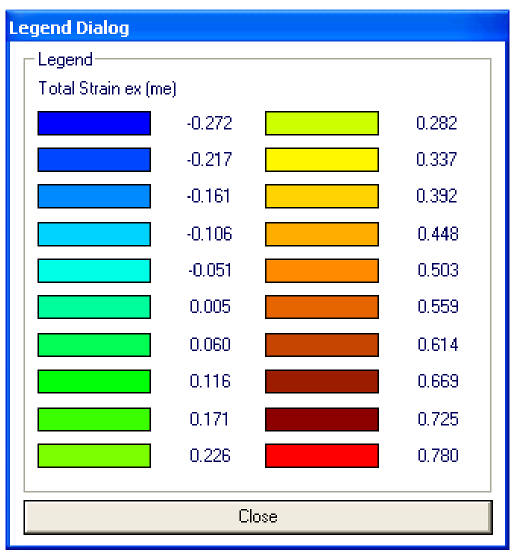


Figure C.1: Legend Dialog for ε_x at Load Stage 26

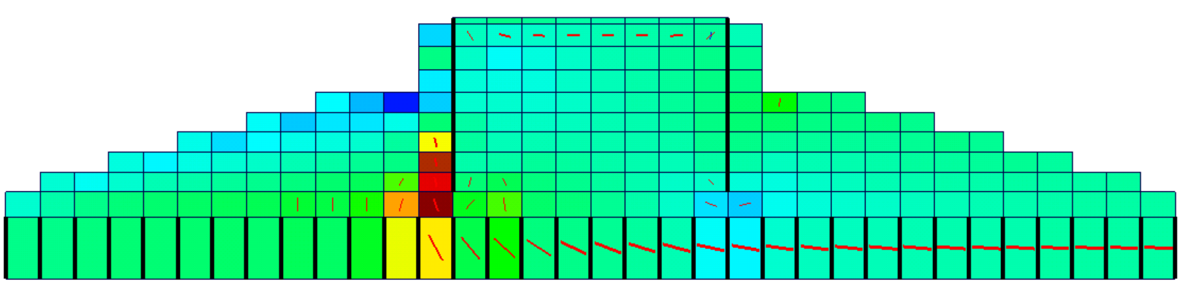


Figure C.2: XZ Section View at $y = 0$ mm

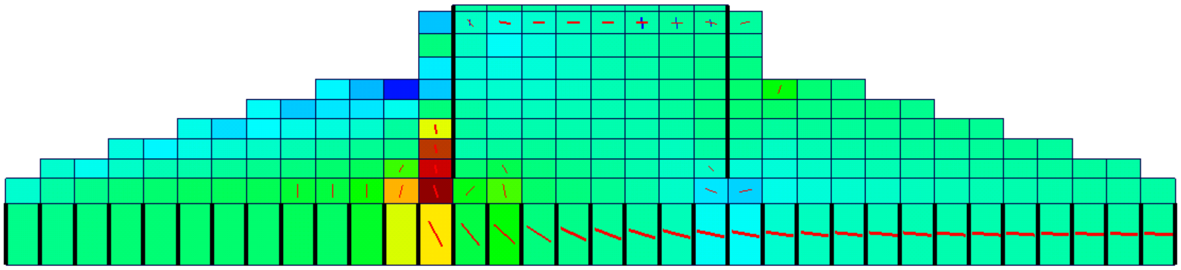


Figure C.3: XZ Section View at y = 1112 mm

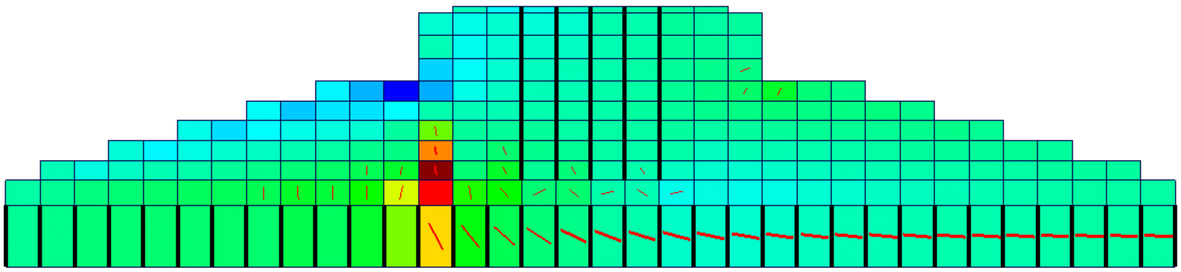


Figure C.4: XZ Section View at y = 2224 mm

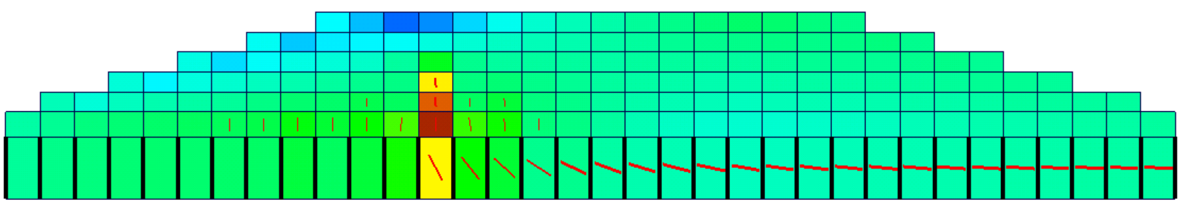


Figure C.5: XZ Section View at y = 3336 mm

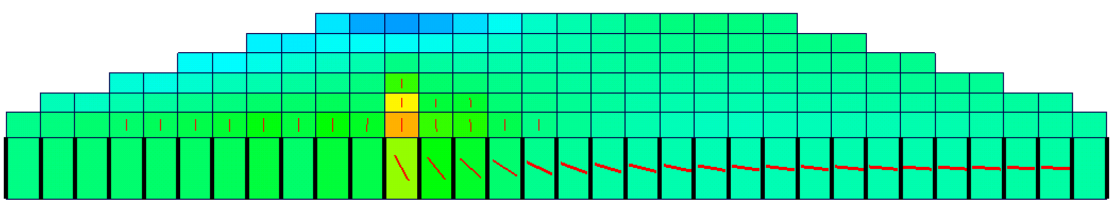


Figure C.6: XZ Section View at y = 4448 mm

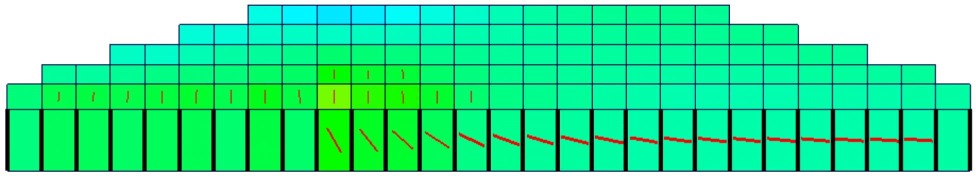


Figure C.7: XZ Section View at y = 5560 mm

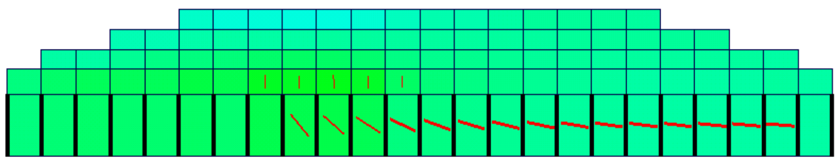


Figure C.8: XZ Section View at y = 6672 mm

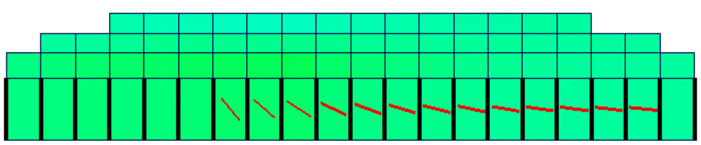


Figure C.9: XZ Section View at y = 7784 mm

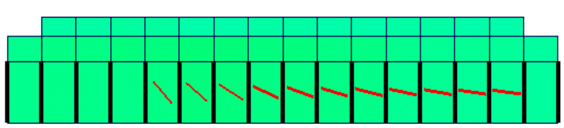
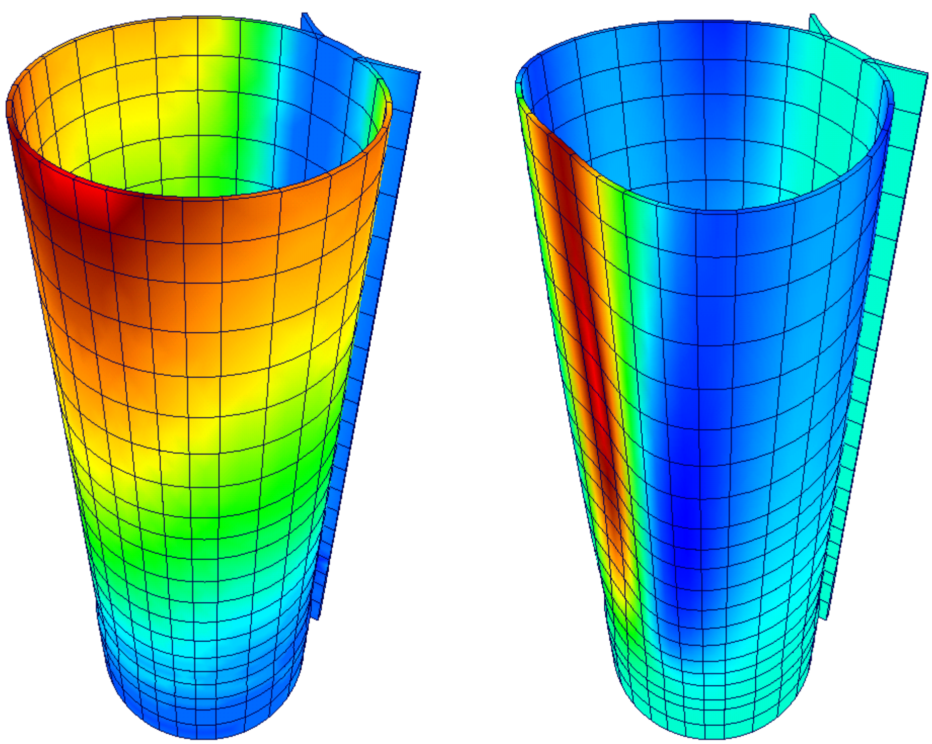


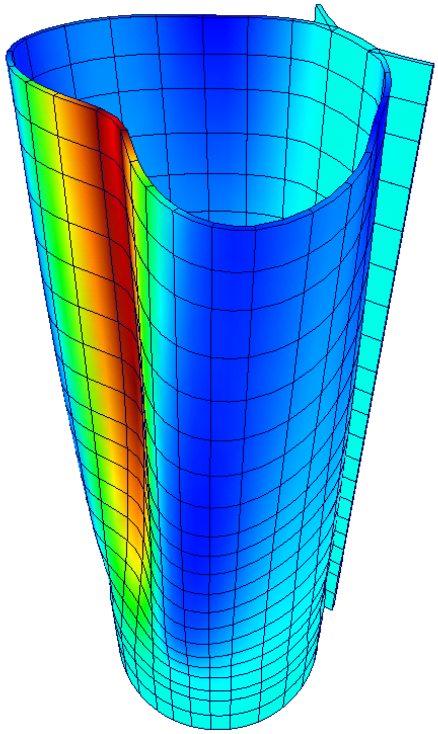
Figure C.10: XZ Section View at y = 8896 mm

Appendix D

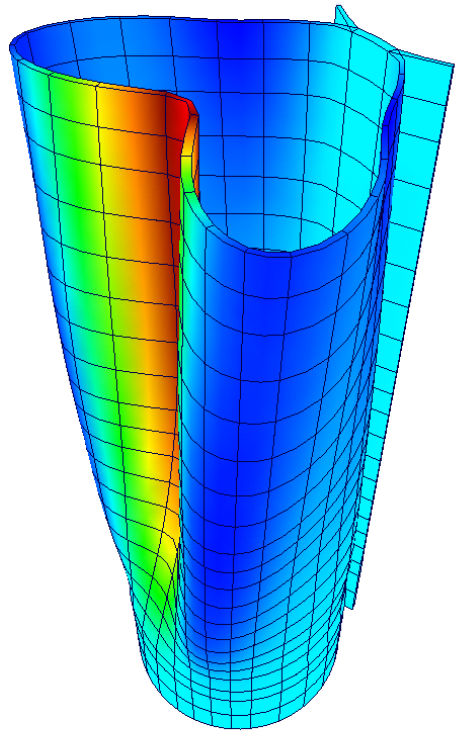
VecTor4 Example Model Deformations and y-direction Displacements



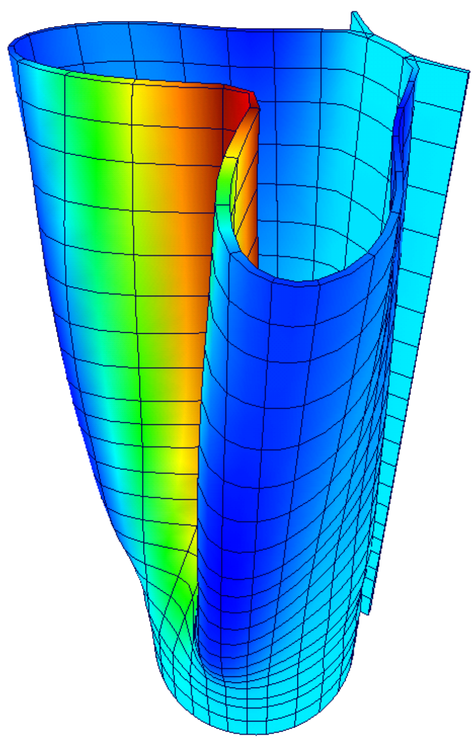
a) Load Stage 1 b) Load Stage 3



c) Load Stage 5



d) Load Stage 7



e) Load Stage 9

Bibliography

Ahn, S. H. (2013). "OpenGl Transformation." http://www.songho.ca/opengl/gl_transform.html. Accessed: 2013-04-08.

Güner, S. (2008). "Performance Assessment of Shear-Critical Reinforced Concrete Plane Frames." Ph.D. thesis, University of Toronto, Toronto, Ontario.

Horton, I. (2008). Ivor Horton's Beginning Visual C++®2008. Wiley Publishing, Inc., 1356 pp.

Huang, H. C. and Hinton, E. (1986). "A New Nine Node Degenerated Shell Element with Enhanced Membrane and Shear Interpolation." International Journal for Numerical Methods in Engineering, vol. 22, pp. 73–92.

Hyrnyk, T. D. (2013). "Behaviour and Modelling of Reinforced Concrete Slabs and Shells Under Static and Dynamic Loads." Ph.D. thesis, University of Toronto, Toronto, Ontario.

Lefas, I. D., Kotsovos, M. D., and Ambraseys, N. N. (1990). "Behavior of Reinforced Concrete Structural Walls: Strength, Deformation Characteristics, and Failure Mechanism." ACI Structural Journal, vol. 87, no. 1, pp. 23–31.

Polak, M. A. and Vecchio, F. J. (1993). "Nonlinear Analysis of Reinforced Concrete Shells." ASCE Journal of Structural Engineering, vol. 119, no. 12, pp. 3439–3462.

Sadeghian, V. (2012). "Formworks-Plus: Improved Pre-Processor for VecTor Analysis Software." Master's thesis, University of Toronto, Toronto, Ontario.

Vecchio, F. J. (1992). "Finite Element Modeling of Concrete Expansion and Confinement." ASCE Journal of Structural Engineering, vol. 118, no. 9, pp. 2390–2406.

Vecchio, F. J. (2000). "Disturbed Stress Field Model for Reinforced Concrete: Formulation." ASCE Journal of Structural Engineering, vol. 126, no. 9, pp. 1070–1077.

Vecchio, F. J. and Collins, M. P. (1986). "The Modified Compression-Field Theory for Reinforced Concrete Elements Subjected to Shear." ACI Journal, vol. 83, no. 2, pp. 219–231.

Vecchio, F. J. and Emara, M. B. (1992). "Shear Deformations in Reinforced Concrete Frames." ACI Structural Journal, vol. 89, no. 1, pp. 46–56.

Wong, P. S. (2002). "User Facilities For 2D Nonlinear Finite Element Analysis of Reinforced Concrete." Master's thesis, University of Toronto, Toronto, Ontario.

Wong, P. S., Trommels, H., and Vecchio, F. J. (2012). VecTor2 and FormWorks User's Manual: Second Edition. University of Toronto, Toronto, Ontario.