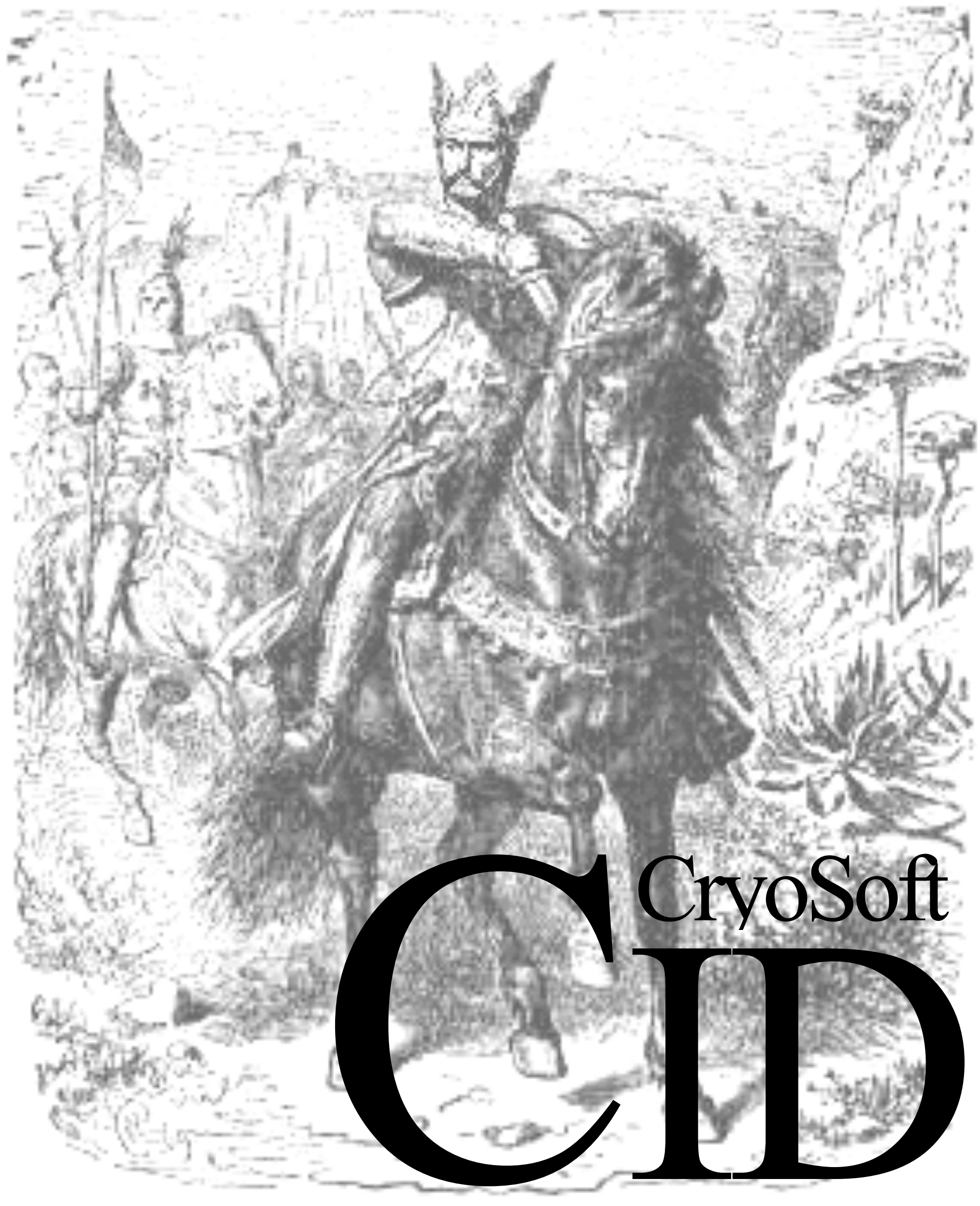
User's Guide

Version 1.0
December 2013



Cable Interactive Designer

DISCLAIMER

Contents

CHAPTER 1

Introduction

What is CID

CID is an event driven Cable Designer, a program with a user friendly Graphical User Interface (GUI) that allows you to simulate and reproduce the processes by which superconducting strands are twisted and compacted to form a high current superconductor. CID simulates the cabling process, starting from a theorical geometry and rearranging strands according to pre-defined geometrical transformations and cable compaction. The resulting cable geometry is then used to calculate electric and magnetic parameters that are necessary for characterization and detailed analysis of the cable properties, such as:

- cross sections of components,
- contact and free perimeters,
- cable contact topology and average conductance among strand couples,
- cable inductance matrix.

In addition to the above calculations, it is possible to define in CID a set of calculation points placed arbitrarily in space around or within the conductor. At these points CID computes the magnetic field and vector potential generated by unit current in each strand, thus allowing subsequent analyses of, e.g., experimental data measured by field or flux sensors.

A CID model

A CID model for a superconductor is defined in an ASCII input file using a free format keywords language described later. The superconductor consists of a cable made by twisted strands and (possibly) a jacket that provides helium tightness and defines the outer geometric boundary of the cable. The cable geometry is defined using a top-bottom breakdown of the pattern of twisted subcable stages starting from the basic elements, the strands, and building the final cable stage by stage in succession. The cable is then inserted in a jacket to form the superconductor. A schematic representation of this process is shown in Figure 1 for the simple case of a cable formed by a quadruplet of triplets of strands (12 strands, cabling pattern 3 x 4) in a round jacket. Each step in the schematic of Fig. 1 corresponds to a building step in the input file.

Note that the shape of the final cabling stage can be selected among several possible variants described later, mimicking the effect of a turck's head in the final compaction stage. Similarly, the jacket geometry can be selected among several possible variants. The strand type and internal composition is defined by the user, and is taken into account in the calculation of the final geometric parameters (e.g. total cross sections of the constituents).

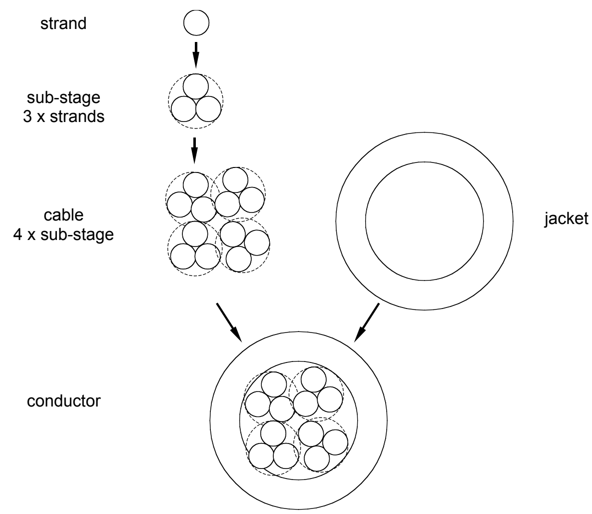


Figure 1. Example of the superconductor build-up from a single strands into a triplet (sub-stage), into the cable of 4 triplets and finally into a conductor in a round jacket.

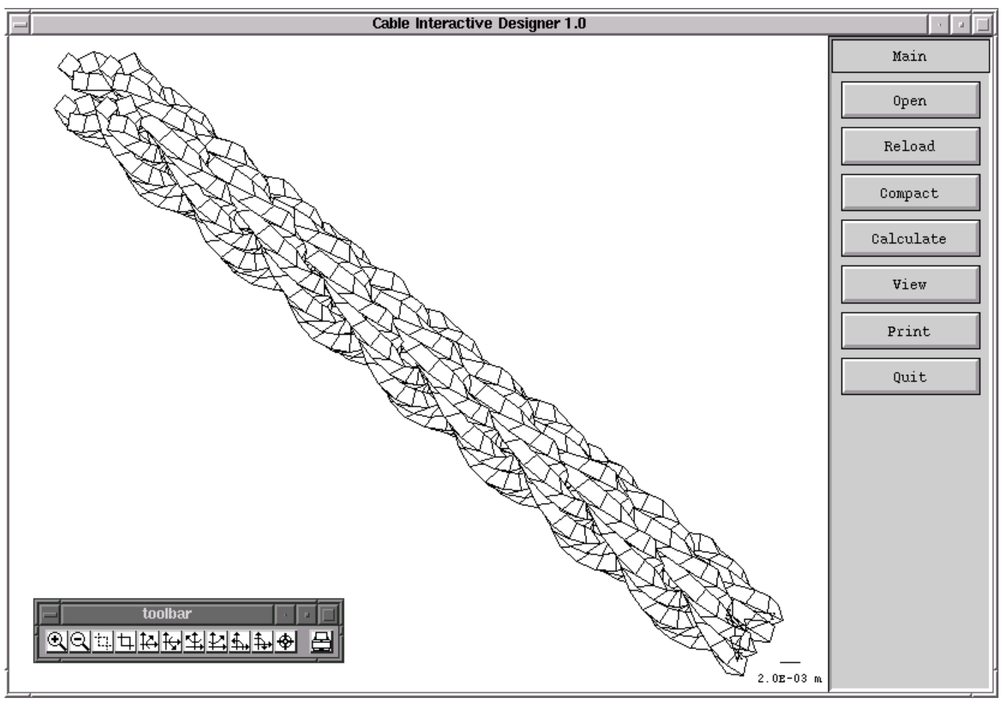


Figure 2. The main window of CID, showing a loaded model and the calculation buttons used to launch the various possible analyses. The CID model corresponds to the simple case of Fig. 1. The jacket is not shown for clarity.

The model is *loaded* in CID through the GUI. CID interprets the model and verifies consistency. Given the cross sectional geometry and the sequence of twist pitches and directions for each substage, CID generates at loading time the centerlines of the cabled strands in 3-D space and meshes this geometry using General Current Elements (GCE's) of hexaedral shape. Each GCE is defined geometrically as an 8-nodes isoparametric elements. The GCE's are the building bricks for the analytical calculation of magnetic field, vector potential and the numerical calculation of inductances, discussed later. An example of a cable geometry realised by CID based on the example of Fig. 1 is shown in Fig. 2 (the jacket is not shown for clarity). Note how each strand is modelled by the assembly of hexaedral bricks.

The syntax of the input file and the details on the CID model are given in Chapter 3 of this manual.

Calculations with CID

Once a model is loaded it is possible perform calculations using the CID GUI. A screen-shot of the main window of CID showing the main menu and a loaded model is reported in Fig. 2. Using the GUI CID allows to perform the following operations and calculation steps:

- a virtual compaction of the cable geometry in a user's defined outer envelope, using an interference and contact force model (the compact button);
- the overall geometry of the cable, including cross sections and exposed or contact perimeters of each component (the geometry button in the calculation menu);
- average conductance matrix among strands, based on an interference and contact model between round strands (the conductance button in the calculation menu);
- inductance matrix among strands (the inductance button in the calculation menu);
- magnetic field generated by a unit current in each strand in a set of calculation points in space (the grid B button in the calculation menu);
- magnetic vector potential generated by a unit current in each strand in a set of calculation points in space (the grid A button in the calculation menu).

All results of above calculations are logged in an ASCII file that can be used for later processing of the results. Finally, the GUI is used to navigate in the model, changing views and display settings.

The functionalities provided by the GUI of CID are described in Chapter 4 of this manual

CHAPTER 2

Installing and Running CID

Platforms

CID is provided as a package developed for running under UNIX or UNIX-like (e.g. Linux) operating system. The reason is that it may require computer intensive calculations, orderly file management and event-driven interactivity. At the time when this manual is written, the platform where CID has been developed is

- Macintosh running MacOS-X (10.8.5 and higher) under XQuartz,(2.7.4) gcc (4.8.1) with gfortran.

The code has been installed and tested on the following platforms:

- IBM-RISC workstations running the AIX-V4 OS;
- Sun-microsystem workstations running the Solaris OS;
- DEC-alpha workstations running OSF1 OS;
- HP workstations running HP-UX OS;
- INTEL PC's running RedHat Linux 6.1 OS.

Although UNIX obeys strict standards, the architecture of the operating and file system may vary from vendor to vendor. It is therefore possible that porting may require minor adaption of code and libraries. Contact us for advice.

In the following sections we assume here that you are running under a UNIX or UNIX-like operating system, and that you are familiar with UNIX commands, directory and file handling. Contact your system administrator for matters regarding UNIX commands and file system.

Although versions of CID have been ported to PC's running the Windows OS (with unix and X11 emultion under CYGWIN), at the time when this manual is written this is not a platform directly supported and part of the instructions provided below may not be directly applicable.

Installation

CID is one of the CryoSoft family of programs. You will have therefore received the CryoSoft package containing CID either as a tar-ball or in pre-installed form. Verify in the CryoSoft installation manual [1] the procedure to be followed for the proper installation of the complete package. The executable codes, `cid` is in the directory `~/CryoSoft/bin/`. You will find the example inputs and post-processing command files in the directory `~/CryoSoft/xample/cid/code1.0/` (the symbol `~/` stands for your home directory)

How to run CID

Start-up To run CID you will need to launch the executable code. In the standard installation on a UNIX system described above CID is launched typing the command:

```
~/CryoSoft/bin/cid
```

Once launched, the interactive window is opened and the program is ready to react to your actions. As a rule, the first thing to do is to load a model, stored in an input file. CID verifies that the input file of the model exists, then starts parsing it, performing at the same time checks on consistency. Once the model is successfully loaded it is displayed using the desired view mode. The GUI will respond to all requests of operations and calculations. If during a session you wish to modify the model and take these modifications into account in the following calculations, this is possible by editing the input file and reloading it with the reload button.

Each run of CID produces a log file containing a report on the case run, run statistics and error messages. The user can control the name of this file; the default file name is `cid.log`.

CHAPTER 3

Model Input Reference

Structure and syntax

The CID model is defined in an input file through a sequence of commands that follow a pre-defined syntax. As the input file is a normal text (ASCII) file, it can be changed with a normal editor, provided that the syntax is respected. CID loads the model in response to the request from the user, after clicking the Load or Reload buttons (see Chapter 4). The input file is opened, and it is read by the input interpreter that parses and analyzes the syntax and the grammar of the various entries. CID extracts the cable topology and geometry and generates the 3-D coordinates of the strands automatically, displaying then, as requested, on the plot window. This section describes the details on the syntax of the input file. For sample input files see Chapter 6.

In general the file contains a series of blocks that are structured as follows:

```
Begin BlockName
        VariableName    value(s)
        VariableName    value(s)

        ………………..
        ………………..

        VariableName    value(s)
End
```

where *BlockName* is a keyword indicating the block type, and must be one of the following valid choices:

`Cable`	define the general characteristics of the cable (last stage)
`Subcable`	define characteristics and properties of subcables (intermediate stages)
`Jacket`	define characteristics and properties of the cable jacket
`Grid`	define characteristics and properties of a grid of points in 3-D space

The content of a block is a series of assignations of a set of values to a generic variable *VariableName*. *VariableName* must be chosen among the set of keywords described in the following sections.

To define a model the input file must contain a `Cable` and a `Subcable` stage. The `Jacket` and `Grid` blocks are optional. Multiple definitions in the same file are possible, and only the last block will be used. To avoid inconsistencies we however advise to avoid redefining blocks.

The structure and content of the input file must comply with the following rules and conventions:

- the identifier of a variable and the corresponding value(s) can appear at any position on the line, they can carry on to several lines and must be separated by blanks or tabs;
- the interpretation is case insensitive;
- abbreviations of the keys are not allowed;
- a character ';' in any position of the command line indicates that the remainder of the line must be considered as a comment. If the ';' is the first character in a line, then the whole line is ignored;
- for an array of variables, the exact number of elements must follow the keyword. The expected number of elements is reported in the description of the variables below. If a keyword or a numeric entry *entry* is repeated N times within an array the alternative implicit syntax `Nx` *`entry`* can be used to shorten the input;
- the variables in the block are read sequentially and are checked at read-in time. For this reason the order of precedence of the variables must be respected whenever a value is needed to proceed with the interpretation of a block;
- repeated variable assignation overrides previous values and is not checked at read-in time;
- the blocks in the file are read sequentially and are checked at read-in time

Input variables reference

The following table contains, in alphabetical order, the keywords defining the input variables, their physical dimensions and meanings for each block type. Predefined possible values are reported in `Courier`. The default value is indicated in the table and underlined.

Note In the tables below we use the following convention for the type of variables:

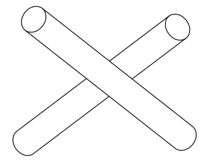
C	character (a string delimited by blanks, tabs or apices)
R	real (a number in floating point or engineering notation)
I	integer (an integer number)

Typing must be respect in the input file to avoid errors or mis-interpretation by the parser.

Cable

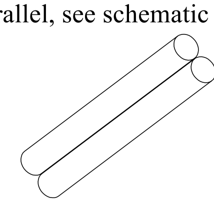
The Cable block describes general characteristics of the last stage of the cable. The final shape of the cable is decided based on which geometric variable is read and defined (a value different from zero) in accordance to the schematic of Figure 3.

Variable	Type	Units	Meaning
`Center`	R	(m)	Array of 3 elements containing the central x,y,z coordinates of the cable. All coordinates of the strands are computed starting from the cable center. The cable is generated in the positive z direction.
`Design`	C	(-)	Composition of the cable, making reference to the subcables defined in the `Subcable` blocks. As an example '3 Sub1 2 Sub2' means that the cable is composed by 3 subcables named Sub1 and by 2 subcables named Sub2. All names must be valid existing subcables, each of them defined within a `Subcable` block. The subcables are placed in accordance with the cable `Type` chosen (see later). For `core twisted` cable types the first subcable in the list will be the central component.
`Diameter`	R	(m)	Diameter of the cable, if `Diameter` is defined the cable is assumed to be round (see schematic in Fig. 3).
`Height`	R	(m)	Height of the cable, if `Height` is defined the cable is assumed to be rectangular (see schematic in Fig. 3).
`LeftHeight`	R	(m)	Left height of the cable, if Left`Height` is defined the cable is assumed to be rectangular and deformed in a keystone shape (see schematic in Fig. 3). In this case also the variable `RightHeight` is needed.
`Length`	R	(m)	Total length of the cable.
`Mesh`	I	(-)	Number of subdivisions in GCE's used to mesh a single strand along the length. The total number of elements in the model will then be to the number of GCE per strand times th number of strands.
`Name`	C	(-)	Cable name.
`Pitch`	R	(m)	Twist pitch of the cable.
`RCross`	R	(Ω)	Interstrand resistance for a single contact among strands of *crossing* type (i.e. axes not parallel, see schematic below).



The value of `RCross` is assigned to all cross contacts found in the calculation of the conductance matrix.

`RightHeight`	R	(m)	Right height of the cable, if Right`Height` is defined the cable is assumed to be rectangular and deformed in a keystone shape (see schematic in Fig. 3). In this case also the variable `LeftHeight` is needed.
`RLine`	R	(Ω m)	Interstrand resistance per unit length for a single contact among strands of *adjacent* type (i.e. axes parallel, see schematic below).



The value of `RLine` is assigned to all line contacts found in the calculation of the conductance matrix.

`Type`	C	(-)	Cable type identifier. Possible values: `twisted` the cable is formed by simple twisting of all subcables defined in the `Design`. `core twisted` the cable is formed twisting all subcables around the first subcable defined in the `Design`.
`S/Z`	C	(-)	Flag describing the twisting direction of the cable. Possible values: `S` the cable is twisted in "S" direction `Z` the cable is twisted in "Z" direction.
`Width`	R	(m)	Width of the cable, if Width is defined the cable is assumed to be rectangular (see schematic in Fig. 3).

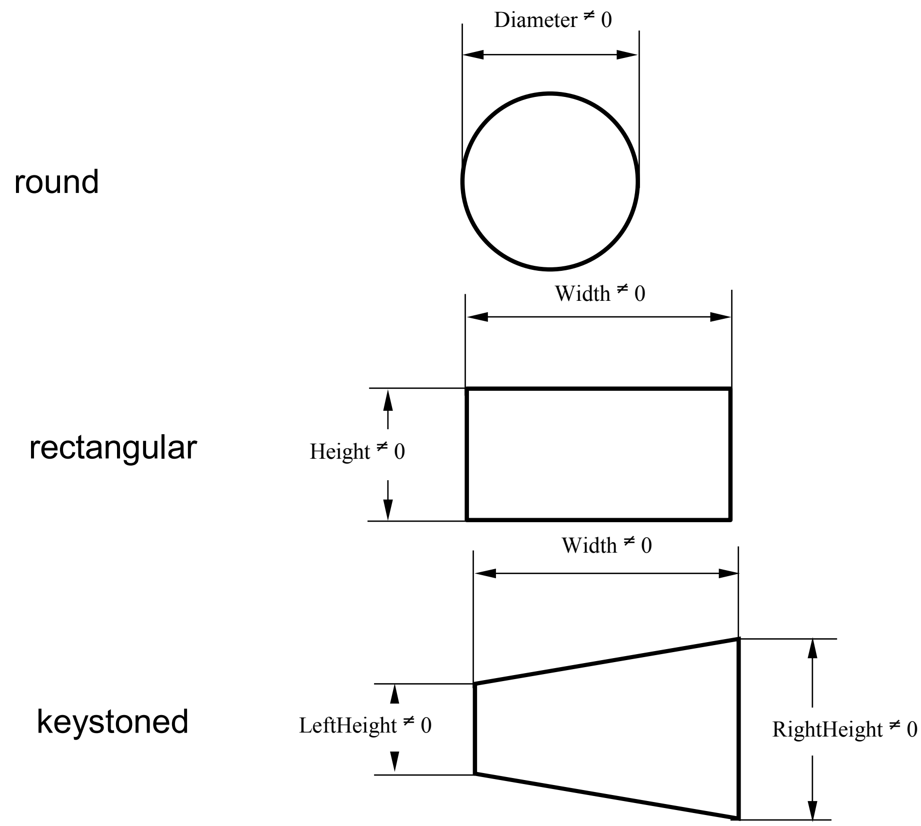


Figure 3. Final geometry of the cable as a function of the variables set in the input file.

Subcable

The Subcable block describes general characteristics of all the subcable stages appearing eventually in the definition of the main cable. The definition of subcables is hierarchical

Variable	Type	Units	Meaning
`Design`	C	(-)	Composition of the subcable, making reference to other subcables defined in the `Subcable` blocks. The same conventions are used as for the `Cable` block.
`Diameter`	R	(m)	Diameter of the subcable. Subcables are always assumed to be round.
`Name`	C	(-)	Subcable name. The name is used for referencing the subcable.
`Pitch`	R	(m)	Twist pitch of the subcable.
`S/Z`	C	(-)	Flag describing the twisting direction of the subcable. Possible values: `S` the cable is twisted in “S” direction `Z` the cable is twisted in “Z” direction.
`Type`	C	(-)	Subcable type identifier. Possible values: `strand` the subcable is a simple strand with given diameter. `twisted` the subcable is formed by simple twisting of all subcables defined in the `Design`.

Jacket

The Jacket block describes general characteristics of a jacket enclosing the cable.

Variable	Type	Units	Meaning
`CurvatureRadius`	R	(m)	Curvature radius of a jacket of type `Squared` or `CircleinSquare`.
`Diameter`	R	(m)	Outer diameter of a jacket of type `Circular` or inner diameter of a jacket of type `CircleinSquare`.
`Height`	R	(m)	Outer height of a jacket of type `Squared` or `CircleinSquare`.
`Name`	C	(-)	Jacket name. The name is used for referencing the jacket.
`Thickness`	R	(m)	Thickness of the jacket of type `Squared` or `Circular`.
`Type`	C	(-)	Jacket type identifier. Possible values: `Circular` the jacket is a circular jacket with given outer diameter and thickness. See schematic in Figure 4. `CircleinSquare` the jacket is an outer square jacket with given width and height surrounding a circular jacket with given diameter. See schematic in Figure 4. `Squared` the jacket is a square jacket with given width, height and thickness. See schematic in Figure 4.
`Width`	R	(m)	Outer width of a jacket of type `Squared` or `CircleinSquare`.

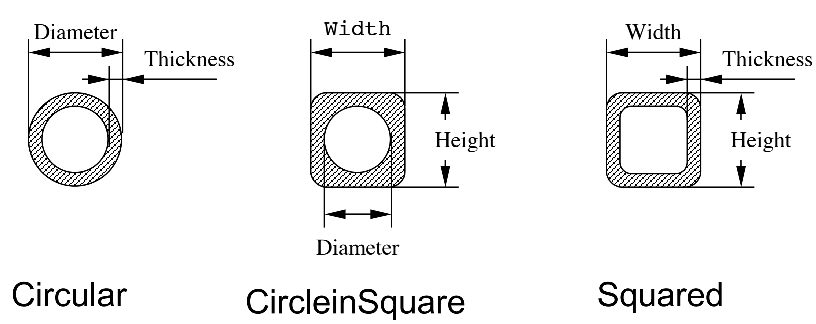


Figure 4. Jacket type definitions.

Grid

The Grid block describes general characteristics of a grid of points in 3-D space. The points are used to perform field and vector potential calculations.

Variable	Type	Units	Meaning
`Center`	R	(m)	Array of 3 elements containing the central x,y,z coordinates of the grid.
`dx`	R	(m)	x dimension of a cartesian grid.
`dy`	R	(m)	y dimension of a cartesian grid.
`dz`	R	(m)	z dimension of a cartesian or cylindrical grid.
`Mesh`	I	(-)	Array of 3 elements containing the number of points along the three dimensions of the grid. The order of the subdivisions are x,y,z for a cartesian grid or R,Theta,z for a cylindrical grid. If an entry is 0 (i.e. no subdivision) the grid has zero dimensions in the corresponding coordinate. This can be used to generate 1-D and 2-D sets of points.
`Name`	C	(-)	Grid name.
`Rin`	R	(m)	Inner radius of a cylindrical grid.
`Rout`	R	(m)	Outer radius of a cylindrical grid.
`Theta1`	R	(deg)	Starting angle of a cylindrical grid.
`Theta2`	R	(deg)	Final angle of a cylindrical grid.
`Type`	C	(-)	Grid type identifier. Possible values: `Cartesian` cartesian grid. The general case is of a 3-D cube with sides parallel to the cartesian axes if all subdivisions are defined in the `Mesh`. This degenerates to a plane or a line in case that one or two subdivisions are zero. See schematic in Figure 5. `Cylindrical` cylindrical grid. The general case is of a 3-D cylinder oriented along the z axis if all subdivisions are defined in the `Mesh`. This degenerates to a surface or a line in case of no subdivision along one or two of the three directions. See schematic in Figure 6.

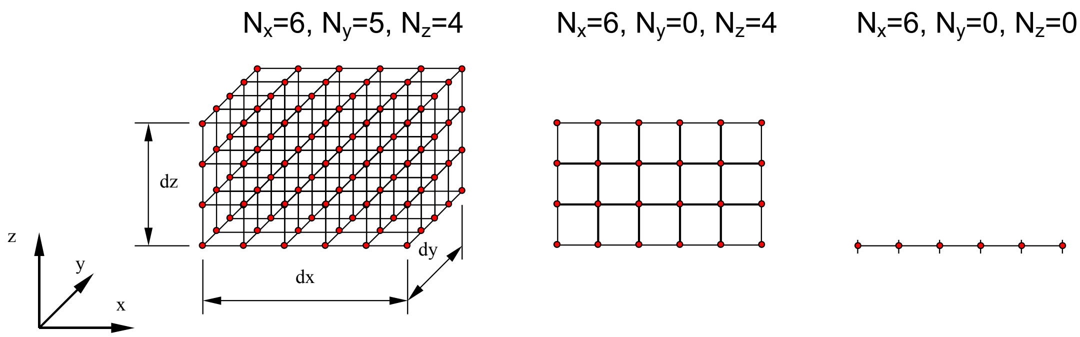


Figure 5. Cartesian grid definition in the general case of subdivisions in all cartesian directions, as well as in the case of no subdivision along one or two cartesian coordinates.

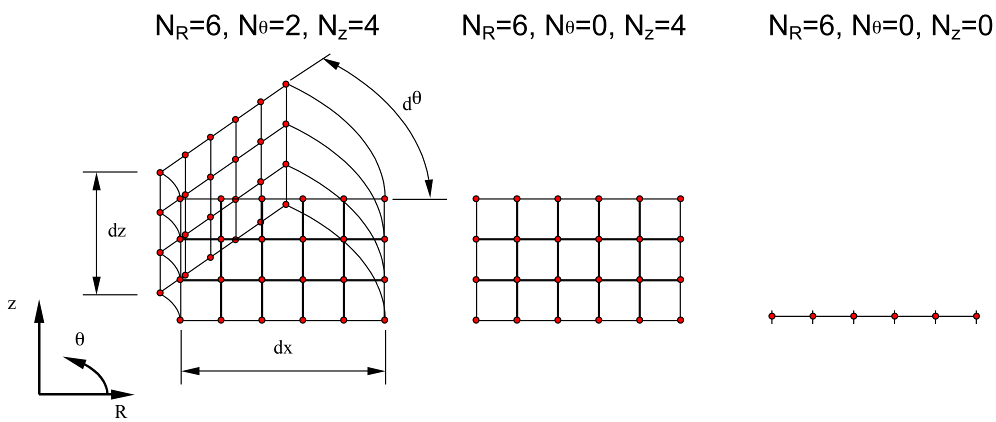


Figure 6. Cylindrical grid definition in the general case of subdivisions in all directions, as well as in the case of no subdivision along one or two cylindrical coordinates.

CHAPTER 4

GUI Menus and Tools Reference

Menus reference

CID has an event-driven GUI (Graphical User Interface) that allows to perform all operations, from model loading, navigation to calculation and plots. The actions are triggered by clicks on the menu buttons, always appearing on the right hand side of the GUI window. Submenus (view and calculations) are also accessed by clicking on the buttons of the main menu. The Back button in submenus is used to return to the upper level.

An overview of the menu tree structure is shown in Fig. 7.

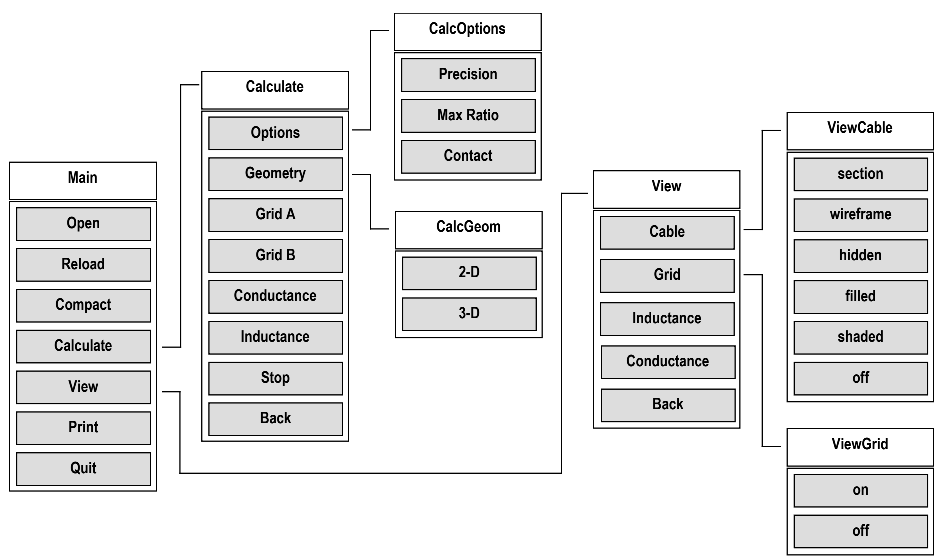


Figure 7. Overview of the menu tree of CID.

Menu buttons can be active or inactive, depending on whether a model has been loaded, and whether a grid for field calculation is defined. The sections below give the correspondance between the menu buttons, tool buttons, accelerators and the actions performed by the program.

Main

The **Main** menu appears first at launching the program and is used to load a model, print the graphic page and communication with all other submenus. The following table describes the buttons in the **Main** menu and the associated actions.

Button	Action
Open	Open the input file containing the model. The input file is selected through a dialog box.
Reload	Reload the current input file, active only if a file is already open.
Compact	Perform cable compaction. This option is inactive in the current release.
Calculate	Open the **Calculate** menu, active only if a cable has been loaded.
View	Open the **View** menu.
Print	Print the contents of the graphics page.
Quit	Quit the program.

Calculate

The **Calculate** menu is used to launch all calculations and analyses.

Button	Action
Options	Set the calculation options. The **CalcOptions** pop-up menu is opened to set the following options: **Precision** — set the relative precision for the calculation of the inductance coefficients in the inductance matrix. A high precision will strongly affect the calculation time. Default setting 1 %. **Max Ratio** — set the ratio of distance among GCE's to the typical GCE dimension for simplified analytical solution of the vector potential. A large value of Max Ratio implies exact solution up to far disctances. Default setting 1000. **Contact** — set the tolerance factor for detecting contact among strands. Default setting 1.
Geometry	Compute the strand geometry and print it to the log file. Two types of calculations are possible: **2-D** — compute and output the total split among the various cross sections in the 2-D plane selected, contact perimeters and free perimeters. **3-D** — compute and output the coordinates of the centers of the strands forming the cable in 3-D space.
Grid A	Compute the vector potential generated by a unit current in each strand on each point of the grid, and print the resulting matrix to the log file. Active only if a grid has been defined.
Grid B	Compute the magnetic field vector generated by a unit current in each strand on each point of the grid, and print the resulting matrix to the log file. Active only if a grid has been defined.
Conductance	Compute the interstrand conductance matrix of the cable and print it to the log file. The matrix is automatically plotted at the end of the computation.
Inductance	Compute the interstrand inductance matrix of the cable and print it to the log file. The matrix is automatically plotted at the end of the computation.
Stop	Stop the computation presently running, all results computed will be lost. The **Stop** button can be used to abort long calculations and reset parameters without killing the program.
Back	Back to the **Main** menu.

View

The **View** menu is used to set viewing options for the cable and the grid.

Button	Action	
Cable	Set the cable viewing options. The **ViewCable** pop-up menu is opened to set the following options:	
	section	2-D view of the cable in the xy plane, shown additionally to the 3-D view. To change the z coordinate of the view press the section area and when the cursor has changed to  on press the whished point along the cable. This feature can be activated only if the cable is in a vertical position, so that z coordinates can be clearly set.
	wireframe	the cable is plotted in 3-D in wireframe mode, transparent view (least time consuming of all plot modes).
	hidden	the cable is plotted in 3-D in hidden-line mode.
	filled	the cable is plotted in 3-D in hidden-line mode with filled surfaces, flat colors.
	shaded	the cable is plotted in 3-D in hidden-line mode with filled surfaces, artificial light.
	off	no cable plot in 3-D space.
Grid	Set the grid viewing options. The **ViewGrid** pop-up menu is opened to set the following options:	
	on	plot the grid points in 3-D space.
	off	do not plot the grid points in 3-D space.
Inductance	Draw the computed conductance matrix as a color map.	
Conductance	Draw the computed inductance matrix as a color map.	
Back	Back to the **Main** menu.	

Tools reference

A toolbar can be used for graphics operations (zoom, fitting of the image, rotations). The toolbar is always in front of the graphic window. The CID toolbar is shown in Fig. 8.

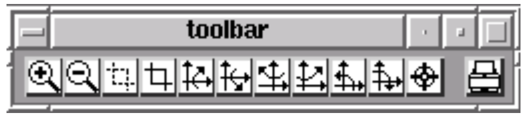


Figure 8. Toolbar of CID.

The following table describes the actions associated with each button in the toolbar.

Tool	Action
	Zoom in in steps.
	Zoom out in steps.
	Fit the plot on the screen. The plot scaling is modified to fit all objects to the graphic window.
	Select an area for plotting. The mouse pointer is changed to cross. Click on the first extreme of the area (e.g. upper left corner) and select the area moving the mouse while holding it down (e.g. to lower right corner). A red frame associated with the mouse position indicates the area selected. Autoscaling is performed to maintain the correspondance of the x and y dimensions.
	Move the view point upward, as like moving along a meridian on a sphere.
	Move the view point downward, as like moving along a meridian on a sphere.
	Move the view point rightward, as like moving along a parallel on a sphere.
	Move the view point leftward, as like moving along a parallel on a sphere.
	Lean the view point leftward without changing its position on a sphere, as like inclining the head to the left shoulder.
	Lean the view point rightward without changing its position on a sphere, as like inclining the head to the right shoulder.
	Center the plot to the selected point on the screen. The mouse pointer is changed to target. Click on the central point of the plot as wished.
	Print the plot currently on the screen.

Key accellerators

Accelerator keys are available for most common operations (opening of a model, navigation, printing, quitting). The following table describes the accelerator implemented and the associated actions.

Key	Action
o or O	Open a file.
p or P	Print the plot currently on the screen.
q or Q	Quit the program.
r or R	Reload.
+	Zoom in in steps.
-	Zoom out in steps.
Arrow Down	Move the plot downwards in steps.
Arrow Left	Move the plot leftwards in steps.
Arrow Right	Move the plot rightwards in steps.
Arrow Up	Move the plot upwards in steps.

CHAPTER 5

Output File Reference

Generic information on the model loaded and all the calculation results are reported to the output ASCII file Cid.log. The file is generated and updated automatically whenever one of the following actions is performed:

Open: as soon as a new model is opened, and extended description of the input file and of the generated objects and strands is written to the log file. This report is updated for every new file opened or reloaded during a session.

Geometry: a 2-D calculation of the cable geometry generates a report containing the overall cross sections of the different strands present in the cable, including overlapping of strands. The total ideal (no overlap) and real (taking into account overlaps) area and perimeters are output to the log file, as well as the empty area inside the cable (void fraction). A 3-D geometry calculation causes the output of the 3-D centers of all GCE's forming a strand. This output is generated for each strand.

Grid A: a vector potential calculation triggers the output of the x,y,z components of the vector potential [T m] generated by each strand (unit current) in the mesh points of the grid.

Grid B: a magnetic field calculation triggers the output of the x,y,z components of the magnetic field potential [T] generated by each strand (unit current) in the mesh points of the grid.

Conductance: the result of a conductance calculation is the number of line and cross contacts found among all strands and their topology as a matrix of (NrOfStrands,NrOfStrands) elements. The interstrand conductance matrix computed using the Rline and Rcross values, with dimension (NrOfStrands,NrOfStrands), is output in units of [Siemens/m].

Inductance: the result of the inductance matrix calculation is a matrix of (NrOfStrands,NrOfStrands) elements, output in units of [H/m].

CHAPTER 6

Worked Examples

A 12 strands cable-in-conduit conductor

As an example of application we consider a 12 strands cable-in-conduit conductor. The cable has the characteristics reported in Tab. 1, and is similar to the first two stages used for the ITER-CS1 conductor. The cable geometry is essentially identical to the one reported in Fig. 1. In the following sections we will detail the input file definition to describe the geometry, and show how to perform the interstrand conductance and inductance calculation, as well as a calculation of the magnetic field influence matrix on a grid of points placed in the middle of the cable length.

Strand diameter	(mm)	0.81
Cabling pattern		3x4x4
1st stage (triplet)		
outer diameter	(mm)	1.75
twist pitch	(mm)	25
2nd stage (quadruplet)		
outer diameter	(mm)	4.21
twist pitch	(mm)	54
3sd stage (quadruplet)		
outer diameter	(mm)	10.17
twist pitch	(mm)	95
G_{cross}	(MS)	1
g_{line}	(MS/m)	20

Table 1. Geometrical and electrical characteristics used for the calculation of the electrical parameters of the first cabling stages of the ITER-CS1 cable.

Input file

The input file corresponding to the geometry of Tab. 1 is reported below. Comments are reported in italics in the file.

manual.input

```
Begin Cable

   name      'CS1 stages 1 and 2'
```

The cableis defined as a twisted assembly of 4 subcables of type `triplet`. *The* `triplet` *subcable is defined later in the file as the assembly of 3 strands, resulting in the desired 3x4 cabling pattern*

```
   type      'twisted'
   design     4 'triplet'
```

The outer diameter of the cable is computed so that it fits exactly 4 equally spaced subcables of type `triplet`

```
   diameter   4.2135438e-3
   pitch      54e-3
   S/Z       'S'
   center     0.0 0.0 0.0
```

The total length modelled is 100 mm, i.e. approximately 2 twist pitches. This length is subdivided in 50 GCE's. The total number of GCE's in the model will then be 50 x 3 x 4 = 600

```
   length     100e-3
   mesh       50
```

Line and cross resistances are input in units of [Ω m] and [Ω] respectively

```
   RLine      0.5e-7
   RCross     1.0e-6

End


Begin Subcable
```

The subcable is defined as the twisted assembly of 3 strands of type `S1`. *Note that the subcable name is used for referencing it at a higher level*

```
   name     'triplet'
   type     'twisted'
   design    3 'S1'
```

The diameter of the triplet is such that it fits exactly 3 adjacent strands of type `S1`

```
   diameter  1.745307e-3


   pitch     25e-3
   S/Z      'Z'

End

Begin Subcable

   name     'S1'
```

The strand type is the basic building block for substages and cables

```
   type     'strand'
   diameter  0.81e-3

End


Begin Grid

   name     'plane'
```

The grid is defined in cylindrical coordinates, with center placed along the z-axis at 50 mm, i.e. at mid cable height

```
   type     'cylindrical'
   center    0.0 0.0 50.0e-3
```

The dimensions of the grid are defined only in R and Theta, and the grid has no extension in z direction

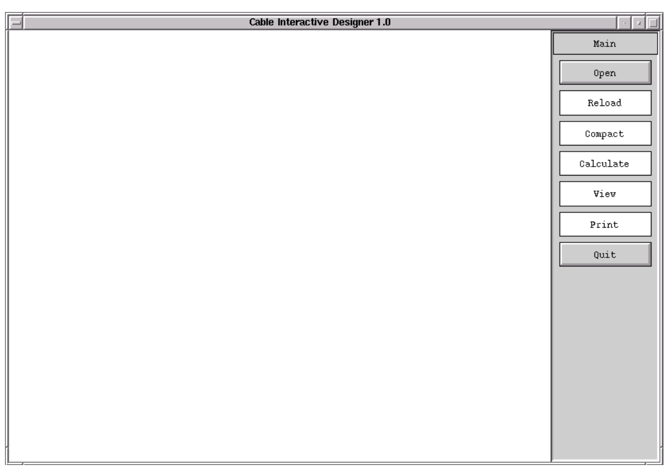
```
   Rin       5.0e-3
   Rout     10.0e-3
   Theta1    0.0
   Theta2   90.0
   dz      100.0e-3
```

The number of subdivisions is 10 in R direction, 10 in Theta direction and no subdivision in z direction. This results in a plane grid that has only R and Theta extension. See also the plots reported later on

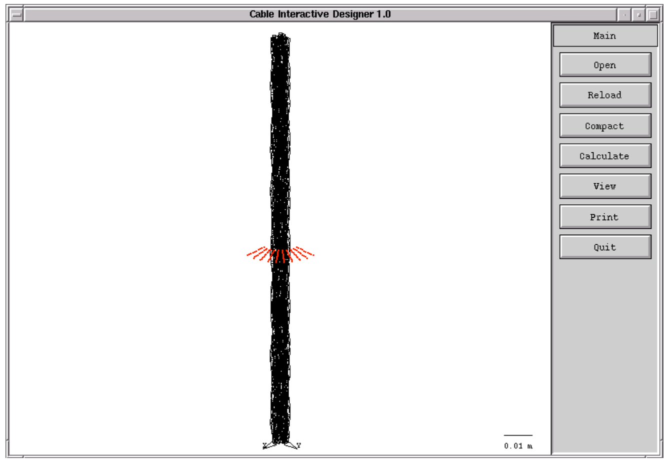
```
   mesh    10 10 0

End
```

CID run

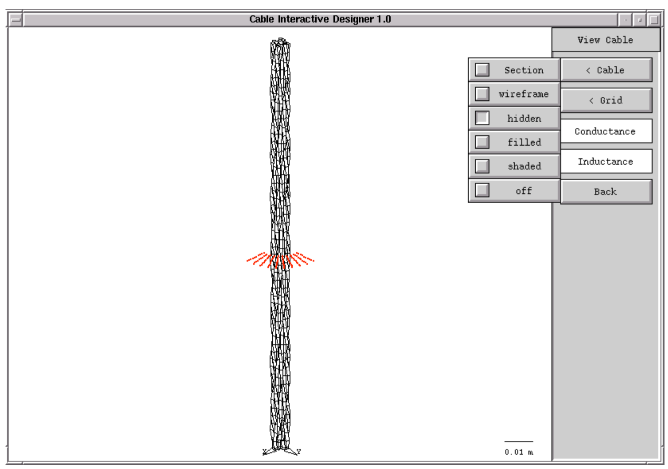
The CID run for the calculation of the interstrand conductance and inductance matrices starts launching the CID GUI, see Chapter 2 for details. CID is then ready to proceed with the session, in the status shown below.



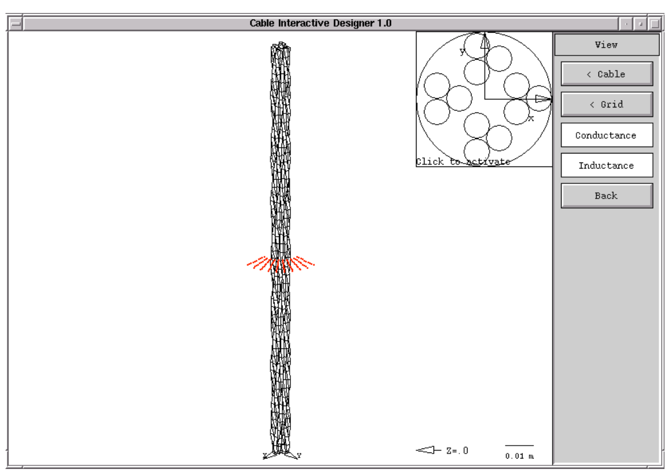
At this point it is necessary to load the model from the input file. Click on the **Open** button and select the input file. CID interprets the geometry, generates the cable and plots it. Also the grid defined in the input file is plotted, and the status is the following:



It is now possible to navigate in the model, changing the view angle using the toolbar, and selecting different display modes for the cable and the grid. The related actions are performed in the **View** menu, reached clicking on the **View** button in the **Main** menu. As a first step we can change the display mode of the cable to hidden lines, rahter than wireframe (the default). This is done in the **View** menu, selecting the **Hidden** mode for the **Cable**:

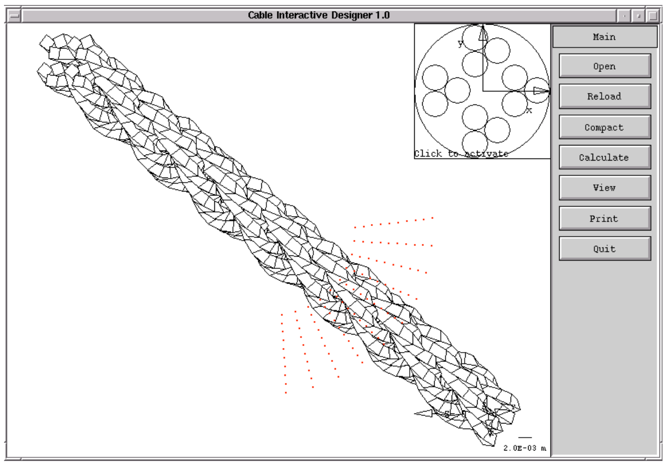


The 2-D cable cross section is displayed activating the 2-D option. To do this click again on the **Cable** button and select the **Section** entry in the pop-up. The result is shown below.

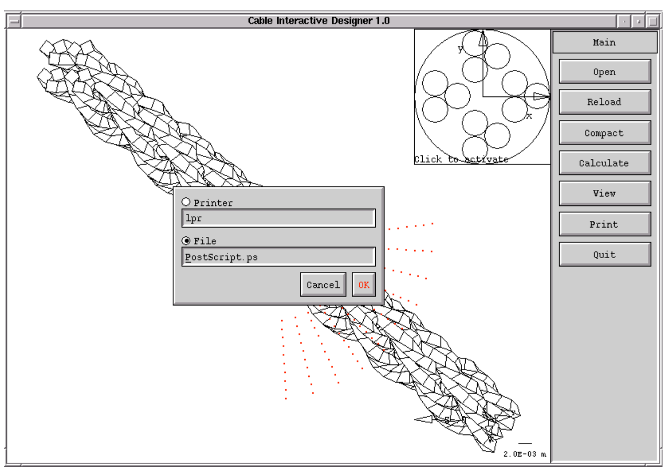


Note the arrow marking the point where the crosss ection is plotted (at Z=0). To move the location for the 2-D cross section plot you click on the 2-D plot area. The mouse is changed to an arrow with a question mark . Clicking along the cable will select a different Z for the 2-D cross section.

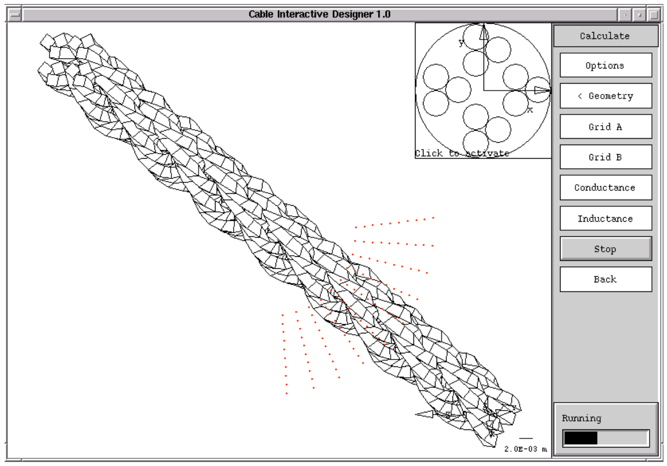
The view point can be changed using the rotation buttons in the toolbar. This allows to navigate in the model. The actions on the toolbar, i.e. changing the view point, area selection and area fitting, are possible from any menu. The result of a change of view point by rotations is shown below:



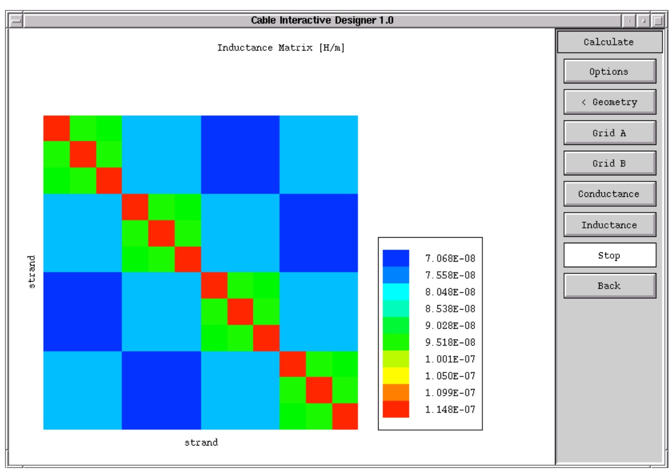
The graphics window can be printed from the main menu clicking on the **Print** button, or alternatively clicking on the print button on the toolbox, or alternatively pressing the “p” or “P” key on the keyboard. You are given a choice of a direct print to a system printer, or to generate a PostScript® file, as shown in the following snap-shot:



Calculations are launched from the **Calculation** menu. Return to the **Main** menu (clicking on the **Back** button), and click on the **Calculation** button. If a calculation is possible the corresponding button is active, in the case that the calculation is not possible, the corresponding button is inactive. This is the case, for instance, for the **Grid A** and **Grid B** calculations in case that no grid has been defined in the input file. To launch a calculation simply click on the corresponding button. Short calculations (e.g. cable geometry in 2-D or 3-D, cable conductance matrix) run almost instantaneously. The cable inductance matrix, on the other hand, may require a long calculation time depending on the number of strands and the number of subdivisions of each strand in GCE's. During an inductance calculation a slider appears at the bottom of the GUI indicating the progress of the calculation.



In the snap-shot above approximately one third of the calculation has been performed. During a calculation the GUI is frozen, apart from the **Stop** button that is activated to abort the calculation performed so far. This feature is present to deal with the case that the execution time is perceived to be too long. Clicking on **Stop** will abort the calculation and return to the normal GUI response, allowing to change parameters (e.g. precision or number of subdivisions in GCE's) and re-launch the calculation without quitting the GUI. At the end of the calculation the structure and module of the computed matrix is displayed in a color scale rendering shown below.



Once the desired calculations are completed, CID can be stopped by returning to the **Main** menu and clicking on the **Quit** button. Note that the session is not saved, and thereofre all settings, other than the results of the calculations, will be lost once the session is closed.

Results on the log file

The results generated by the CID run are logged on the file `cid.log`. In addition the file contains a summary of the model that has been read. Supposing that during the session described in the previous section the user has clicked on the **Inductance** button, the log file will be the following:

cid.log

```
Cable Interactive Designer Version 1.0
file created at 16/11/2002   8:46:26


16/11/2002   8:47:22

Input: manual.input

NrObjects= 3
Cable
  Name              = CS1 stages 1 and 2
  Type              = twisted
  Geometry          = Circular
  Diameter [m]      =   4.2135E-03
  Center x,y,z [m] =   0.0000E+00   0.0000E+00   0.0000E+00
  Length [m]        =   1.0000E-01
  Pitch [m]         =   5.4000E-02
  S/Z               = S
  Mesh x,y,z        = 50
  subobjects        = 4
    subobject nr. 1  = triplet
    subobject nr. 2  = triplet
    subobject nr. 3  = triplet
    subobject nr. 4  = triplet

Object nr. 2
  Name              = triplet
  Type              = twisted
  Diameter [m]      =   1.7453E-03
  Pitch [m]         =   2.5000E-02
  S/Z               = Z
  subobjects        = 3
    subobject nr. 1   = S1
    subobject nr. 2   = S1
    subobject nr. 3   = S1
Object nr. 3
  Name              = S1
  Type              = strand
  Diameter [m]      =   8.1000E-04

Grid
  Name              = plane
  Type              = Cylindrical
  Center x,y,z [m] =   0.0000E+00   0.0000E+00   5.0000E-02
  R in [m]          =   5.0000E-03
  R out [m]         =   1.0000E-02
  Theta1 [deg]      =   0.0000E+00
  Theta2 [deg]      =   9.0000E+01
  dZ [m]            =   1.0000E-01
  Mesh R,Theta,z   = 10 10 1

created  12 strands
nr. of elements per strand: 50
total nr. of elements 600

***********************************************
********** Inductance calculation *************
***********************************************

Relative precision =   1.0000E+00  %
Maximum ratio of dimensions =   1.0000E+03
```

```
 Inductance matrix [H/m]:
  1.1478E-06   9.4594E-07   9.4060E-07   7.8595E-07   7.8524E-07
7.8463E-07   7.1122E-07   7.0979E-07   7.1047E-07   7.8641E-07
7.8277E-07   7.8507E-07
  9.4594E-07   1.1472E-06   9.4540E-07   7.7982E-07   7.8628E-07
7.8260E-07   7.0806E-07   7.1119E-07   7.0979E-07   7.8399E-07
7.8556E-07   7.8493E-07
  9.4060E-07   9.4540E-07   1.1468E-06   7.8096E-07   7.8406E-07
7.8568E-07   7.0681E-07   7.0828E-07   7.1138E-07   7.8121E-07
7.7964E-07   7.8606E-07
  7.8595E-07   7.7982E-07   7.8096E-07   1.1470E-06   9.4586E-07
9.4021E-07   7.8562E-07   7.8265E-07   7.8472E-07   7.1134E-07
7.0722E-07   7.0924E-07
  7.8524E-07   7.8628E-07   7.8406E-07   9.4586E-07   1.1478E-06
9.4585E-07   7.8387E-07   7.8611E-07   7.8524E-07   7.1029E-07
7.1118E-07   7.1022E-07
  7.8463E-07   7.8260E-07   7.8568E-07   9.4021E-07   9.4585E-07
1.1469E-06   7.8090E-07   7.7999E-07   7.8618E-07   7.0907E-07
7.0714E-07   7.1126E-07
  7.1122E-07   7.0806E-07   7.0681E-07   7.8562E-07   7.8387E-07
7.8090E-07   1.1468E-06   9.4542E-07   9.4056E-07   7.8601E-07
7.7950E-07   7.8142E-07
  7.0979E-07   7.1119E-07   7.0828E-07   7.8265E-07   7.8611E-07
7.7999E-07   9.4542E-07   1.1473E-06   9.4595E-07   7.8503E-07
7.8566E-07   7.8398E-07
  7.1047E-07   7.0979E-07   7.1138E-07   7.8472E-07   7.8524E-07
7.8618E-07   9.4056E-07   9.4595E-07   1.1477E-06   7.8481E-07
7.8265E-07   7.8624E-07
  7.8641E-07   7.8399E-07   7.8121E-07   7.1134E-07   7.1029E-07
7.0907E-07   7.8601E-07   7.8503E-07   7.8481E-07   1.1475E-06
9.4553E-07   9.4084E-07
  7.8277E-07   7.8556E-07   7.7964E-07   7.0722E-07   7.1118E-07
7.0714E-07   7.7950E-07   7.8566E-07   7.8265E-07   9.4553E-07
1.1467E-06   9.4556E-07
  7.8507E-07   7.8493E-07   7.8606E-07   7.0924E-07   7.1022E-07
7.1126E-07   7.8142E-07   7.8398E-07   7.8624E-07   9.4084E-07
9.4556E-07   1.1476E-06

 Program exit at 16/11/2002   9:04:06
```

CHAPTER 7

References

[1] CryoSoft Installation Manual, Version 8.0, 2013.