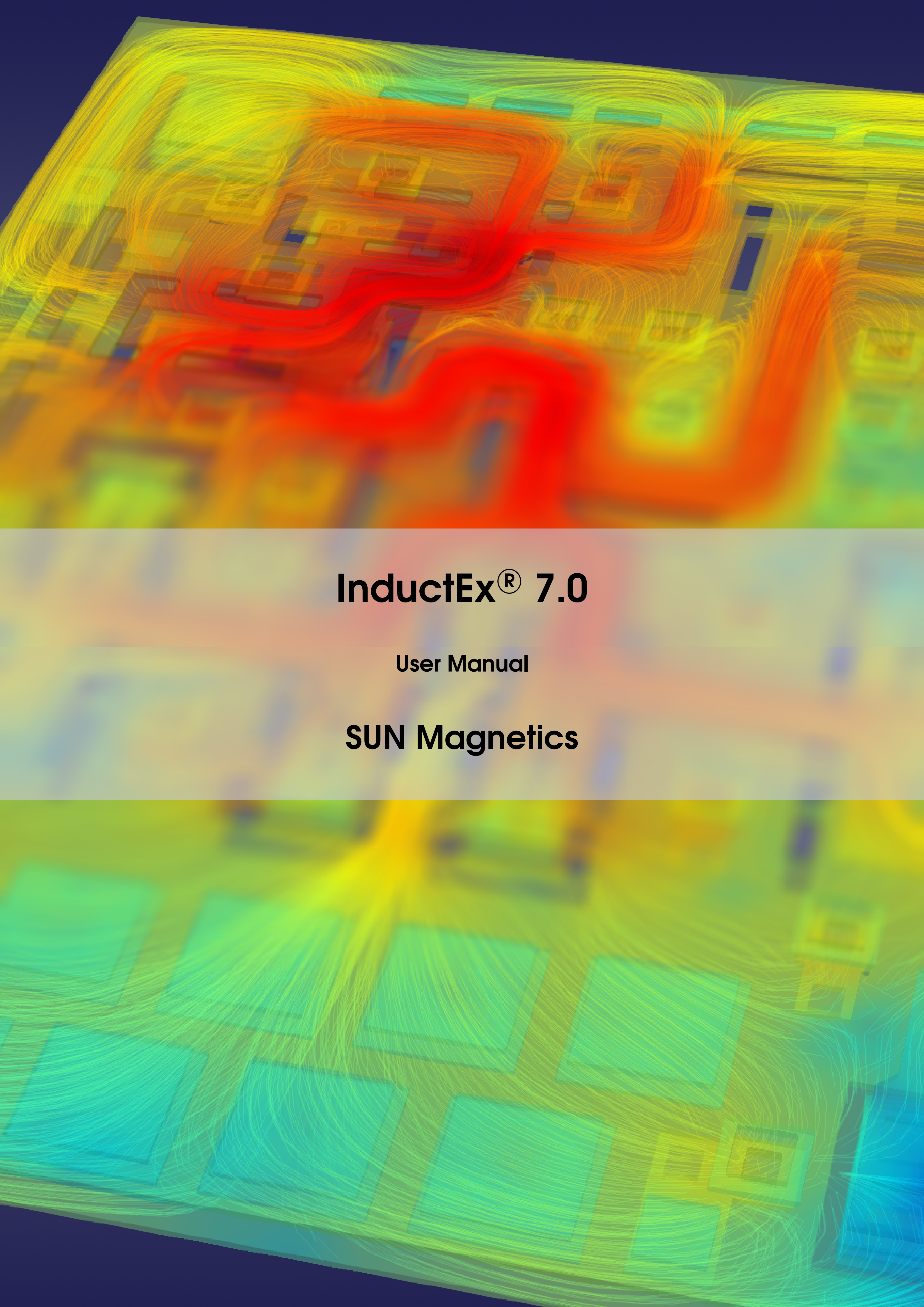


InductEx® 7.0

User Manual

SUN Magnetics

SUN MAGNETICS (RF) (PTY) LTD
15 De Beer Street
Stellenbosch, 7600
Republic of South Africa

WWW.SUN-MAGNETICS.COM

Date: 5 March 2023

Credits

InductEx (the code base, file translators, pre-processor and post-processing algorithms and visualisation tools) is the product of the combined research and development efforts of Coenrad Fourie, Kyle Jackman, Johannes Delport, Mark Volkmann, Paul le Roux, Rebecca Roberts, Thomas Weighill, Ruben van Staden and Pierre Lötter.

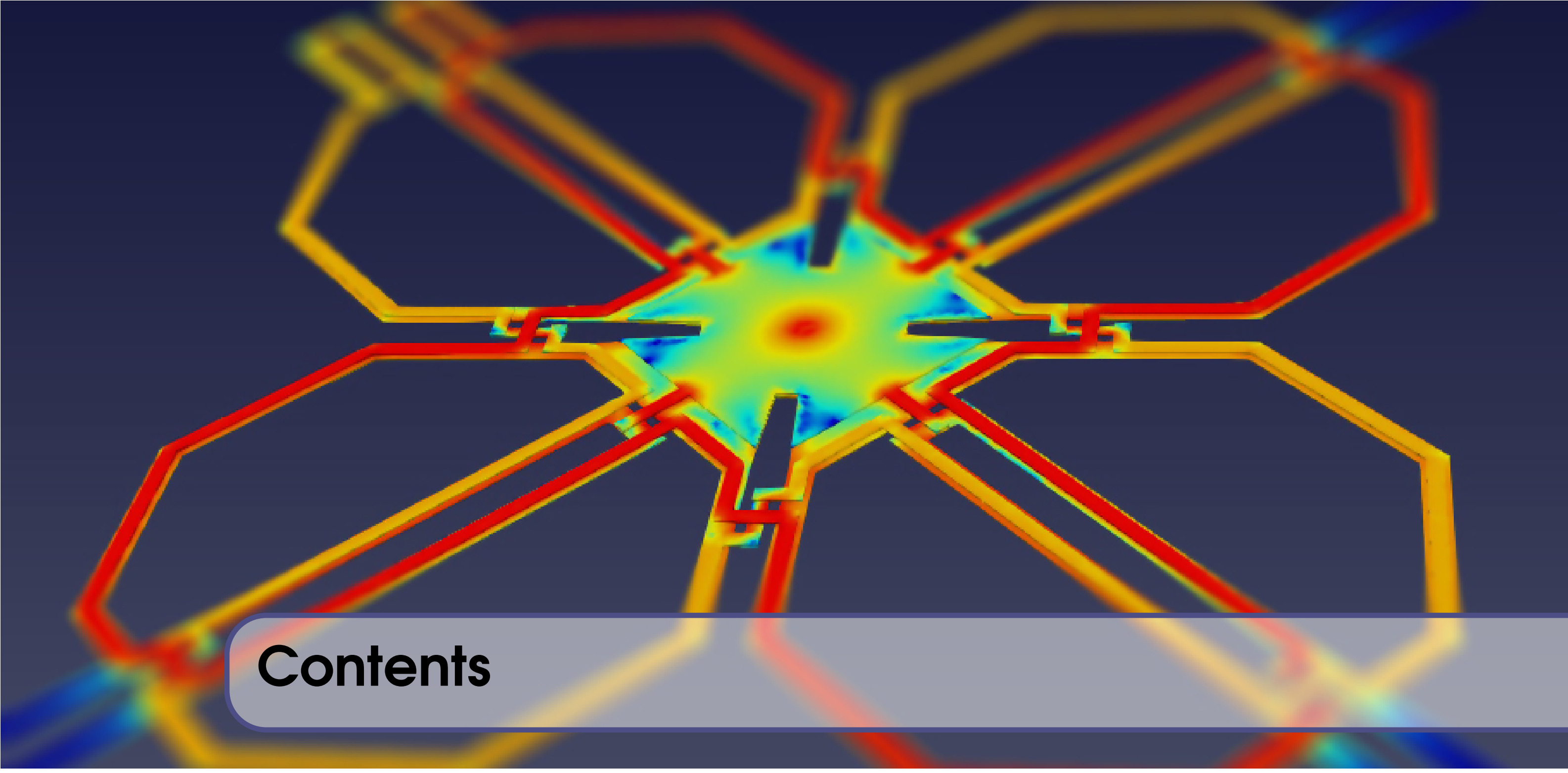
InductEx also builds on the work done by Matton Kamon and Steve Whiteley (FFH), Paul Bunyk and Sergei Rylov (Lmeter), Angus Johnson and Bala Vatti (the polygon clipper unit).

Process-specific support with the assistance of Thomas Ortlepp, Olaf Wetzstein and Jürgen Kunert (FLUXONICS), Oleg Mukhanov and Alex Kirichenko (Seeqc) with Timur Filippov (Hypres), Nobuyuki Yoshikawa, Yuki Yamanashi and Kohei Ehara (AIST), and Sergey Tolpygo (MIT Lincoln Laboratory).

Integration into Cadence design environment with the help of Vasili Semenov (Stony Brook University) and Timur Filippov (Hypres).

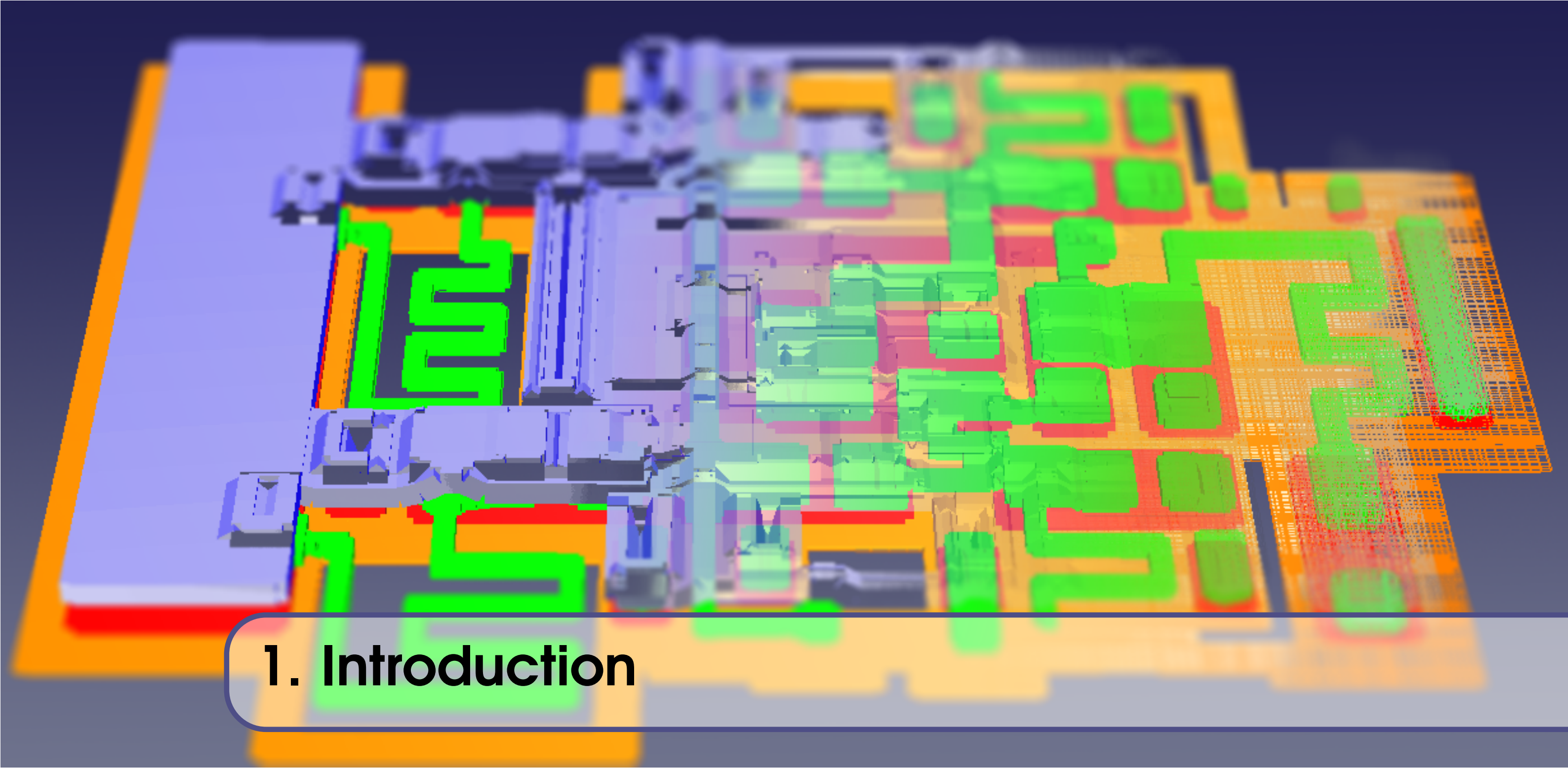
Early version quality assurance (by the patient Beta testers and bug reporters): Vasili Semenov, Timur Filippov and Alex Kirichenko, Pascal Febvre, Romain Collot, Ugur Yilmaz, Felix Jaeckel, Ruslan Hummatov, Xizhu Peng, Naoki Takeuchi, Damian Steiger, Oliver Brandel, Mankit Wong, Jonathan Stark, Steve Kaplan, Manuel Castellanos-Beltran, Kobus Meyer, Lieze Johnston (née Schindler), Mark Volkmann, Rodwell Bakolo and Hein van Heerden.

InductEx is based upon work supported financially by the South African National Research Foundation, grant numbers 69006, 78789 and 93586, with additions developed under contracts W911NF14-C-0089 and W911NF-17-1-0120 with the Intelligence Advanced Research Projects Activity (IARPA)



Contents

InductEx



1. Introduction

1.1 A Note on Modelling

Before you read any further, or do any experiments, take note:

Numerical calculations are only as reliable as the models used to obtain them. No solution is absolutely correct – some are just better than others. You are strongly advised not to put too much trust in a calculation result until you are familiar with modelling and interpreting results and have verified one of your own extractions against a known measurement result for a similar structure.

If you have limited experience in modelling layouts for parameter extraction, please study the examples described in this manual carefully before you attempt to do your own extractions. A simple modelling mistake (such as inverting the polarity of a port on a layout or making a connection mistake in a circuit netlist) can lead to wildly inaccurate solutions.

***InductEx* warns about possible modelling errors. It is helpful to know how to interpret those warnings to identify mistakes.**

1.2 History

InductEx [1–5] was developed to enable VLSI circuit designers to extract the inductance of complex 3D structures in superconducting integrated circuits, but also supports normal conductivity and the calculation of inductance in simpler structures such as SQUID loops or wires. Current distribution in, and magnetic fields in and around conductors can also be computed. Although early versions were focused on inductive substructures, *InductEx* is now aimed at full-gate extraction [5] and analyses that determine external magnetic field and trapped flux [6–8] coupling to circuit inductance.

InductEx also supports capacitance, impedance and S-parameter calculations for superconductor and normal conductor integrated circuit and printed circuit board layouts.

InductEx functions well as a stand-alone command line programme that accepts GDSII input files created by any layout software, but was carefully designed to be integrated with CAD software through its own, more powerful IXI interface.

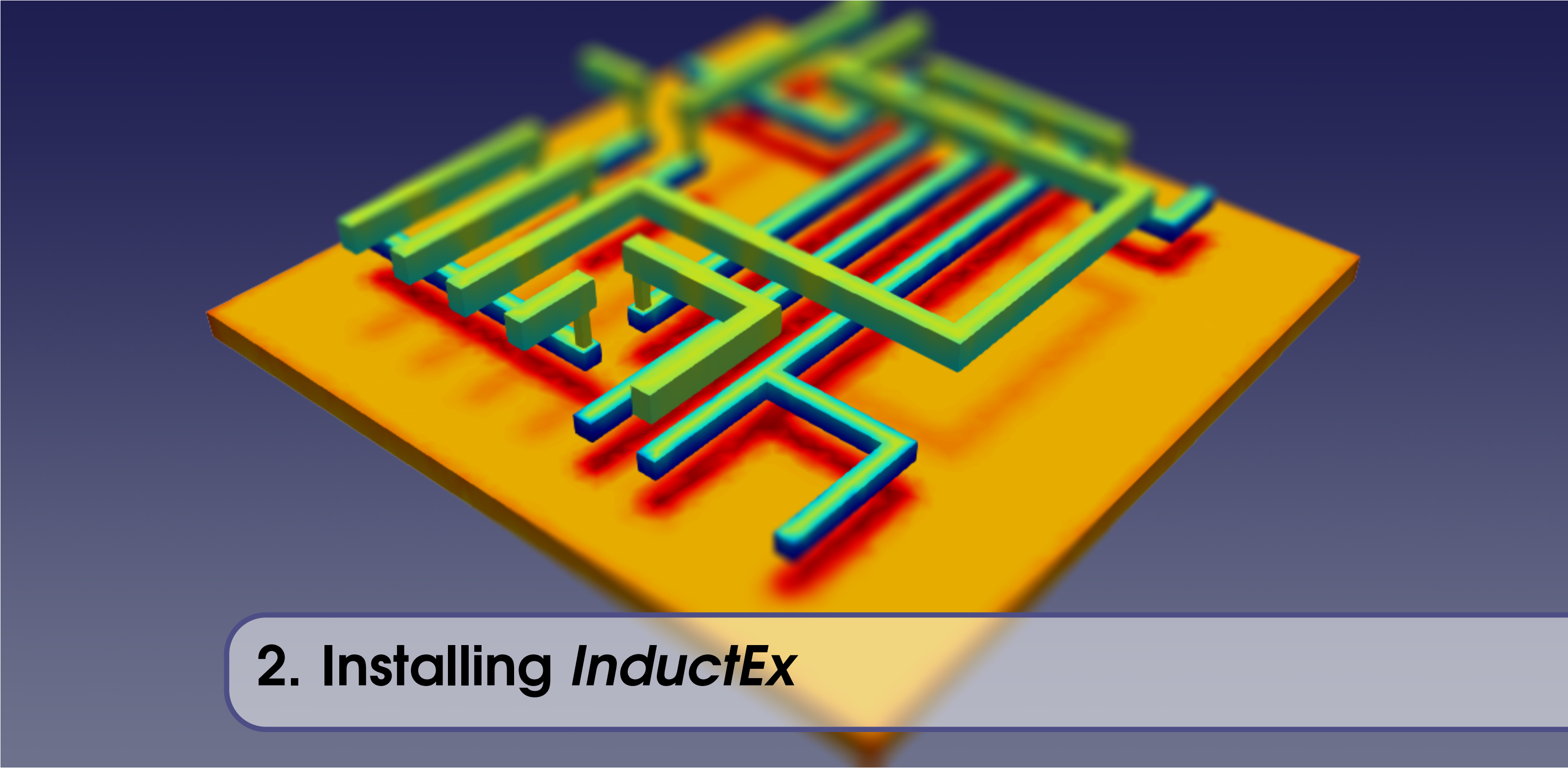
InductEx is a powerful tool for geometry processing, modelling and segmentation of integrated circuit layout structures and the solution of inductive or capacitive networks. For inductance calculation, *InductEx* uses numerical engines to calculate current distribution in layout structures, from which inductance and coupling is determined. *InductEx* uses its own numerical solver, *TetraHenry* [9], a purpose-built magnetoquasistatic, electroquasistic, EMQS and full-wave solver.

Your use of *InductEx* is subject to the licence terms made available on the SUN Magnetics website (`www.sun-magnetics.com`) and included with every installation file.

Part of the design philosophy behind *InductEx* is easy integration into an EDA tool chain. *InductEx* thus slots well into the Synopsys design environment, for which it is actively maintained.

InductEx also interfaces well with LayoutEditor and Klayout, can be used with XIC, and with the correct script files it works directly from Cadence Virtuoso.

Finally, *InductEx* was developed to allow inductance extraction of complete circuit cells. The intention is to allow users to work directly on "ready-for-fabrication" cells without the need to alter layouts during modelling. To this end, *InductEx* supports auxiliary layers and operators, and cells that pass extraction may be sent for fabrication without the need to strip out modelling text or objects.



2. Installing *InductEx*

2.1 Microsoft Windows

Download the install file for Windows from the SUN Magnetics website (`www.sun-magnetics.com`), open the file and follow the instructions. You will be asked to select an install directory ([AppDir]). The installer automatically adds the directory for the *InductEx* executables (or binaries) ([Appdir]\bin) to the path. The default licence directory, ([AppDir]\licenses\), is added to the system environment variable `IXLICDIR`, and the licence directory is created during install. However, you need to place the licence file in this directory yourself.

In some cases the installer does not add the directory for the *InductEx* binaries to the path. You then have to add it to the path manually. If, for example, your install directory is `c:\utils\InductEx`, you can (permanently) set the path in MS Windows from the command prompt with:

```
Setx path '%path%;c:\utils\inductex\bin'
```

The install directory can also be added to the path by selecting *Advanced System Settings* from the *Control Panel*, opening the *Advanced* tab and clicking on *Environment Variables*. You can alter the system environment variable for the licence file in the same way (see Fig. 2.1). The installer sets `IXLICDIR` as a system environment variable. Under Windows 11 you may need to declare this as a user environment variable.

After installation, the uninstall shortcut and user manual can be found under "*InductEx*" in the Windows Start Menu.

InductEx is executed directly from the command prompt. You have to run the command prompt application as administrator to enable the licence directory environment variable and the correct path.

2.2 Linux

InductEx and its auxiliary tools are packaged in a tar.xz zipped tarball distribution file. Unpack it to the directory of your choice.

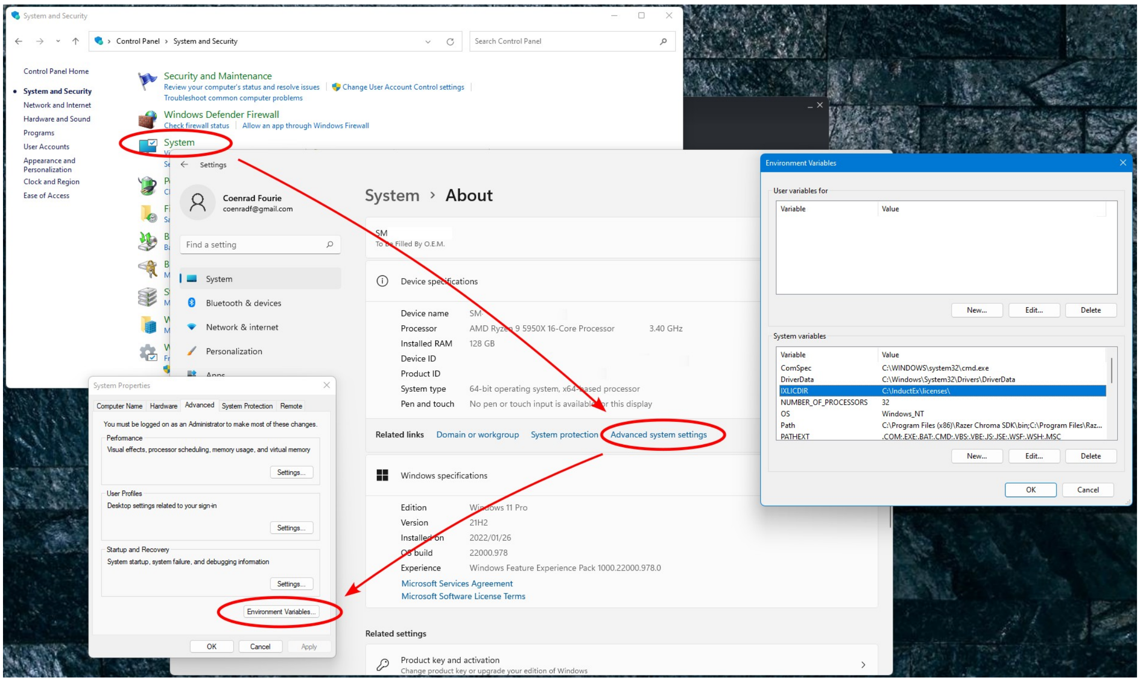


Figure 2.1: Setting system environment variables in Windows 11. Add the user environment variable manually if needed.

The *InductEx* Linux 64-bit distribution contains tools that are built on the latest version of Ubuntu and works for Linux kernel 3.

2.3 macOS

For macOS, *InductEx* and its auxiliary tools are packaged in a zip file. The preferred location for the binary files is in `/opt/local/bin`.

If *InductEx* is not accessible from the terminal, it may be that the path does not include the location of the binaries.

Verify the path environment variable with:

```
Echo $PATH
```

The binaries directory can be added to the path with:

```
Export PATH=$PATH:~/opt/local/bin
```

If you use MacPorts and execution of the *TTH* engine fails due to a symbol binding or symbol not found error, installation of cctools and gcc10 should resolve the issue:

```
sudo port install cctools
sudo port install gcc10
```

InductEx and its components are not yet signed with an Apple developer ID, and you will be warned about their execution. If the binaries are not executable, each can be made so:

```
sudo chmod 777 /opt/local/bin/inductex
```

```
sudo chmod 777 /opt/local/bin/tth
```

etc.

Now open the directory `/opt/local/bin` in Finder and select to open each of the binaries (`inductex` and `tth`) with Terminal. After you accept, *InductEx* will execute normally in Terminal from any other directory.

2.4 Licence File

InductEx requires the presence of a valid licence file to execute. This file is always called `ix_license.txt`, and is issued by the *InductEx* team upon submission of a system identification file, `IX_sysID.txt`.

If *InductEx* cannot find a licence file, it calculates a system identification number and writes this to file `IX_sysID.txt` in the same directory. The *InductEx* team then uses this ID to generate a licence key unique to your computer system. The licence file, `ix_license.txt`, must be placed where *InductEx* can find it. Under MS Windows, it can be in any directory, provided that the directory is identified with the environment variable `IXLICDIR`. The default install setup sets `IXLICDIR = c:\InductEx\licenses`.

Under Linux and Mac OS X, the licence file must be placed in the directory `/usr/local/share/inductex/licenses/ix_license.txt`. You will need root privileges to create the licence directory and copy the license file here. Also note that you will at least need read permission for world users, which can be checked with `ls -l` and set with `chmod`.

It is possible to override the `IXLICDIR` environment variable under MS Windows, or the default directory under Linux and Mac OS X by directly specifying the location of the licence file in the parameter string when *InductEx* is executed. This is done with the “-L” switch. Some examples are:

```
inductex someexample.ixi -L c:\mydocs\licenses\ix_license.txt
inductex anotherexample.ixi -L "c:\name with spaces\ix_license.txt"
```

2.5 Required third party software - Gmsh

For tetrahedral meshes used with the *TetraHenry* engine, the 3rd party open source meshing tool Gmsh [10] must be used, unless you have access to another mesher that outputs files in the .msh format used by Gmsh.

Gmsh is available from `www.gmsh.info` for Windows, Linux and macos, and the binary is included with the *InductEx* installation files for Windows and Linux.

It is possible and advisable to place the Gmsh executable with the *InductEx* executables for Windows and Linux.

2.5.1 Centos

Under Centos, in the terminal, make sure that pip is installed, or add it with

```
sudo yum install python3 python3-pip
```

Now install Gmsh with:

```
pip3 install gmsh --user
```

If the Gmsh binary shipped with the InductEx install files is used, Centos 7 and older could fail to execute Gmsh due to a missing Gmsh dependency, libGLU.so.1. To resolve this, install libGLU with:

```
sudo yum install libGLU
```

2.5.2 macOS

In macos, download the latest installation as a disk image (.dmg) and mount the image (open the .dmg file with DiskImageMounter). The Gmsh binary can be located in the Gmsh.app package as shown in Fig. 2.2.

Now drag the Gmsh app to the Applications folder (in the top left corner of Fig. 2.2).

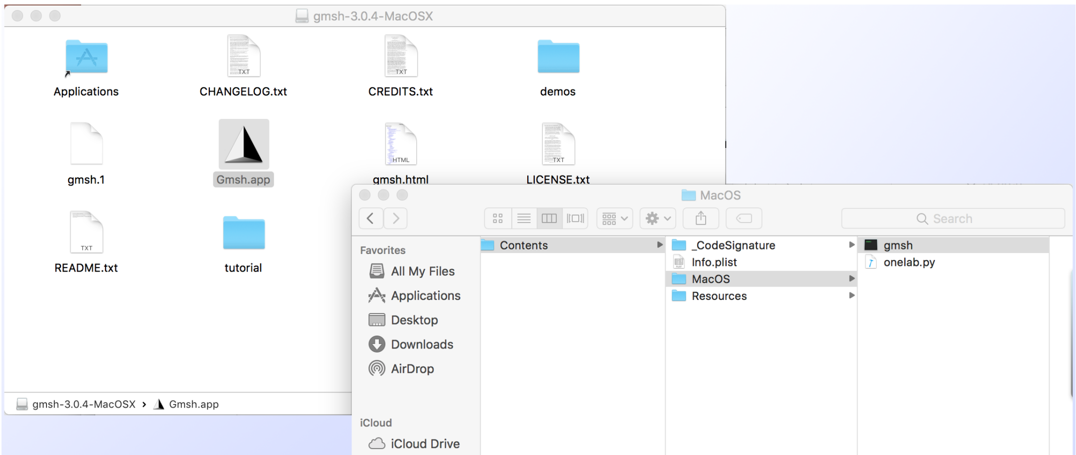
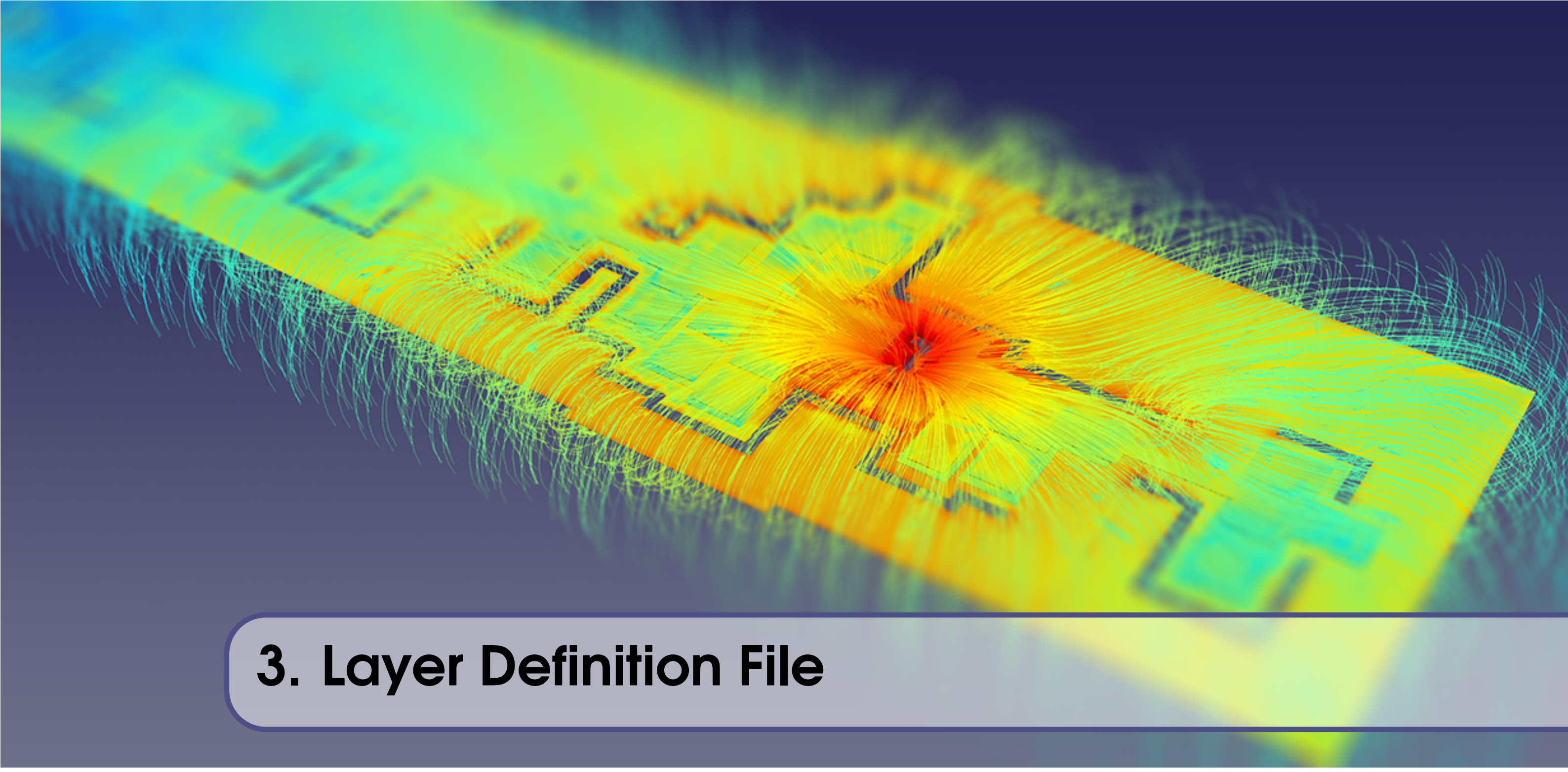


Figure 2.2: Locating the Gmsh binary in the Gmsh.app package under macOS.

After this, Gmsh will be visible to *InductEx* . You can test if this step is successful by opening Terminal and running:

```
/Applications/Gmsh.app/Contents/MacOS/gmsh
```

Gmsh should open in GUI mode.



3. Layer Definition File

3.1 Use of the Layer Definition File

InductEx is primarily aimed at layer-based devices such as integrated circuit dies or printed circuit board structures (or a combination of these), which are usually defined as a set of layer mask layouts.

The layer definition file describes to *InductEx* how a layout drawing (which is essentially a set of disconnected two-dimensional objects) should be modelled in three dimensions with the correct layer sequence and connections. As such, it describes the fabrication process, with the important caveat that the **description is focused on creating numerical models that have the same geometry as fabricated circuit structures** but may differ (substantially) from the actual fabrication sequence.

The **layer definition file** is thus also referred to as the **process file** or the **technology file**.

Without the layer definition file, *InductEx* has no information with which to build a three-dimensional calculation model from a layer-based two-dimensional layout.

The layer definition file controls or configures *InductEx* by defining both the modelling and process technology parameters used to build extraction models.

Modelling parameters are not derived from the actual fabrication process. These parameters merely control segment size, ground plane modelling, optional segment blanking, etc.

Process technology parameters describe the actual layers in terms of model construction order (which is not necessarily the same as the fabrication order), dimensions, mask-to-wafer bias and physical parameters such as penetration depth.

During model construction, every layout input file must be assigned a corresponding layer definition file.

Standard layer definition files for superconductor integrated circuit processes of which the design rules are in the public domain are available on the SUN Magnetics website.

Layer definition files for some other processes are available to registered users.

3.2 File structure

3.2.1 Control blocks

The layer definition file (LDF) contains a parameter block, layer definitions and optional operator definitions.

The global parameters, as defined in section 3.2.3, are defined inside a control block starting with

```
$Parameters
```

and ending with

```
$End
```

Parameter entry is described in section 3.2.2.

Every layer has to be defined as well. Layers are defined in control blocks starting with

```
$Layer
```

and ending with

```
$End
```

The layer parameters with default values and functionality are listed in section 3.2.4.

Apart from parameter and layer definitions, *InductEx* allows the user to define specific operators that may be added to layouts to control modelling. These operators, if used effectively, can reduce segment count and solution time, but need to be used with care. Operator parameters are listed in section 3.2.5.

Every operator has to be defined in the layer definition file. Operator control blocks start with

```
$Operator
```

and end with

```
$End
```

3.2.2 LDF parameter entry

Global or layer parameters, or parameters for operators and constructors, are entered in the format

```
parametername = value
```

The `value` can be an integer or floating point number, a string (with no spaces), a boolean value (`TRUE` or `FALSE`), or an array of entries. When `value` is an array, the array entries are separated by spaces and the array is enclosed in square brackets, for example:

```
LayerADD = [10 15 22]
```

3.2.3 LDF global parameters

AbsMin (*Floating point; Default = 0.01*)
Sets the radius of a test sphere around a current density node within which it is mapped to a geometry node when filaments (cuboid meshes) are used. The value must **not** be zero).

BlankAllCutsGP (*Boolean; Default* = FALSE)
If TRUE, BlankAllLayer also blanks the ground plane.

BlankAllLayer (*Integer*)
Sets the optional global blanking layer number. No objects or parts of objects falling within blanking layer objects are segmented, which allows user-defined model trimming.

BlankXLayer (*Integer*)
Sets the optional x-direction blanking layer number. Any objects or parts of objects falling within x-blanking layer objects are only segmented in the y direction. Applicable to cuboid filament meshes only.

BlankYLayer (*Integer*)
Sets the optional y-direction blanking layer number. Any objects or parts of objects falling within y-blanking layer objects are only segmented in the x direction. Applicable to cuboid filament meshes only.

ChipEdgeLayer (*Integer*)
Sets the optional chip edge layer number. A chip edge is determined from the outer edges of the union of all objects on this layer, and is used to delimit dielectric boundary layers and build chip edges for chip-scale extraction.

CropGP (*Boolean; Default* = TRUE)
Boolean value to indicate that negative mask ground planes should be cropped to within GPOverHang of the union of all non-ground plane structures (this reduces unnecessary segments). It does not affect positive mask ground planes.

DataTypeNotZero (*Boolean; Default* = FALSE)
If TRUE, GDSII objects with DataType <> 0 are also accepted for model building.

DecimationDistance (*Floating point; Default = Units/100*)
Sets the Hausdorff distance (or epsilon), in Units, used by the Ramer-Douglas-Peucker polygon boundary decimation algorithm if PolyDecimation is TRUE.

DielectricMeshSize (*Floating point; Default = SegmentSize*)
Sets the maximum triangle sidelength for mesh elements in a dielectric-dielectric interface layer (for capacitance calculation).

DielectricOverhang (*Floating point; Default = GPOverHang*)
Sets the distance in Units between the edge of any object and the edge of a rectangular plane that describes the outline of a meshed dielectric interface plane (for capacitance calculation).

EpsilonR (*Floating point; Default = 1*)
Optional parameter that defines the relative dielectric constant, or relative permittivity, of isolation layers.

EpsilonRAbove (*Floating point; Default = 1*)
Optional parameter that defines the relative dielectric constant, or relative permittivity, of the global environment above the last manufactured layer on the wafer and outside of the chip edge. The default is 1 for free space.

EpsilonRBelow (*Floating point; Default = 11.7*)
Optional parameter that defines the relative dielectric constant, or relative permittivity, of the global environment below the lowest layer on the wafer (essentially the substrate). The default is 11.7 for silicon at 300 K.

ExtractBoundLayer (*Integer*)
Sets the optional extraction boundary layer (according to its GDS layer number). If a polygon

exists on this layer, all other layout objects will be clipped to its boundaries to limit the size of an extraction model.

FieldCageScale (*Floating point; Default = 2.0*)
Sets the scaling of the cuboid cage around a meshed model in which electric or magnetic field is calculated for visualisation purposes. The longest dimension of the meshed model in any Cartesian axis is designated unity, and the minimum scaling value that *InductEx* accepts is 1.1.

FieldCageZScale (*Floating point; Default = 1.0*)
Sets the z-axis scaling of the cuboid cage around a meshed model in which electric or magnetic field is calculated for visualisation purposes. FieldCageScale is designated unity. The minimum scaling value that *InductEx* accepts is 0.01.

FieldMeshSize (*Floating point; Default = 2.0*)
Sets the size of mesh elements on the outside of the cubical field calculation cage in Units.

Frequency (*Floating point; Default = 10E9*)
The frequency of the port excitation sources. For purely superconductive circuits at frequencies below the gap frequency, this does not influence the results. With resistive components, the frequency dependent skin depth affects current distribution in the normal conductors.

GMSHMeshAlgorithm (*Integer; Default = 1*)
Selects the 2D mesh algorithm for Gmsh when triangular meshes are used. Options are:

1 MeshAdapt.
2 Automatic.
5 Delaunay.
6 Frontal.
8 DelQuadA.

GMSHMeshAlgorithm3D (*Default = 1*)
Selects the 3D mesh algorithm for Gmsh when tetrahedral meshes are used. Options are:

1 Delaunay.
2 New Delaunay.
4 Frontal.
5 Frontal Delaunay.
6 Frontal Hex.
7 MMG3D.
9 R-Tree.

GPLayer (*Integer*)
Indicates the GDS layer number of the ground plane, which is used to fill negative-mask ground planes to the defined boundary.

GPOverHang (*Floating point; Default = 2.5*)
Sets the distance in Units between the edge of any object and the edge of the ground plane that is generated to cover the smallest necessary area beneath structures. A smaller value results in artificially higher calculation results. 2.5 is a good value for all processes, but 7.5 is recommended when lines crossing holes in the ground plane are investigated (the ground plane automatically wraps around holes). This parameter is disregarded for positive mask ground planes.

GridSnapOn (*Boolean; Default* = FALSE)
If TRUE, snaps all layout object vertices (after polygon union and bias adjustments) to a grid defined by SnapSize.

HFilaments (*Integer; Default = 1*)
The global value for the number of height filaments into which segments are divided. Can be overrided per layer. Applicable to cuboid filament meshes only.

Lambda (*Floating point; Default = 0.09*)
Global value for London penetration depth. Can be overrided per layer. If Lambda is defined for a layer, Sigma and MuR are ignored.

Lambda0 (*Floating point; Default = 0.09*)
Global value for the London penetration depth at 0 K if temperature is specified. Can be overrided per layer.

LambdaFit (*Floating point; Default = 4*)
Exponent for temperature dependent London penetration depth.

LastDieLayerOrder (Deprecated) (*Integer*)
Indicates the order of the last layer fabricated on the wafer. This is used to separate on-chip and off-chip layers when packaging, bonding and excitation coil structures are included in a model in versions before 7.0. From 7.0 up, off-chip layers are handled via multiple chip/process definitions.

PolyDecimation (*Boolean; Default* = TRUE)
If TRUE, decimates the boundary of a polygon with the Ramer-Douglas-Peucker algorithm to find a similar polygon with fewer points. The dissimilar distance is defined with DecimationDistance.

ProcessHasGroundPlane (*Boolean; Default* = TRUE)
Boolean value to indicate ground plane in process. Usually only FALSE for monolayer HTS circuits.

Relief (*Boolean; Default* = TRUE)
If FALSE, *InductEx* does not adjust mesh elements for tetrahedral and triangular meshes to reflect relief of structures overlapping other structures in unplanarised layers below.

ResZero (*Floating point; Default = 1E-3*)
Sets the smallest calculated resistance value printed to the output – below this the resistance is considered zero and an element is considered to be purely inductive.

SegmentSize (*Floating point; Default = 1.0*)
Sets the maximum side length any segment in Units. Larger values speed up calculations but cause higher inductance results. Always compare calculations against a known result after you adjust SegmentSize. (Replaces old parameter “GapMax”).

SheetCurrents (*Boolean; Default* = TRUE)
Boolean value to set the use of sheet current models in conducting planar structures when *TetraHenry* is used. Set FALSE to model plane structures as volumes rather than sheets.

Sigma (*Floating point; Default = 10*)
Bulk conductivity of resistive layers (global). Calculated as the inverse of resistance per square times layer thickness for thin films, and expressed as Ω^{-1}Units^{-1}. For example, if unit size is in µm, and layer thickness is 80 nm, a conductivity of $10/(\Omega\,\mu\mathrm{m})$ equals a sheet resistance of $1.25\,\Omega/\square$.

SnapSize (*Floating point; Default = AbsMin*)
Sets the grid size to which object vertices are snapped if GridSnapOn is TRUE.

SubstrateThickness (*Floating point; Default = 50*)
Sets the thickness of the substrate layer in Units, used for dielectric boundary modelling.

TerminalInRange (*Floating point; Default = 1*)
Sets the distance (in Units) from the rectangular boundary around any terminal object within which a text label will be accepted as “linked” to that object.

TermLayer (*Integer*)
Indicates the GDS layer number of the layer on which terminals are drawn.

TC (*Floating point; Default = 9.2*)
Global critical temperature in kelvins for transition to superconductivity. Can be overrided

per layer.

Temperature (*Floating point*)
Global temperature. The default is not set. If temperature is defined, all London penetration depth values are calculated from $\lambda(T) = \lambda(0)[1-(T/T_c)^P]^{(-1/2)}$, where P is the defined by LambdaFit.

TextLayer (*Integer*)
Sets the text layer (according to its GDS layer number).

UnitI (*Floating point; Default = 1.0*)
Sets a unit size in Ampere to which all current results are normalized. Current results are divided by the value of UnitI after calculation.

UnitL (*Floating point; Default = 1.0*)
Sets a unit size in Henry to which all inductance (self and mutual) results are normalized. Inductance results are divided by the value of UnitL after calculation. As an example, inductance results can be normalized to PSCAN dimensionless units by setting UnitL = 2.64E-12. If UnitL = 1E-6, then results are displayed in µH, while UnitL = 1E-12 leads to results in pH.

UnitR (*Floating point; Default = 1.0*)
Sets a unit size in Ohm to which all resistance results are normalized. Resistance results are divided by the value of UnitR after calculation.

Units (*Floating point; Default = 1E-6*)
The unit size in meters in the layer definition file. The default is 1 µm.

VirtualGap (*Floating point; Default = 0.2*SegmentSize*)
Size of the gap cut by a virtual cut terminal when cuboid filaments are used. The default is SegmentSize since version 5.06. It was fixed at AbsMin in earlier versions.

3.2.4 LDF layer parameters

Bias (*Floating point; Default = 0*)
Defines the mask-wafer offset for the layer. The value is added to the boundaries of any object on the layer (positive values "grow" the object, while negative values "shrink" the object). For processes with zero mask-wafer offset, Bias is still available as a calibration parameter to fit extractions to measurements. Bias is not supported when tetrahedral modelling and the *TetraHenry* engine is used.

DecimationDistance (*Floating point; Default = Units/100*)
Sets the Hausdorff distance (or epsilon), in Units, used by the Ramer-Douglas-Peucker polygon boundary decimation algorithm for this layer if PolyDecimation is TRUE.

EpsilonR (*Floating point; Default = 4.6*)
Defines the relative dielectric constant, or relative permittivity.

Filmtype (*Character*)
Sets the film type of the layer. The options are:

A Auxiliary layer.
C Cut layer (everything underneath ablated away).
I Isolation layer.
N Normal (resistive) layer that is not segmented (used for instance for superconductive circuits where only inductance is extracted, and resistive components should be ignored during calculation).
R Resistive layer that is segmented.
S Superconducting layer.

HFilaments (*Integer; Default = Global HFilaments*)
Sets the number of height filaments into which segments on this layer will be divided when

cuboid filament meshing is used. This is only applied to superconductive layers. More filaments result in exponentially longer calculation times, but it is suggested that this value is set to 2 if a layer is thicker than 1.5 times the penetration depth, and to 3 if a layer is thicker than 2.5 times the penetration depth. For known superconductor thin-film processes it is of no benefit to exceed 3.

IDensity (*Floating point*)
: Optional parameter that sets the current density of a superconductive Josephson junction layer. This is multiplied by the area of objects in the layer if it exists between the terminals of a port to give junction critical current. This parameter can be used in other processes to yield any linear relationship to the area of a layout object of interest.

IsGP (*Boolean; Default* = FALSE)
: Used for auxiliary ground planes in processes with multiple ground plane layers. When set to TRUE, this layer is cropped similarly to the main ground plane if the global parameter CropGP is TRUE, while metal layers below and above cast shadows on this layer.

Lambda (*Floating point; Default = Global Lambda*)
: Optional parameter that defines the London penetration depth of the layer. If undefined, the global value is used.

Lambda0 (*Floating point; Default = Global Lambda0*)
: Optional parameter that defines the London penetration depth of the layer at 0 K. If undefined, the global value is used.

LambdaFit (*Floating point; Default = Global LambdaFit*)
: Optional parameter that defines the London penetration depth of the layer at 0 K. If undefined, the global value is used.

LayerAdd (*Array of Integer*)
: GDS/layer number of any other layer to be added to this layer[1]. Useful for layer operations.

LayerSub (*Array of Integer*)
: GDS/layer number of any other layer to be subtracted from this layer. Useful for layer operations.

Mask (*Default = 0*)
: Sets the mask polarity of the layer. The options are:
  - **1** Positive layer (deposited where objects are defined). Normally all superconductive and resistive layers.
  - **0** Layer does not contribute to the vertical height of the wafer – typically used to define auxiliary or operational layers.
  - **-1** Negative layer (etched away where objects are defined). Normally all isolation and anodization layers, and only effective on metal layers designated as ground planes.
  - **-2** Reserved for the blanking layer.
  - **-3** Reserved for the blanking layer.
  - **-4** Reserved for the terminal layer.

MuR (*Default = 1*)
: Optional parameter that defines the relative permeability of a normal metal layer (ignored for superconducting layers). The default is 1 for free space.

Name (*String*)
: This parameter is required and sets the name of the layer for terminal label definitions. The name is also applied to output files for visualization.

Number (*Integer*)
: This parameter is required, and sets the layer number associated with this layer (which coincides with the GDSII layer number if GDSII input is used). Since version 4.26, this

[1] Layer addition and subtraction precedence is: 1 – LayerAdd; 2 – LayerSub.

number can range from 0 - 255.

Order (*Integer*)
Required parameter with integer value 0 or higher that sets the construction order of the wafer. The lowest order layer is processed first and the rest in sequence of increasing order. The order numbers do not have to be sequential. Different layers cannot share the same order.

PlanarModel (*Integer; Default = 0*)
Sets the planarization model applied to this layer. The options are:

0 No planarization.

1 Full planarization up to last layer before this one. If set to 1 for an isolation layer directly below a metal layer, it creates depressions at vias through this isolation layer. This implements the complemented caldera planarization method used by AIST for the ADP process [11]. If set to 1 for a metal layer, vias to lower layers will be of the stud type.

2 (and higher) Reserved for future models.

SegmentSize (*Floating point; Default = Global SegmentSize*)
Optional parameter that sets the maximum x and y dimensions of any segment in Units (see the description under global parameters in section 3.2.3 for more detail). If undefined, the global value is used.

Shadows (*Boolean; Default* = FALSE)
If TRUE, enables shadow casting of the outlines of other metal layers to this layer. If undefined, the value of IsGP is used.

SheetCurrents (*Boolean; Default = Global SheetCurrent*)
Optional boolean parameter to set the use of sheet current models in the layer when *Tetra-Henry* is used.

Sigma (*Floating point; Default = Global Sigma*)
Optional parameter that defines the bulk conductivity of a layer (see the description under global parameters in section 3.2.3 for more detail). If undefined, the global value is used.

TC (*Floating point; Default = Global TC*)
Optional parameter to define critical temperature in kelvins for transition to superconductivity. See the description under global parameters in section 3.2.3 for more detail. If undefined, the global value is used.

Temperature (*Floating point; Default = Global T*)
Optional layer temperature. See the description under global parameters in section 3.2.3 for more detail. If undefined, the global value is used.

Thickness
Required parameter that defines layer thickness in Units.

ViaBypass (*Boolean; Default* = FALSE)
When set to true for a conductive layer of mask type -1 or +1, via etching through an isolation layer directly above this layer will continue to the isolation layer directly below this layer if no conductive object is in the way.

3.2.5 LDF operators

LayersConnect (*Array of Integer*)
Creates a list of GDS layer numbers (separated by space or TAB characters) of layers on which nodes should be electrically connected by the operator.

LayersRemove (*Array of Integer*)
Creates a list of GDS layer numbers (separated by space or TAB characters) of layers on which objects should be removed by the operator.

LayersTransform (*Array of Integer*)
Creates a list of GDS layer numbers (separated by space or TAB characters) of layers on

which objects should be transformed by the operator.

Name

This parameter is required, and sets the name of the operator as it is used in text labels on layouts.

Type (*Default = UD*)

Sets the operator type. The options are:

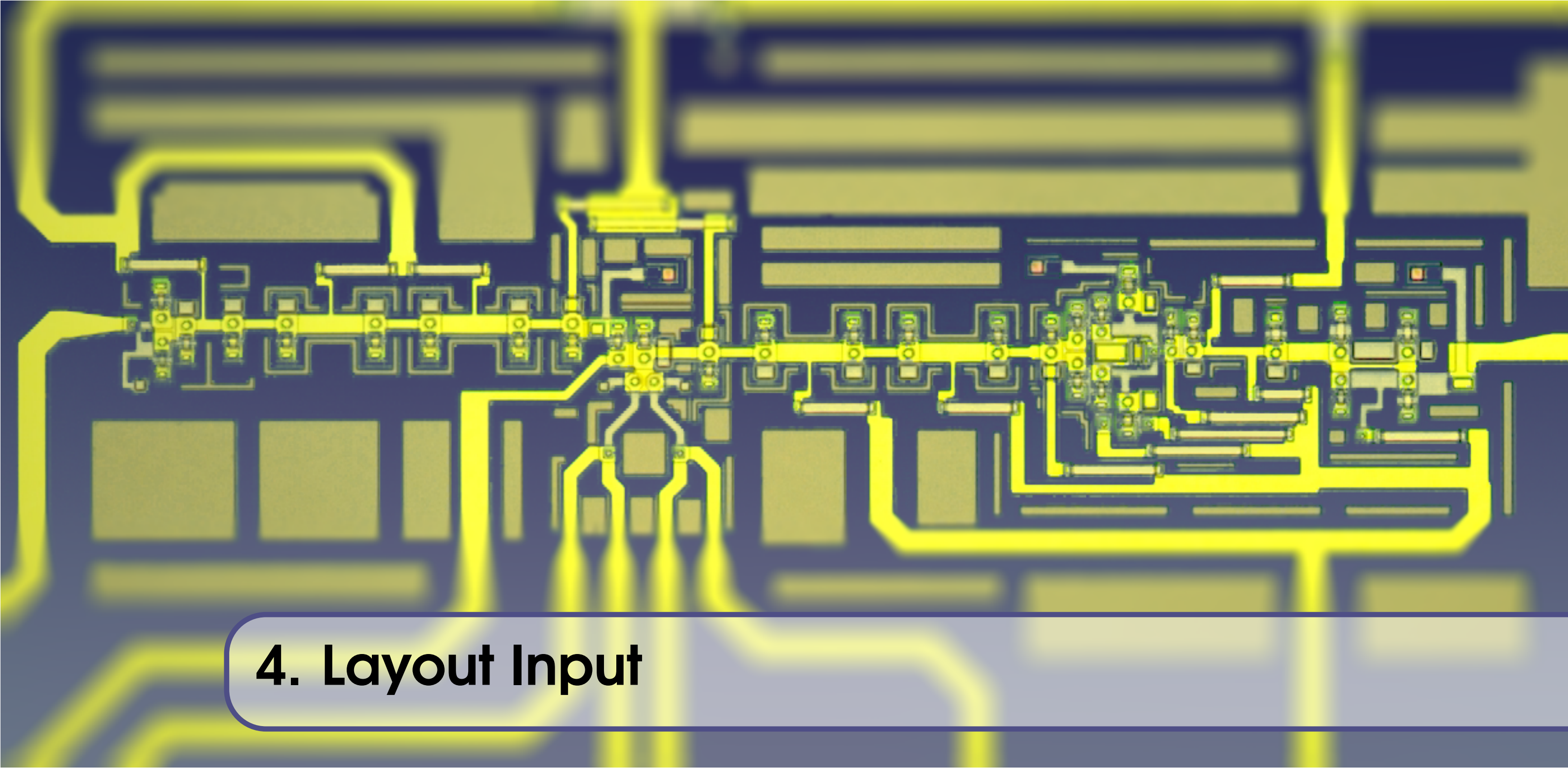
EC Electrical connection – Connects nodes on the layers defined in LayersConnect electrically (without segments), and removes objects in layers defined by LayersRemove within which operator text labels fall. Electrical connections are only made between nodes falling inside the polygon with smallest area in any of the layers defined by LayersRemove.

LD Layer delete – Removes all objects in layers defined by LayersRemove. Useful to get rid of excess structures, such as multiple ground planes when these do not influence extraction results.

MR Make rectangle – Changes all polygons in the layers defined by LayersRem and within which the operator text label falls to rectangles.

OD Object delete – Removes objects in layers defined by LayersRemove within which operator text labels fall. Useful to get rid of excess structures, such as multiple ground planes when these do not influence extraction results.

UD Undefined – does nothing



4. Layout Input

InductEx was originally developed to processes structures in the GDSII stream file format (more commonly referred to just by the file extension GDS). This is an industry standard binary format, and most layout tools can convert layouts to GDS. This format is still useful for layout entry, although more functionality is available with the *InductEx*-specific IXI input file format.

4.1 The IXI file format

4.1.1 IXI description

The IXI input file format is text based and human readable. It was developed to allow an easy interface from Cadence Virtuoso while maintaining readability, but has evolved to allow more complicated geometry read-in that what is possible with GDS files.

The legacy IXI format uses a flat hierarchy with a limited object set, and was deprecated with the release of *InductEx* v6. Legacy files can still be processed if *InductEx* is called with the `-leg` switch.

The IXI file has evolved for *InductEx* v.7 so that it can now represent more than just a die layout, but also act as a project file that combines multiple layouts, external geometries, materials, wire bonds and bump bonds to model a complex multi-chip or chip-and-package environment.

Record identifiers in an IXI file are preceded by the `$` character, and record declarations are terminated with an `$End` command.

The IXI input file is read in partially sequentially. A variable has to be declared before it is used, but materials and bond properties can be specified anywhere. The limitation arises from the ability of variables to be used in or referenced from other variables.

4.1.2 IXI units

The IXI input file uses 1 m as the unit for any coordinate or dimension entry outside of a `STRUCT` record.

All coordinates inside `STRUCT` records are in Unit linked to the associated process technology file. The default unit is 1 µm.

Rotation angles in the IXI input file are always specified in degrees.

4.1.3 IXI records and fields

An IXI file consists of a collection of records, each with a number of fields and values. Some records can also contain other records. The records are:

Ball The definition of the ball geometry parameters for a ball-and-stitch wire bond.

The `MeshSize` field sets the mesh size of the ball. `BallDiameter` sets the centre diameter of the ball, while `PadDiameter` sets the diameter of the ball foot where it bonds to a pad. `BallHeight` sets the height of the ball from bad to wire joint. The top diameter is determined by the wire parameters. The field values represent the dimensions in metres, and are real-valued. The fields define the ball geometry as shown in Fig. 7.5.

The `BallSegments` field takes an integer value to set the the number of segment rings between the ball foot and the top of the ball.

The `Name` field takes a text string. This is the name by which the `Ball` record is referenced in the `WireBondSetup` record.

`$BALL`
  `BallDiameter` *real value or expression*
  `BallHeight` *real value or expression*
  `BallSegments` *integer*
  `MeshSize` *real value or expression*
  `Name` "*text*"
  `PadDiameter` *real value or expression*
`$END`

Bump The declaration of a bump bond for a flip chip interconnect.

The `BondSetup` field references the name of the `BumpSetup` record that defines the bump parameters.

The `StartPin` and `EndPin` fields identify the start and end pins for the bump bond. The pins must be on different dice, each imported with `Die` and with one die flipped 180 degrees around the x or y axis. If pin locations are not aligned in the z-direction, the bump centre is placed at the midpoint of the x and y coordinates of the start and end pins. The pin names are specified in the field value as *diename.pinname*.

`$BUMP`
  `BondSetup` "*text*"
  `EndPin` "*text.text*"
  `StartPin` "*text.text*"
`$END`

BumpSetup The definition of the bump bond parameters for flip chip interconnects.

The `MeshSize` field sets the mesh size of the bump bond structure. `Diameter` sets the centre diameter of the bump, while `PadDiameter` sets the diameter of the bump foot on either end where it bonds to a pad. The field values represent the dimensions in metres, and are real-valued.

`CircumferenceSections` sets the the number of segments around the circumference of the bump.

`LengthSections` sets the the number of segment rings between the bump lower and upper contact surface of the bump.

The `Name` field takes a text string. This is the name by which the `BumpSetup` record is referenced in the `Bump` record.

`$BUMPSETUP`
  `CircumferenceSections` *integer*

`Diameter` *real value or expression*
`LengthSections` *integer*
`Material` "*text*"
`MeshSize` *real value or expression*
`Name` "*text*"
`PadDiameter` *real value or expression*
`$END`

Die A layout in GDSII or IXI format for importation as a separate die into the layout model. The die must have an associated layer definition file to define the process. Any number of dice can be imported, as long as they do not overlap during model building.
The `Layout` field takes a text string that specifies the name (and path, if needed) of the layout input file, which can be of type GDS or IXI.
The `Name` field takes a text string. This is the name by which the `Die` record is referenced when a wire bond or bump bond is declared in a `Bump` record or `WireBond` record.
The `Process` field takes a text string that specifies the name (and path, if needed) of the layer definition file that describes the fabrication process for the die.
The `Rotate` and `Translate` fields each take a single real-valued coordinate triple as value. Coordinate triples are placed in parentheses and read from consecutive lines (one triple per line), so that the coordinates are on new lines. If more than one coordinate triple is specified for these fields, only the last is used. Objects on the die are rotated and translated as specified by the `Rotate` and `Translate` fields.
Translation coordinates are specified in metres and rotation angles are in degrees using the right-hand rule. Rotation is applied strictly around the x-axis, then the y-axis and finally the z-axis. Note that three-dimensional rotation is not commutative, so that the order is important.
Labels to identify ports, fluxon positions and bond points can be specified in an optional port label file or included in the GDS layout.

`$DIE`
`Layout` "*text*"
`Name` "*text*"
`PortFile` "*text*"
`Process` "*text*"
`Rotate`
(R_x, R_y, R_z)
`Translate`
(T_x, T_y, T_z)
`$END`

FPlane A 2D plane used for field calculations. The plane is defined as a rectangle between opposite corners (x_1, y_1) and (x_2, y_2) in the xy plane at $z = 0$ and then rotated and translated as specified. This record is called outside of a `Struct` record, so that all coordinates (position and translation) are in metres. Rotation angles are in degrees. z-axis coordinates in `Coords` are ignored.
The `Coord`, `Rotate` and `Translate` fields each take real-valued coordinate triples as values. Coordinate triples are placed in parentheses and read from consecutive lines (one triple per line), so that the coordinates are on new lines. `Coord` takes exactly two coordinate triples. If more are specified, only the first two are used. If fewer are specified, the field plane is discarded. For the `Rotate` and `Translate` fields, only the last coordinate triple is used if more than one is specified.

`$FPLANE`

Coords
$(x_1, y_1, 0)$
$(x_2, y_2, 0)$
Rotate
(R_x, R_y, R_z)
Translate
(T_x, T_y, T_z)
$END

Gds A GDS layout file in GDSII format from which layout objects are imported into the main (primary) layout. The `GDS` record can only be used inside the main `Struct` record before any other objects are defined, because it rewrites the structure array. Other structures and objects can be defined after this importation. If multiple GDS files are specified, the last one will be imported. In this version of *InductEx*, rotation and translation of the imported layout objects from the GDS file are not supported.

Note that this record is not used to import an entire chip layout as part of a multi-chip model (that is done with `Import`), but rather to add layout objects to single layout. It is retained for legacy purposes, but application is severely limited.

$GDS
Name "*text*"
$END

Import A Gmsh geometry file for importation, with all dimensions in Unit metres (the default is 1 metre).

The `Name` field specifies the path and name of a .GEO file to import.

Upon read-in all point coordinates are multiplied with `Scale` (which is real-valued) and then rotated and translated as specified. Translation coordinates are specified in metres and rotation angles are in degrees using the right-hand rule and rotated strictly around the x-axis, then the y-axis and finally the z-axis.

`MeshSize` takes a real value and is specified in metres, but is optional. If it is declared, `MeshSize` sets the mesh size for all physical objects in the imported geometry file. If undeclared, the characteristic length (which determines mesh size) of every point in the geometry is retained, but adjusted for Unit and Scale.

Any number of terminals can be linked to a physical surface in the geometry file, each declared with a `Terminal` field, followed by two values (each a text string). The first value string is the terminal label, which should end in a "+" or "-" character to indicate terminal polarity. The second value string is the surface label that must match the name of a physical surface in the geometry file.

All structures must be declared as `Physical` records inside the `Import` record. Material properties are defined in separate `Material` records and linked inside the `Physical` records. There is no limit on the number of physical objects in an imported model.

$IMPORT
MeshSize *real value or expression*
Name "*text*"
Rotate
(R_x, R_y, R_z)
Scale *real value or expression*
Terminal "*text*" "*text*"
Translate
(T_x, T_y, T_z)
Unit *real value or expression*

```
    $PHYSICAL
      ...
    $END
  $END
```

Mask A mask object, defined outside of the main `Struct` record, that defines a mask area as a polygon. Masks are used to mask out the layer-linked segment size of objects and to enforce a specified mesh size over masked area. There is no limit on the number of masks defined. Where masks overlap, the last defined mask applies.

All dimensions in the `Mask` record are in Unit metres, as specified in the layer definition file of the process used for the layout. By default, Unit is 1 µm.

`MeshSize` sets the mesh size for all nodes inside the mask.

The `XY` field takes two-dimensional coordinate pairs as values. Coordinates are real values or expressions. Coordinate pairs are placed in parentheses and read from consecutive lines (one pair per line). There is no limit to the number of coordinate pairs.

`$MASK`
  `MeshSize` *real value or expression*
  `XY`
  (x_1, y_1)
  (x_2, y_2)
  ...
  (x_N, y_N)
`$END`

Material A material record that defines material parameters for a physical structure. Multiple physical structures can reference the same material.

The `Name` field takes a text string. This is the name by which the `Material` record is referenced when material is assigned in a `Physical` record.

The `EpsRIn`, `Lambda`, `MuR` and `Sigma` fields take real values or expressions.

`EpsRIn` is the relative permittivity or dielectric constant.

`Lambda` is the London penetration depth in metres.

`MuR` is the relative permeability.

`Sigma` is the bulk conductivity in Ωm or S/m.

`$MATERIAL`
  `EpsRIn` *real value or expression*
  `Lambda` *real value or expression*
  `MuR` *real value or expression*
  `Name` "*text*"
  `Sigma` *real value or expression*
`$END`

ParamString A parameter string that may span multiple lines:

`$PARAMSTRING`
  text string that
  may span several lines
`$End`

Path A path or wire. A `Path` record is only processed if it is defined inside a `Struct` record, because it defines a layout object linked to technology file.

The `Layer` field takes a text string which must match a layer name defined in the layer definition file for the process linked to the layout. It is case insensitive.

The `Text` field is optional. If specified, it takes a text string which is linked to the layout path for the purpose of defining a port terminal.

The `Width` field takes a real values or expressions and sets the path width in Units as defined in the layer definition file for the process linked to the layout.

The `XY` field takes two-dimensional coordinate pairs as values (real or expressions). Coordinate pairs are placed in parentheses and read from consecutive lines (one pair per line). There is no limit to the number of coordinate pairs. Coordinate values are in Units as defined in the layer definition file for the process linked to the layout.

Note: Paths in the IXI format conform to PathType 0 of the GDSII format, so that the there is no structure extending beyond the start and end nodes.

`$PATH`
  `Layer` "*text*"
  `Text` (*Optional*) "*text*"
  `Width` *real value or expression*
  `XY`
  (x_1, y_1)
  (x_2, y_2)
  ...
  (x_N, y_N)
`$END`

Physical A physical record, only processed inside an `Import` record, that defines the physical object in a .GEO file to import.

The `Name` and `Material` fields take text strings as values. The value of `Name` must match that of a physical object in the .GEO file specified in `Import`, and the value of `Material` must match that the name of a material defined anywhere in the IXI file.

`$PHYSICAL`
  `Material` "*text*"
  `Name` "*text*"
`$END`

Poly A polygon. The last vertex must equal the first, except when only two vertices are defined to indicate a collapsed box for port or terminal definition. A `Poly` record is only processed if it is defined inside a `Struct` record, because it defines a layout object linked to technology file.

The `Layer` field takes a text string which must match a layer name defined in the layer definition file for the process linked to the layout. It is case insensitive.

The `Text` field is optional. If specified, it takes a text string which is linked to the layout path for the purpose of defining a port terminal.

The `XY` field takes two-dimensional coordinate pairs as values (real or expressions). Coordinate pairs are placed in parentheses and read from consecutive lines (one pair per line). There is no limit to the number of coordinate pairs. Coordinate values are in Units as defined in the layer definition file for the process linked to the layout.

`$POLY`
  `Layer` "*text*"
  `Text` (*Optional*) "*text*"
  `XY`
  (x_1, y_1)
  (x_2, y_2)
  ...
  (x_N, y_N)
  (x_1, y_1)
`$END`

Sref A reference to a structure inserted at the `XY` coordinates. The structure is reflected around the *x*-axis before angular rotation if `Reflect` is *true*.

The `Angle` field takes a real value or expression. Angular rotation is measured in degrees and applied counterclockwise.

the `Name` field takes a text string that must match a defined structure name, and the `Reflect` field takes a Boolean text (case insensitive) of either `true` or `false`.

The `XY` field takes a two-dimensional coordinate pair as value. The coordinates are real values or expressions, and are specified in Units as defined in the layer definition file for the process linked to the layout. Coordinates are placed in parentheses and read from consecutive lines (one pair per line). *InductEx* halts when no coordinate pair is specified, and uses the first coordinate pair if more than one pair is specified.

`$SREF`
  `Angle` *real value or expression*
  `Name` "*text*"
  `Reflect` *boolean*
  `XY`
  (*x*, *y*)
`$END`

Struct A structure with `Name` *name* containing any number of `GDS`, `Poly`, `Path`, `Sref` and `Text` records:

`$STRUCT`
  `Name` "*text*"
  Collection of GDS, PATH, POLY, SREF, TEXT records
`$END`

Text A text label.

The `Layer` field takes a text string which must match a layer name defined in the layer definition file for the process linked to the layout. It is case insensitive.

The `Text` field takes a text string that is applied as the label text. If the text string contains spaces, as with a port definition that specifies name and layer(s), the string must be enclosed in double quotation marks.

The `XY` field takes a two-dimensional coordinate pair as value. The coordinates are real values or expressions, and are specified in Units as defined in the layer definition file for the process linked to the layout. Coordinates are placed in parentheses and read from consecutive lines (one pair per line). *InductEx* halts when no coordinate pair is specified, and uses the first coordinate pair if more than one pair is specified to anchor the text label.

`$TEXT`
  `Layer` "*text*"
  `Text` "*text*"
  `XY`
  (*x*, *y*)
`$END`

Variable A variable a name and a numerical or expression value:

`$VARIABLE`
  `Name` "*Text*"
  `Value` *real value or expression*
`$End`

Wedge The definition of the wedge geometry parameters for a wedge-wedge wire bond, or the stitch geometry for a ball-and-stitch wire bond.

The `Name` field takes a text string. This is the name by which the `Wedge` record is referenced in the `WireBondSetup` record.

The `MeshSize` field sets the mesh size of the wire bond wedge structure.
All the other fields take real values or expressions to define the wedge geometry as shown in Figs. 7.3 and 7.4.

`$WEDGE`
- `FootChamfer` *real value or expression*
- `FootHeight` *real value or expression*
- `FootLength` *real value or expression*
- `FootWidth` *real value or expression*
- `MeshSize` *real value or expression*
- `Name` "*text*"
- `Throat` *real value or expression*
- `ToeHeight` *real value or expression*
- `ToeLength` *real value or expression*

`$END`

WireBondSetup The definition of wire bond parameters.

The `StartType` field takes a text string that selects the type of bond foot. Currently, only "ball" and "wedge" are supported ("wedge" is the default).
The `EndType` field takes a text string that selects the type of bond foot. Currently, only "wedge" is supported (this is also the default).
The `StartBond` and `EndBond` fields take text strings that must match the name of a defined `Wedge` record (or a defined `Ball` record in the case of `StartBond`).
The `MeshSize` field sets the mesh size of the wire part of the wire bond structure. `Diameter` sets the diameter of the wire. `VertexHeight` sets the wire bond arc above the highest foot for a wedge-wedge wire bond, and the maximum height of the wire above the ball top for a ball-stitch wire bond. `BendRadius` sets the radius of both bends in the wire of a ball-stitch wire bond, while `TopLength` sets the length of the horizontal section of wire between the two bends of a ball-stitch wire bond. The field values represent the dimensions in metres, and are real-valued.
`CircumferenceSections` sets the the number of segments around the circumference of the bump.
`LengthSections` sets the the number of segment rings in the wire arc of a wedge-wedge wire bond, thus between the start and end feet. For a ball-stitch wire bond, `LengthSections` sets the number of segments in each of the two bends in the wire.
The `Name` field takes a text string. This is the name by which the `WireBondSetup` record is referenced in the `WireBond` record.
The `Material` field takes a text string, which must match that the name of a material defined anywhere in the IXI file.

`$WIREBONDSETUP`
- `BendRadius` *real value or expression*
- `Diameter` *real value or expression*
- `EndBond` "*text*"
- `EndType` "*text*"
- `LengthSections` *integer*
- `Material` "*text*"
- `MeshSize` *real value or expression*
- `Name` "*text*"
- `StartBond` "*text*"
- `StartType` "*text*"
- `TopLength` *real value or expression*

```
    VertexHeight real value or expression
  $END
```

WireBond The declaration of a wire bond interconnect.

The `BondSetup` field references the name of the `WireBondSetup` record that defines the wire bond parameters.

The `StartPin` and `EndPin` fields identify the start and end pins for the wire bond. The pins can be on the same die or on different dice, each imported with `Die`. The die surfaces must have the same normal. The pin names are specified in the field value as *diename.pinname*.

```
  $BUMP
    BondSetup "text"
    EndPin "text.text"
    StartPin "text.text"
  $END
```

4.1.4 IXI variables and expressions

IXI variable names are not case sensitive. Variables must be defined before they are used, so as a rule it is advisable to define variables at the top of an IXI file.

Expressions are defined with variables identified by the % symbol. Such expressions can be evaluated as long as only the standard arithmetic operators (+, -, *, /) and parentheses are used. For instance, if variables `d`, `w` and `scale` have been defined, the following are examples of valid expressions:

```
( %d , -%d )
( %d + %w , -%d + %w )
( %d * (%scale+%d) )
```

4.1.5 IXI import of GEO objects

In order to allow off-chip objects to be combined with planar integrated circuit layout structures, *InductEx* supports read-in of Gmsh .geo geometry files with a heavily limited syntax.

Each .geo geometry file should preferably describe a single three-dimensional object, such as a shield canister or a coil, although multiple physical objects in a single geometry file is supported.

The dimensions in the .geo file are read in metres, unless a different unit is specified (such as `1e-6` to transform dimensions to µm). A scaling factor can be also specified. *InductEx* multiplies the scaling factor with the unit length.

InductEx recognises:

- Point
- Line
- Line Loop
- Curve Loop
- Plane Surface
- Surface Loop
- Volume
- Physical Surface
- Physical Volume

The coordinates and characteristic length of a Point may be expressions containing variables.

Irrespective of what Gmsh accepts, *InductEx* only recognises a variable in a .geo file if the variable name starts with the "_" character. Variable definitions are also limited to a single variable per line.

Expression evaluation is supported for variable values, point coordinates and characteristic lengths. The following functions in Gmsh expressions are recognised by *InductEx*:

- Cos
- Sin

Physical surfaces are used to provide external links to specific surfaces to be used as terminals for ports.

Physical volumes are used to identify the objects that should be meshed and included in a model for *InductEx* .

Material properties such as penetration depth, conductivity and permeability can be linked to specific physical volumes.

4.2 GDSII

For GDSII input files, *InductEx* only reads objects with DATATYPE = 0 and text labels with TEXTTYPE = 0.

If, for any reason, DATATYPE or TEXTTYPE values other than zero should be processed, you may set the DataTypeNotZero parameter in the layer definition file to TRUE.

4.2.1 Limitations of GDSII support

InductEx does not read in Node Elements from GDSII files.

4.3 Port label input file

When a port label input file (in ASCII text format) is specified with the `-p` parameter, port labels, fluxon positions and bond points (both text and coordinates) are read in from this file. These text labels are added to any port, fluxon or bond position labels that may have been read in from the geometry input file. The file may have any suffix, although `.txt` is preferable.

Port declarations follow the rules set out in Section 6, and the file syntax is:

P*label*; (x, y)

where *Plabel* is the port label that contains the port name and positive and/or negative terminal layers as described in Section 6. The label coordinates within the layout are at x and y, which is in `Units` as defined in the layer definition file.

Examples of valid definitions in a port label input file are:

`P1 M2 M0; (0, 25.5)` - Port P_1 between layers M2 and M0, at $x = 0$; $y = 25.5$.
`PIN+ M2; (10, 20)` - Port P_{IN}, with positive terminal on M2, at $x = 10$; $y = 20$.
`#BondPin1; (1500, -200)` - Bond point at $x = 1500$; $y = -200$.

4.4 Modelling aids defined on layouts

Some modelling aids are available to prune or simplify circuit structures during modelling, mostly to reduce the demand on computing resources.

4.4.1 Extraction boundary

InductEx allows you to specify an extraction boundary for a die, outside of which all structures are ignored. This is helpful in large circuit layouts where the structures to be extracted do not span the entire layout. With an extraction boundary, structures outside the area of interest are not discretized, which saves calculation time and memory.

The extraction boundary also makes it easy to separate isolated sections of a negative mask layer (mostly ground planes in current recycling circuits) during modelling.

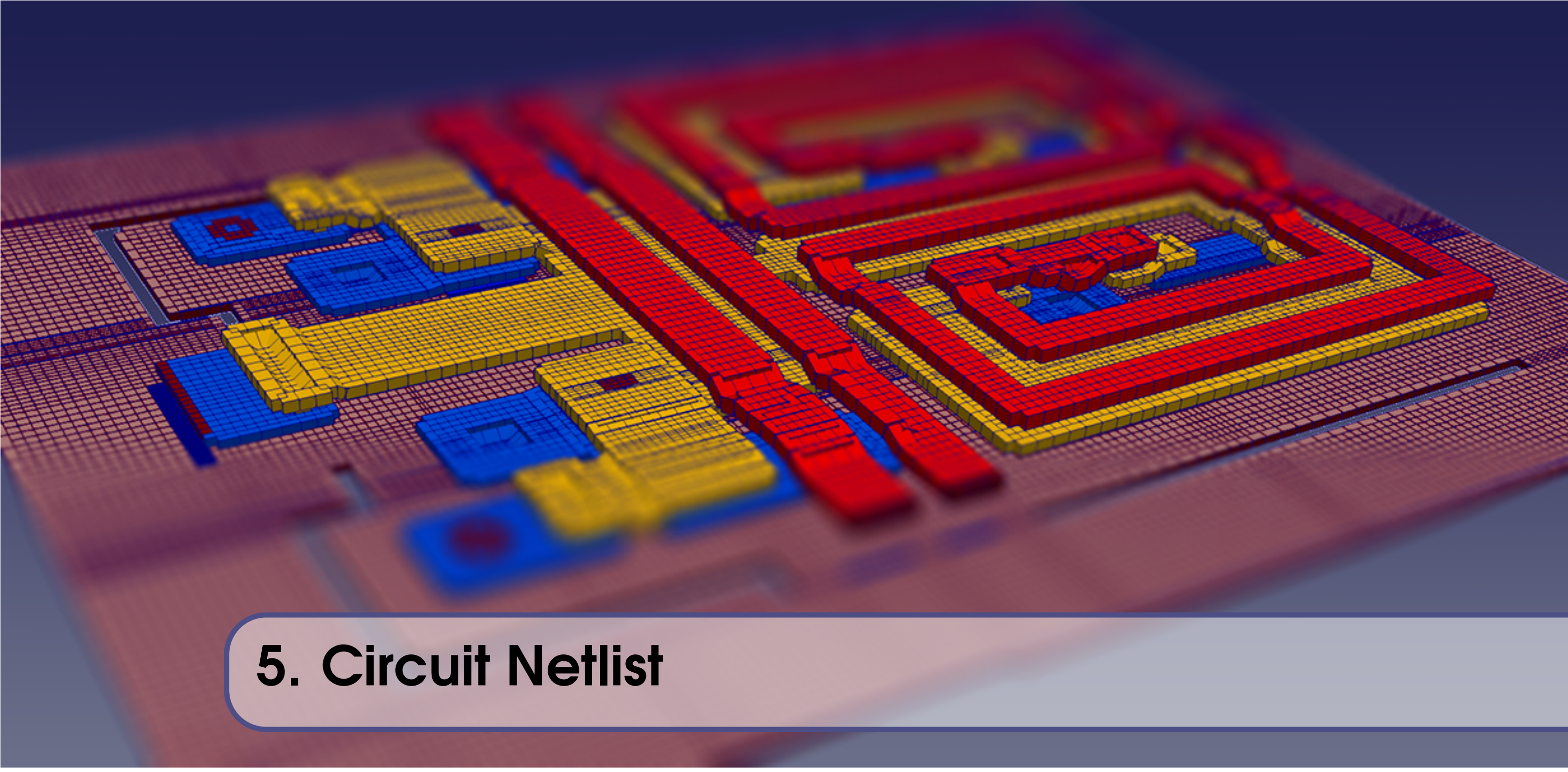
The extraction boundary is defined on the layout in a special non-fabrication layer which is set with the `ExtractBoundLayer` global parameter in the layer definition file. Any combination of layout objects may be used – it is thus possible to define several isolated extraction zones.

4.4.2 Operator declarations

Operators are declared with text labels on any layer. Operators act at the coordinates of the text label. Syntax is:

`@`*Name*

with the first character always "`@`". The name of the operator must match that of an operator declared in the layer definition file, although the name is case-insensitive. If no match is found, the operator is ignored.



5. Circuit Netlist

For most inductance calculation problems with *InductEx* a circuit netlist file is needed. This netlist file is in part a subset of the SPICE electrical netlist. The netlist should contain all the inductors and mutual inductors in the circuit model, and the values assigned to each will be used as the "design value" during post-processing.

5.1 Title line

An *InductEx* netlist file does not start with a title line. If you use a title line, start it with a comment marker.

5.2 Comments

If a line in a netlist file starts with "*", *InductEx* discards the line as a comment.

In-line comments are marked with "//". All text after the marker until the end of the line is discarded as a comment. A comment cannot be closed.

5.3 Circuit elements

Inductors, resistors and mutual inductance (or coupling elements) are supported.

Elements are defined as:

L*name* *node+* *node-* *designvalue* <[*back-annotate*]> <coupling=off>
R*name* *node+* *node-* *designvalue* <[*back-annotate*]>
K*name* *Lname1* *Lname2* *designcoupling* <[*back-annotate*]>
M*name* *Lname1* *Lname2* *designvalue*

Node names may be any alphanumeric combinations excluding whitespace characters such as SPACE and TAB.

All design values are optional. If these are not specified, *InductEx* only prints the extracted values and does not list any comparison between design and extracted values. Design values are in the units specified in the layer definition file by parameters UnitI, UnitL or UnitR.

The back-annotate assignment, where name of the circuit element to which the extracted value must be assigned in the back-annotated file is enclosed in square brackets, is also optional. If it is not defined, the specific parameter value will not be written to a back-annotated file.

An optional `coupling=off` parameter excludes an inductor from coupling calculations when external magnetic fields are applied.

Coupling for mutual inductance is specified with `K` for compatibility with SPICE, although `M` can also be used if SPICE compatibility is not required. Back-annotation is only possible for a `K` element.

Holes are never defined in the circuit netlist file – *InductEx* adds components for holes automatically when these are specified with flux labels on the layout.

5.4 Ports

The netlist should contain port definitions for the extraction setup. Port definitions on layouts are discussed in Section 6. Some valid definitions in the netlist file are:

```
P1        1       0
Ptwo      N1      N2
PAny_name posterm ground
J3        node1   node2   250
B1        node1   node2   100E-6
Ibias     node+   node-   source=i   amplitude=0.5E-3
```

Port identifiers are the symbols `P`, `J`, `B` and `I`. These can be used in any way, but it allows users to label junctions and bias current sources more clearly.

If a numerical value is listed after the node names, and the port is defined in a via, the value is used as the design value for the via area. For a Josephson junction, it is possible to extract the junction critical current from the layout size and the trilayer's `IDensity` parameter in the layer definition file.

The source type of a port is by default assumed to be a voltage source with amplitude of 1 V. This can be altered for specific analyses, such as finding current distribution in a circuit, by specifying the source type is current (`i`). The amplitude is also controllable.

5.5 Control block

A control block may be included in the netlist file, and can be used to assign values to certain variables in order to exert control over modelling and solution methods. An example of the use of the control block is:

```
.control
  SET RCond = 1e-16
  SET AbsTol = 1e-6
.endc
```

Control block statements contain a command (`SET`, `SWEEP`), parameter name, equal sign and a value or range of values. All parameter values assigned in the control block override the values assigned in the layer definition file.

5.5.1 BACK-ANNOTATE command

For compact SPICE model extraction is back-annotation of calculated parameters to a SPICE-based simulation deck. The circuit netlist file used by *InductEx* is stripped of components such as Josephson junctions and current, voltage or phase sources, and is populated with excitation ports. The circuit netlist is often also stripped of resistive components. The extraction circuit netlist is thus practically never the same as the SPICE simulation deck.

With back-annotation, *InductEx* writes extracted parameter values directly to the SPICE simulation deck. The target deck file is nominated in the control block in the extraction circuit netlist file with the `BACK-ANNOTATE` command. The target deck file has to exist, and any parameters have to be identified in sqaure brackets in the extraction netlist, as shown in Section 5.3.

If the target simulation netlist is a subcircuit in a larger simulation deck, the subcircuit can be identified too. Syntax is:

`BACK-ANNOTATE` *targetfilename*
`BACK-ANNOTATE` *targetfilename* `SUBCKT` *targetsubcircuitname*

As a safety precaution, the target netlist file is never overwritten but a new file, with the string `"_annotated"` appended to the target file name, is created.

5.5.2 SET command

Inside the control block, global and layer parameters can be assigned new values with the `SET` command.

Syntax is:

`SET` *parametername = newvalue*
`SET layer` *layername parametername = newvalue*

Global parameters currently supported are: `frequency`, `gpoverhang`, `lambda0`, `lambdafit`, `rcond`, `segmentsize`, `tc`, `temperature`, `uniti`, `unitl` and `unitr`.

Layer parameters supported are: `bias`, `hfilaments`, `lambda`, `lambda0`, `lambdafit`, `segmentsize`, `sigma`, `mu_r`, `tc`, `temperature` and `thickness`.

Solution stability parameters supported are: `rCond` and `nonZeroSVThreshold`.

Examples are:

```
SET gpOverHang = 5.0
SET segmentsize = 1.5
SET layer M1 lambda = 0.085
```

5.5.3 SWEEP command

Inside the control block, parameter sweeps for global and layer parameters can be defined with the SWEEP command. Single and double parameter sweeps are supported, and results are written to a Matlab-compatible M-file for easy visualization.

Syntax examples are:

`SWEEP` *parametername startvalue incrementvalue stopvalue*
`SWEEP layer` *layername parametername startvalue incrementvalue stopvalue*
`SWEEP` *parameter1 start1 inc1 stop1 parameter2 start2 inc2 stop2*
`SWEEP variable` *variablename startvalue incrementvalue stopvalue*

`SWEEP variable` *var1* *start1* *inc1* *stop1* `variable` *var2* *start2* *inc2* *stop2*

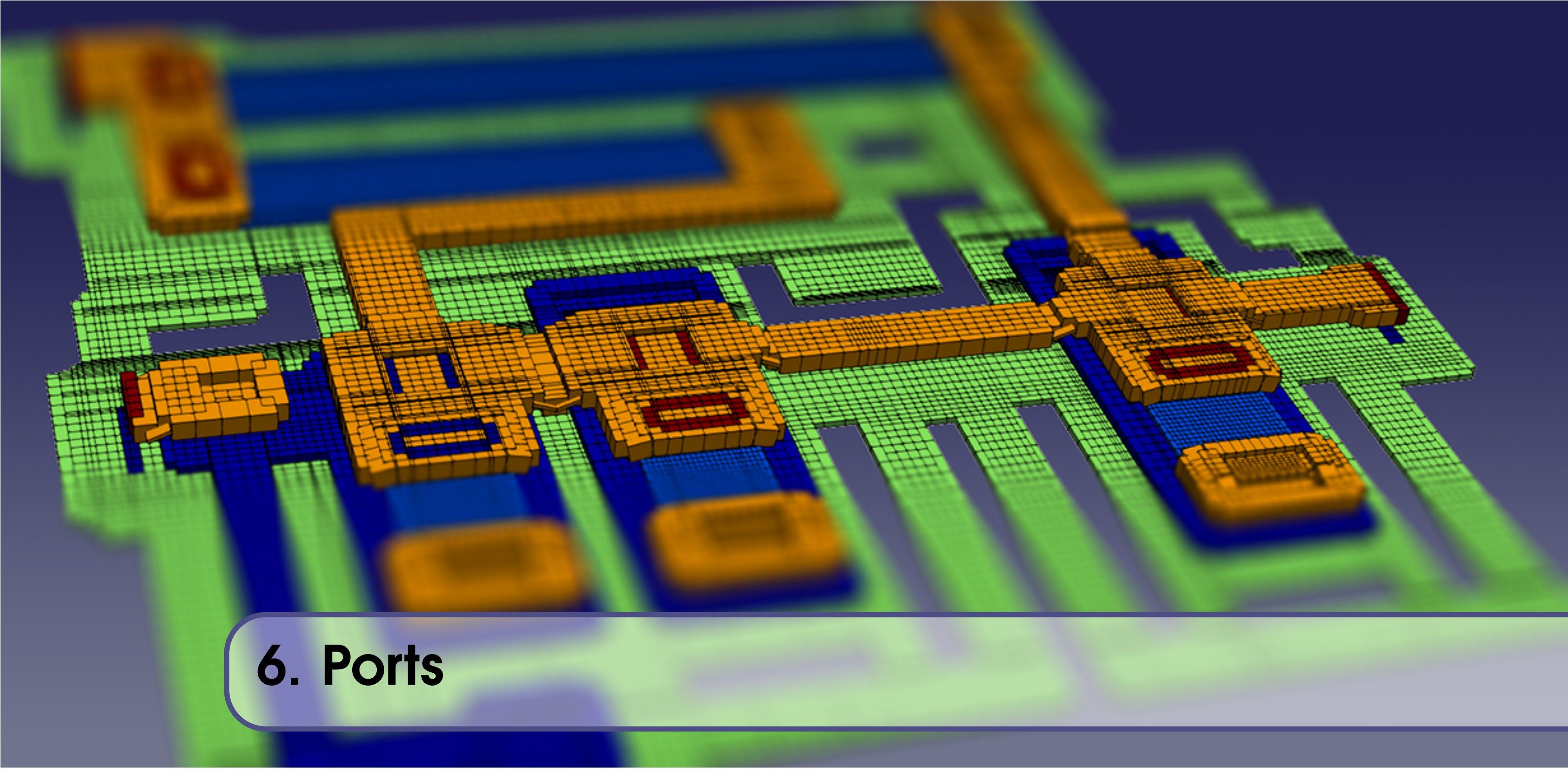
Global parameters currently supported are: `frequency`, `gpoverhang`, `segmentsize` and `temperature`.

Layer parameters supported are: `bias`, `segmentsize`, `hfilaments`, `lambda`, `sigma`, `mur` and `thickness`.

Variables used in `SWEEP` commands must be defined in an .IXI input file as described in Section 4.1.

Examples are:

```
SWEEP frequency 1e10 1e10 1e11
SWEEP layer M1 segmentsize 2.0 -0.2 1.0
SWEEP layer M1 lambda 0.07 0.01 0.1 layer M1 bias -0.2 0.1 0.2
SWEEP variable d 10 1 20 variable w 5 5 10
```



6. Ports

6.1 Ports for inductance extraction

6.1.1 Definition of ports and terminals

For inductance extraction, port definition is a very important aspect of modelling. A good understanding of how ports are used in the extraction process and how port definitions propagate to extraction models in *InductEx* is thus necessary for accurate modelling.

For basic inductance calculation, voltage ports are defined in a model, as shown in Fig. 6.1. These ports are excited with a nominal voltage of 1 V at the specified frequency.

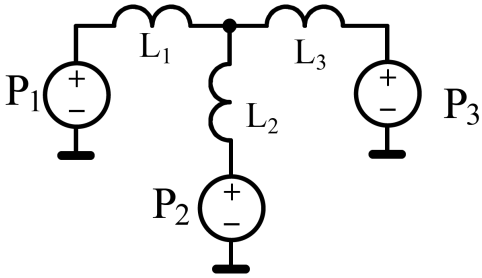


Figure 6.1: Example of how ports (nominally voltage sources) are connected in *InductEx* models for parameter extraction.

A port must have two terminals – positive and negative – but each terminal can be connected to more than one terminal object. Such terminal objects are defined on the TERM layer.

6.1.2 Terminal geometries of inductance ports in integrated circuit models

InductEx is primarily aimed at integrated circuit structures. The standard definition of a port is thus a single TERM object that is applied to every layer to which the port is connected. Such a vertical stack may have multiple layers assigned to each terminal.

Fig. 6.2 shows an illustration of several types of inductance port definitions on layouts processed with *InductEx*.

Ports P_{sl1} and P_{sl2} are defined with a wire/path object on the TERM layer that exactly coincides with the edge of stripline conductor in layer M1. The TERM layer object should be a single line segment with the end vertex coordinates matching the vertices of the stripline conductor.

Port P_{via1} does not need a terminal object, and can be used for vias or similar structures such as planar Josephson junctions. The smallest via inside which the port label coordinates fall is elevated to the TERM layer and used to define the port geometry on every layer linked to the port. All other vias inside the stack are deleted during modelling. In this example, the positive terminal is connected to layers M3, M2 and M0, while the negative terminal only connects to layer M1.

Port P_{cpw1} is defined with three labels; each connected to a line on the TERM layer at the edge of the coplanar waveguide layout objects. The centre conductor is connected to the positive terminal, while the ground conductors on either side are connected to the negative terminal.

Finally, a polygon port is used for one of the negative terminal objects of Port P_{cpw2}. Since version 6.1, such a polygon may coincide with and edge of the conductor polygon, or exceed the boundaries of the conductor polygon.

For tetrahedral meshing, a polygon port terminal is always defined in the bottom surface of the conductor to which it is linked.

6.1.3 Ports drawn on layouts

In order to provide port/terminal information to *InductEx*, a TERM layer is needed. Any layer can be used, as long as it is called TERM, and the layer number or index corresponds to the parameter `TermLayer` in the layer definition file.

Terminals are drawn as polygons, boxes or paths in the terminal layer. It is not necessary to draw terminals over vias or Josephson junctions, as *InductEx* will automatically use the via or junction boundaries to determine the terminal dimensions. Therefore, it is mostly only necessary to draw terminals as lines on the edges of input/output lines. Collapsed boxes (zero width) can be used to draw line terminals, but these do not conform to proper GDSII protocol and might not be exported correctly by most programmes. **When using GDSII files, it is better to draw line terminals as paths or 2-point polygons** on the TERM layer. The path width is immaterial, as InductEx only uses the centre line as the actual terminal (thereby forcing a zero width path). Path width is therefore only used as a visual aid during layout.

6.1.4 Port or terminal label declarations on layouts

Ports or terminals are declared in the TEXT layer, and their geometries defined in the TERM layer. The coordinates of a text label is used to match the port text to a structure in the TERM layer. The standard syntax is:

P*name* [*postermlayers*] [*negtermlayers*] *value*

P*name*+ [*postermlayers*]

P*name*- [*negtermlayers*]

P*name*/ *termlayer* *direction*

The `P` character is the port identifier. Other valid identifiers are `I` to identify current source and `B` and `J` to identify Josephson junction ports. Hole ports use the `F` identifier.

If the optional suffix characters `+` or `-` are added to the first terminal text, it indicates that this is only one terminal of the port and specifies the polarity (positive or negative). If the suffix is in {`\` `|` `/`}, the port is a virtual cut. [*postermlayers*] is the space-separated list of names of the layers

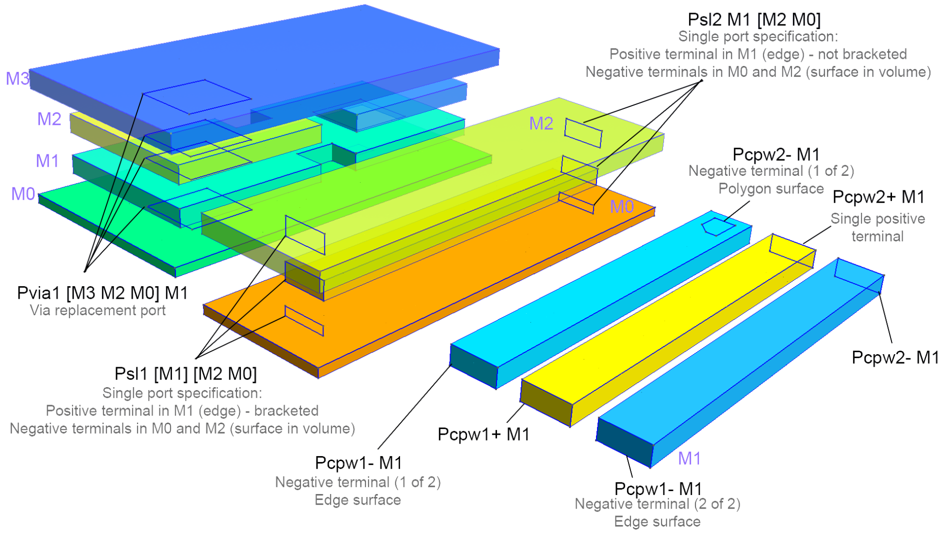


Figure 6.2: Illustration of different methods to define ports with *InductEx*.

on which the terminal (defined by the polarity) is located, or the positive terminal if the polarity is not specified. If polarity is not specified, the list of layer names [*negtermlayers*] is required for the negative terminal. For a virtual cut port, *termlayer* is limited to a single layer, while *direction* defines the axis along which the port is connected to a conductor (x or y, which should coincide with the predominant axis of current flow) and the direction of the positive terminal. The virtual cut opens a slit in the conductive structure on which it is defined, and should run edge-to-edge. The sides of the slit are exactly `SegmentSize`/2 from the centre of the virtual cut line.

Examples of valid labels are:

`P1 M2 M0`	-	Port P_1, with positive terminal in layer M2 and negative in M0.
`PTWO [M1] M0`	-	Port P_{TWO}, with positive terminal in layer M1 and negative in M0.
`P3 M1 [M2 M0]`	-	Port P_3, with positive terminal in layer M1 and negative in M2 and M0.
`P4+ M2`	-	Positive terminal of Port P_4, in layer M2.
`P4- M0`	-	Negative terminal of Port P_4, in layer M0.
`P5/ YBCO +y`	-	Port P_5, a virtual cut through a conductor in layer YBCO, with current flow and port connection along the y axis. The positive terminal is in the positive direction along the y axis.)
`P6/ M3 -x`	-	Port P_6, a virtual cut through a conductor in layer M3, with current flow and port connection along the x axis. The positive terminal is in the negative direction along the x axis.)
`J7 M2 M1`	-	Port J_7, with positive terminal in layer M2 and negative in M1. Typically used for Josephson junctions.
`I8 M2 M0`	-	Port I_8, with positive terminal in layer M2 and negative in M0.
`Fnine M0`	-	Fluxon port F_{nine}, in layer M0.

If the port text label coordinates fall inside or on the boundary of an object in the TERM layer, this is used as the port definition.

When no TERM layer object can be linked to the port label coordinates, and if the positive and negative terminals were defined in the same label, *InductEx* searches for a via object (with `MASK = -1`) that encloses the label coordinates and is ordered between the superconducting layers on which the terminals are declared. If more than one via object satisfies these conditions, the one with the smallest area is used. When such a via is used as a terminal object, it is moved to the TERM layer (and should show up as such when a GDSII output file is generated with the `-d` option). For convenience, it is therefore not necessary to define a terminal layer object over a Josephson junction when it is to be used as a port.

Terminal geometries may be defined anywhere in on the edge or inside of a superconductive layer object.

Currently, *InductEx* also accepts text labels on layers other than the text layer when reading in terminal declarations. However, terminal text must be placed in the top cell of a layout, unless the `-X` switch is used to force text propagation up the cell hierarchy.

For instances where text labels are not supported by a layout tool (such as WaveMaker), or where it is required for convenience to alter port labels without the use of a layout tool, port terminals can be defined in a text file. This is discussed in Section 4.3.

6.1.5 Port definition in external file

Ports can be defined in an external file, as detailed in Section 4.3.

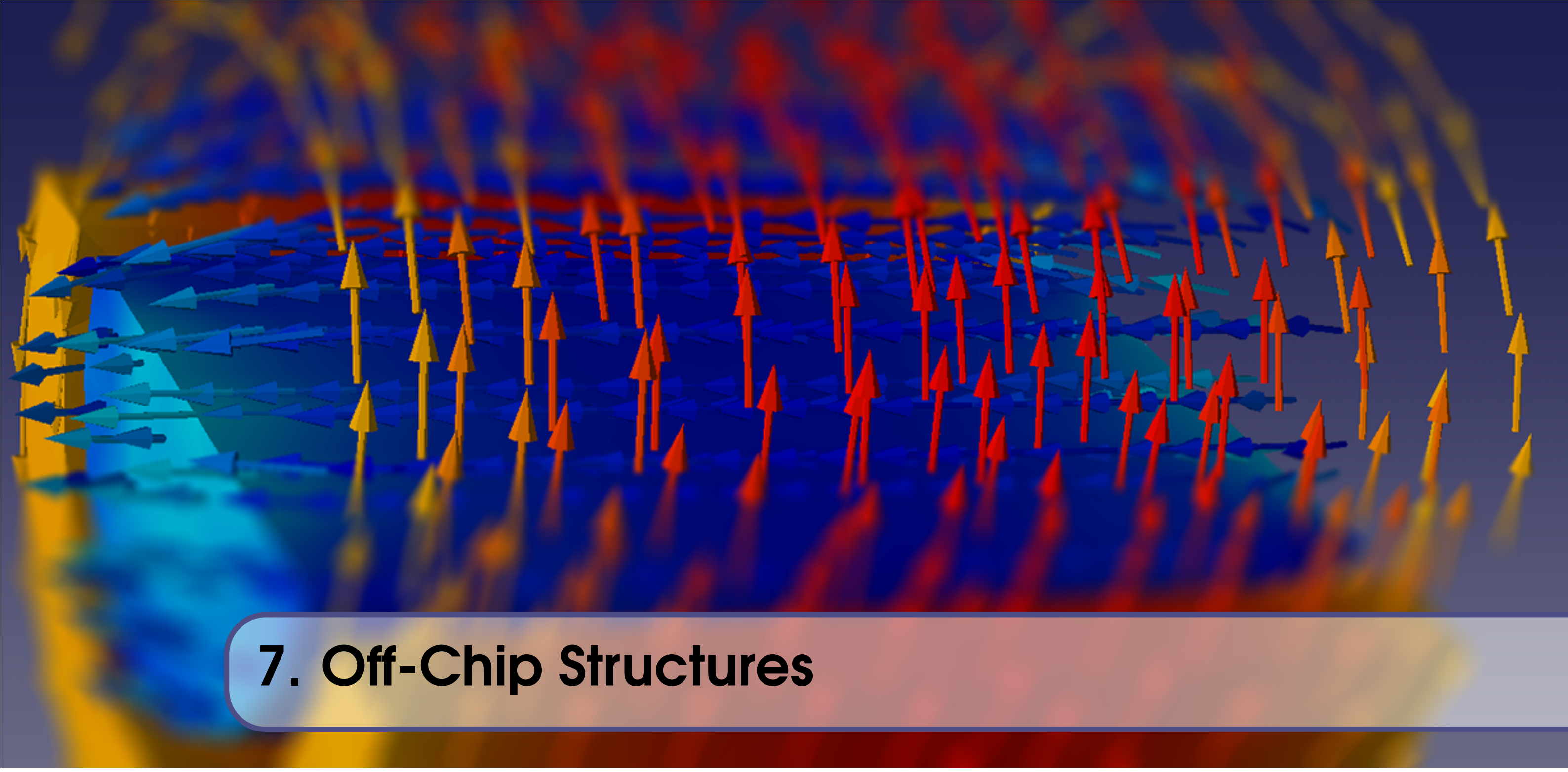
6.2 Conductors (Ports) for capacitance extraction

For capacitance extraction, each conductor is identified with a label on the layout. The label must start with the character "C", followed by any name.

Conductive objects that are linked together in a layout through metal contacts or vias are linked the same capacitance conductor.

6.3 Ports for impedance extraction

Impedance extraction uses ports similar to those used for inductance extraction, but the requirements are more strict. Currently, only two impedance ports are allowed. The first two ports read in by InductEx are used, and the rest discarded. These ports should be on the outer edges of conductive objects. Line terminals are advised.



7. Off-Chip Structures

7.1 Arbitrary structures from Gmsh

For off-chip structures that cannot be defined as layout objects in a planar integrated circuit construction, *InductEx* supports structure read-in from Gmsh geometry files.

In this version of *InductEx* , all off-chip structures are meshed with tetrahedra.

7.1.1 Limitations

InductEx currently only supports a basic Gmsh geometry structure that consists of:

- Point,
- Lines
- Line Loop,
- Curve Loop,
- Plane Surface,
- Surface Loop, and
- Volume.

Physical Surfaces are used to expose surfaces to *InductEx* for the purpose of connecting port terminals.

Physical Volumes are used to define parameters for the Volumes, with only sigma, lambda and relative permeability that can be defined.

The OpenCASCADE engine for constructive geometry is not supported.

7.1.2 Variables

Gmsh supports variable declarations and expression evaluation for the declaration of other variables or coordinates of structures. The use of variables in a the Gmsh `.geo` file allows flexibility when importing structures into *InductEx*, so that a shield geometry can for instance be scaled in length or diameter or plate thickness without the need to build a separate geometry for each instance.

A variable in a Gmsh geometry file can have any name that is not a keyword, but *InductEx* only accepts variables that start with the underscore character "`_`".

7.1.3 .GEO file structure

The Gmsh .geo file structure supported by *InductEx* starts with optional variables, such as:

```
_var1 = 0.02e-3;
_var2 = _var1*2;
```

Points are declared next, with x, y and z coordinates (which can be parameterized in terms of the variables, and may use functions such as Sin and Cos supported by both *InductEx* and Gmsh) and a characteristic length. The characteristic length determines the size of mesh elements. Points must be numbered strictly sequentially, starting from 1. Examples are:

```
Point(1) = {1e-3, 2e-3, 0, 1e-3};
Point(2) = {_var1*Cos(_var2*2), 1e-3, 0, _charlength1};
```

Lines are used to connect points. Lines must be numbered strictly sequentially, starting from 1. An example is:

```
Line(1) = {1, 2};
```

Lines are looped to create a closed loop for surface definition. Both `Line Loop` and `Curve Loop` are accepted. Loops must be numbered strictly sequentially, starting from 1. Examples are:

```
Line Loop(1) = {-81, 1, 82, -21};
Curve Loop(2) = {5, 7, 9, -11};
```

Surfaces must be numbered sequentially starting from 1, and are defined from Line Loops with:

```
Plane Surface(1) = {125};
```

Surfaces are grouped together to enclose a volume with Surface Loop. Surface loops must be numbered sequentially, starting at 1:

```
Surface Loop(1) = {10, 12, 13, 14, 15, 16};
```

Finally, a volume is constructed from Surface Loops with:

```
Volume(1) = {44};
```

Volumes are numbered sequentially, starting from 1.

Physical volumes are composed of one or more volumes and are assigned with the Physical Volume line. *InductEx* only processes physical volumes and their constituent components as structures. Each physical volume must have a name that only has to be unique in that .geo file. Multiple physical volumes can be declared.

The format is:

```
Physical Volume("name") = {n};
```

The material properties of a physical volume are not defined in the .geo file, but in the .ixi file from which it is called.

7.1.4 Importing the .GEO file

Any number of .geo file structures can be imported into an *InductEx* layout through the `$IMPORT` record in an IXI layout input file, as discussed in Section 4.1.3.

7.2 Package modelling

InductEx allows the modelling of packaging for integrated circuits for the purpose of parameter extraction.

Currently, wedge-type and ball-and-stitch wire bonds are supported as geometric models, so that an integrated circuit can be modelled complete with wire bonding to a printed circuit board or chip carrier module. Capacitance extraction is not supported yet when wire bonds are present.

An example of such a chip and carrier package is shown in Fig. 7.1.

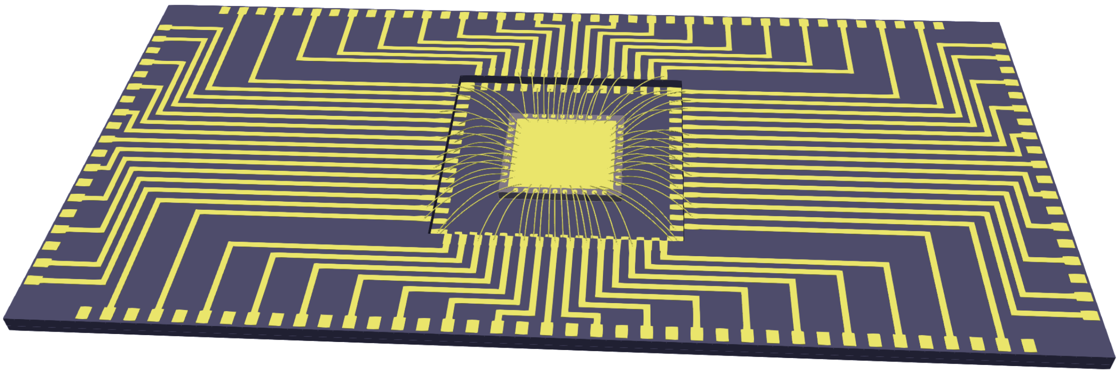


Figure 7.1: Example of a superconductor integrated circuit die mounted on a ceramic chip carrier and wire bonded to traces on multiple levels of the chip carrier.

Package modelling requires tetrahedral segments for the bonds, although chip layers can use triangles.

Package modelling requires setup in an IXI file, with every chip die or carrier printed circuit board imported from a GDS or IXI file with the `Die` record in an IXI file. Each chip of printed circuit board can have a different layer definition file to describe its process.

Bond structures are defined in the main IXI file. Bonds are connected between bond locations defined on die or printed circuit board layouts.

Warning: If the top metal layer is modelled with sheet currents (triangular mesh), a bond foot should not be surrounded entirely by a via directly below the top layer (thus seat inside a via), or the mesh will fail to connect the bond wire to the chip.

7.2.1 Wedge-type wire bonds

A wedge-type wire bond is shown in Fig. 7.2. The wire bond is modelled as a parabola, with the vertex placed at `VertexHeight` above the highest bond pad. Wedge-type wire bonds can be instantiated between any two bond locations on any dies, although the die surfaces must have the same normal. Currently, dies are required to lie flat (a normal of (0, 0, 1)) The die surfaces can have different z positions.

The wedge foot geometry is controlled by the length, height and width of the foot, the diameter of the wire, the length of the chamfer on the foot, the length and height of the toe protruding from the wedge foot, and the throat indentation where the wire meets the foot. The dimensions are detailed in Fig. 7.4 and Fig. 7.3.

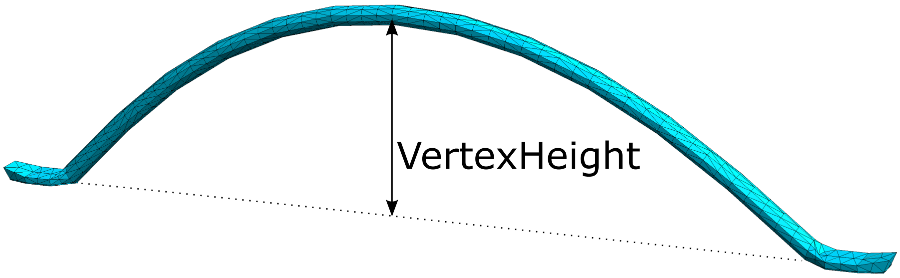


Figure 7.2: Wedge-type wire bond.

The wire bond model has 10 sections around the circumference of the wire (fixed), but the sections along the length of the wire can be selected with `LengthSections` to control the smoothness of the parabolic arc.

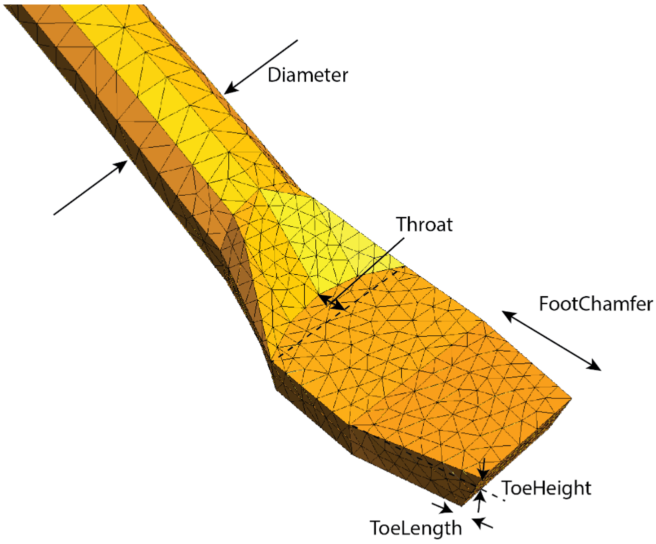


Figure 7.3: Wedge wire bond model top view showing wedge foot detail.

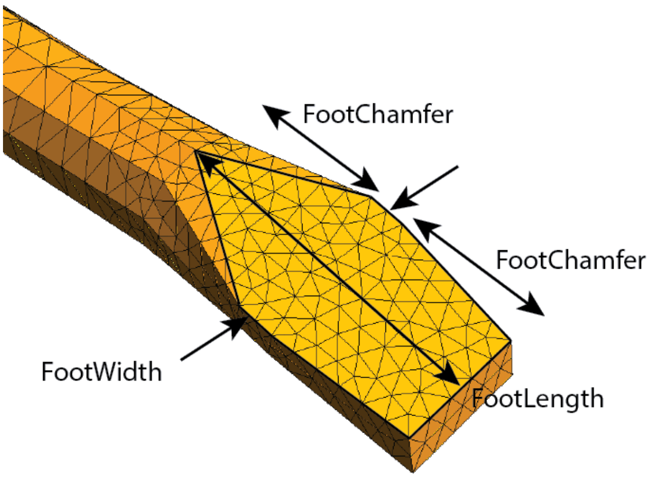


Figure 7.4: Wedge wire bond model bottom view showing contact surface of wedge foot.

7.2.2 Ball-and-stitch wire bonds

A ball-and-stitch wire bond is shown in Fig. 7.5. The start foot must be of type `Ball` and the end foot must be of type `Wedge`. The stitch uses the same model shown in Figs. 7.4 and 7.3. The ball parameters are shown in Fig. 7.5. Currently, dies are required to lie flat (a normal of (0, 0, 1)) The die surfaces can have different z positions.

The `VertexHeight` parameter sets the height of the wire above the top of the ball. *InductEx* will fail to create a bond model of the end foot is higher (has a larger z value) than the height of the wire set by the start foot, ball height and `VertexHeight`.

The wire bond model has 10 sections around the circumference of the wire (fixed), but the number of segments along the two bend arcs of the wire can be selected with `LengthSections` to control the smoothness of the bend arcs.

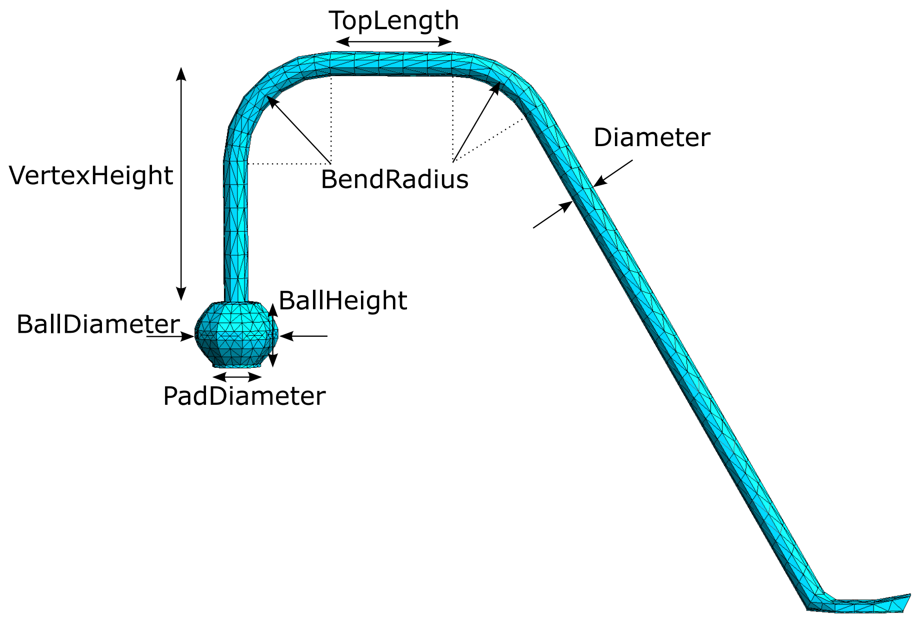


Figure 7.5: Ball-and-stitch wire bond model.

A ring with 5% of the ball height is slipped underneath the ball to weld the ball to the surface of a chip which may consist of multiple surface polygons and holes.

7.2.3 Bump bonds

InductEx can model solder bump bonds as shown in Fig. 7.6, where the effects of the `PadDiameter` and `Diameter` parameters are shown. The bumps have the same diameter at the upper and lower connection. If `Diameter` is smaller than `PadDiameter`, the bump is concave.

The smoothness of a bump can be controlled by specifying the number of segments around the circumference with `CircumferenceSections`. For convex or concave bumps, the smoothness along the bump height can be controlled with `LengthSections`.

The bump height is determined by the distance between the die surfaces between which a bump is defined. A ring with 5% of the bump height is slipped between each pad of the bump and the surfaces of the chips to which the bump connects to weld the bump surfaces to chip surfaces which may consist of multiple polygons and holes.

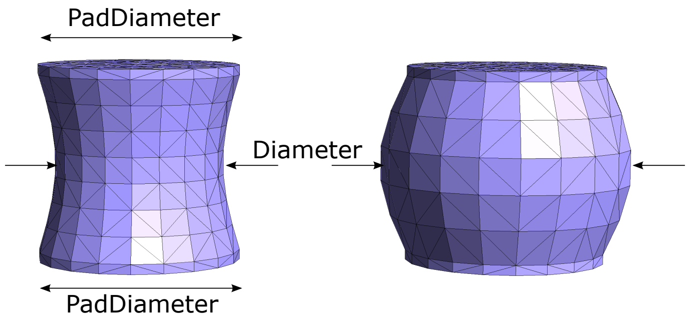
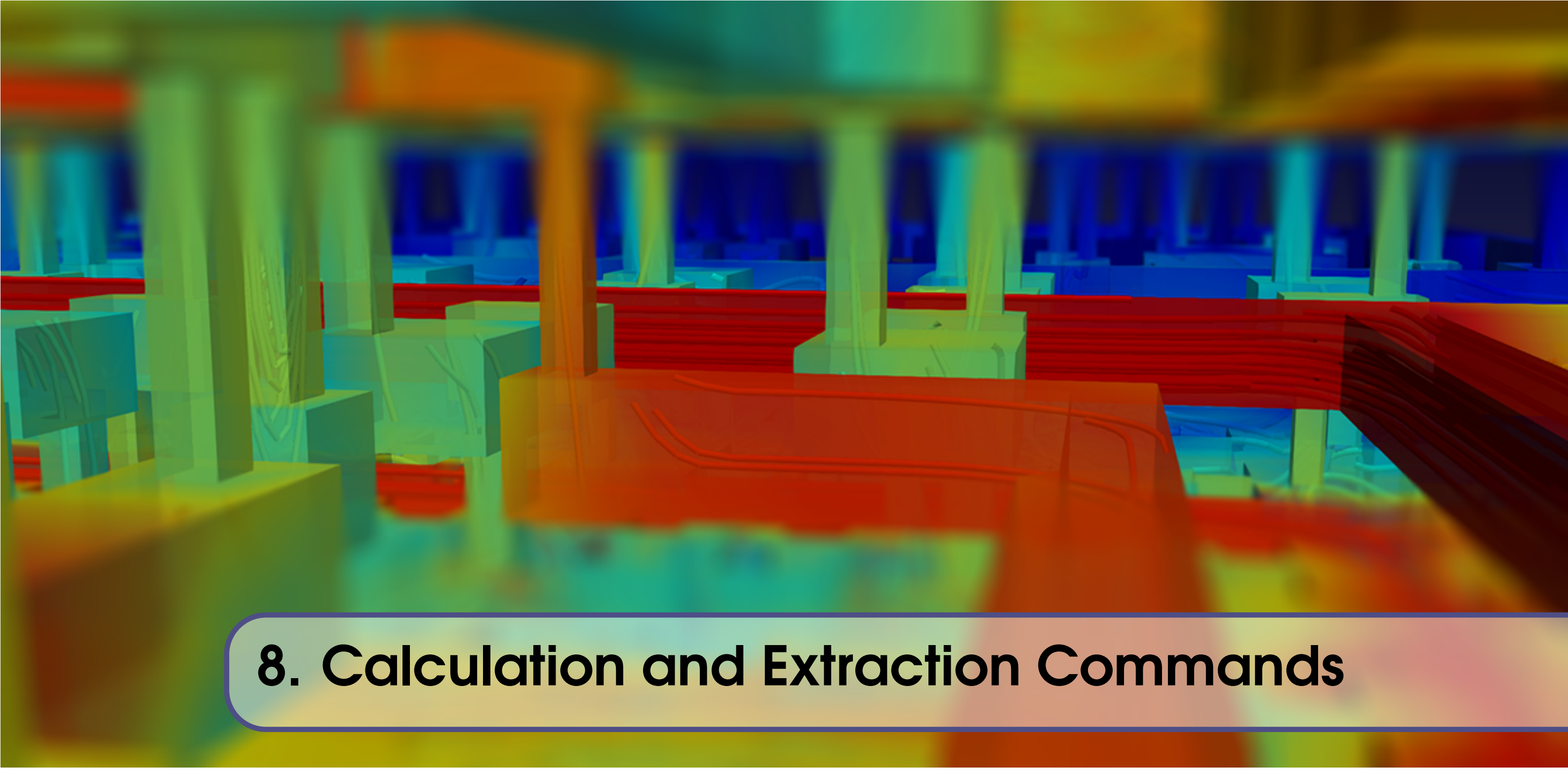


Figure 7.6: Concave and convex solder bump models. Both bumps have 20 sections around the circumference, while the concave bump has 7 length sections and the convex bump has 5 length sections.



8. Calculation and Extraction Commands

All calculation and model extraction options require a layout input file (IXI or GDS) and a process definition file (LDF) to describe the model geometry. The specific calculation mode is then set with appropriate command line switches (which may be specified in the IXI input file). Some calculation modes require other input files, as detailed in this section.

8.1 Inductance

Inductance is calculated by default for any number of inductances when one of the numerical engines is specified and a circuit netlist file is provided. The netlist file is required to describe inductors, the connection of inductors to ports and other inductors, and to list all mutual inductances. The engine mesh switches are `-fh` for cuboid filaments and `-th` for the hybrid tetra/triangle meshes.

The default mesh model file name can be overrided with the `-i` switch.

Typical execution commands are:

```
inductex layout.ixi -l process.ldf -n netlist.cir -i model.geo -th
inductex layout.gds -l process.ldf -n netlist.cir -i model.inp -fh
```

8.2 Resistance

Resistance is calculated for any number of resistive components in a calculation model which includes non-superconducting materials. *InductEx* is called exactly as for an inductance solution, and results will include inductance and resistance for each respective component.

If no resistors are specified in the extraction netlist, *InductEx* will add a series resistor to each inductor when resistive segments are detected in mesh.

8.3 Capacitance

Capacitance is calculated in the Maxwell capacitance matrix format for any number of conductors when selected with the `-qt` switch. No netlist is used.

A typical execution command is:

```
inductex layout.ixi -l process.ldf -qt
```

8.4 Characteristic impedance

Characteristic impedance can only be calculated for a single inductive component with a separate ground/return path (each of which may consist of multiple conductive structures) when the `-Z` switch is used. The layout structure must be labelled with two terminals, strictly named `Z1` and `Z2`. A netlist file is not used.

A typical execution command is:

```
inductex layout.ixi -l process.ldf -Z
```

8.5 S-parameters

S-parameters are calculated for a two-port network when the `-S` switch is used. The layout structure must be labelled with at least two terminals, of which the first two are used. A netlist file is not used.

Note: in the current version of *InductEx* , S-parameter calculation is only available for thin-film structures modelled with sheet currents, so that only on-chip structures modelled with triangular meshes are supported. S-parameter analysis thus forces all layers to use sheet currents, overriding any process file settings or parameter string switches.

The frequency range, number of calculation steps and the source and load impedances are specified in the command line parameter string as six optional parameters after `-S`: *FrequencyScale* (LIN or DEC), *StartFrequency* (real), *StopFrequency* (real), *NumberOfFrequencyPoints* (integer), *SourceImpedance* (real) and *LoadImpedance* (real).

The default values are: *FrequencyScale* = DEC, *StartFrequency* = 1×10^9 Hz, *StopFrequency* = 10×10^9 Hz, *NumberOfFrequencyPoints* = 10, *SourceImpedance* = 50 Ω and *LoadImpedance* = 50 Ω.

Typical execution commands are:

```
inductex filter.gds -l process.ldf -S
inductex filter.gds -l process.ldf -S LIN 1e9 10e9 100 50 50
```

8.6 Compact SPICE model

All compact SPICE models are fully compatible with JoSIM.

8.6.1 Back-annotation of results

Compact SPICE model extraction requires back-annotation of calculated parameters to a SPICE-based simulation deck. No switches need to be set. If the `BACK-ANNOTATE` command is included in the extraction netlist, as detailed in Section 5.5.1, and a target simlation deck exists, *InductEx* will write a compact SPICE model that includes all extracted parasitic elements.

8.6.2 Coupling from external magnetic field

Coupling from an external field is calculated for an inductive netlist extraction when the `-b` switch is used. This extraction requires at least a layout input file, a layer definition file and an extraction netlist.

A typical example is:

```
inductex layout.gds -l process.ldf -n netlist.cir -i model.geo -th -b
```

8.6.3 Coupling from moats

Coupling from fluxons trapped in moats is calculated for an inductive netlist extraction when any `"F"` label is included on the layout. The required files and execution commands are the same as for inductance calculation.

8.7 Critical field analysis

A critical field analysis is done for every moat labelled for extraction when the `-B` switch is used. This extraction requires a layout input file and layer definition file, but not an extraction netlist (holes are marked on the layout).

A typical example is:

```
inductex layout.gds -l process.ldf -i model.geo -th -B
```

8.8 Visualization

Several calculated quantities can be written to visualization files. *InductEx* uses the .VTK format for ParaView.

8.8.1 Current density

Current density visualization in all conductors is activated with the `-j` switch during inductance calculation.

A typical execution command is:

```
inductex layout.ixi -l process.ldf -n netlist.cir -i model.geo -th -j
```

8.8.2 Electric field strength

Electric field strength visualization over a defined volume or one or more planes is activated with the `-E` switch during capacitance calculation.

A typical execution command is:

```
inductex layout.ixi -l process.ldf -qt -E
```

8.8.3 Magnetic field intensity

Magnetic field intensity visualization over a defined volume or one or more planes is activated with the `-M` switch during inductance calculation.

A typical execution command is:

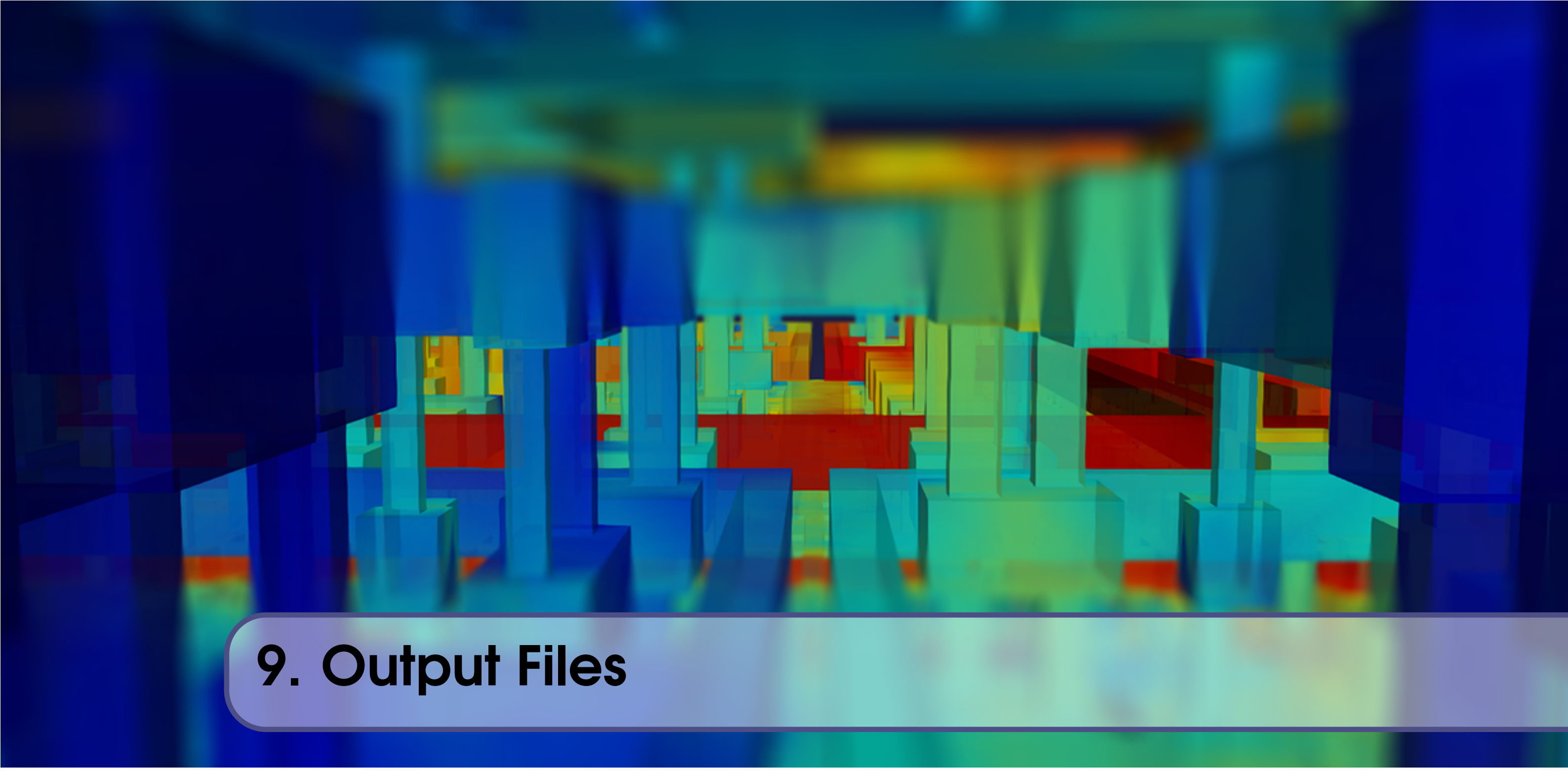
```
inductex layout.ixi -l process.ldf -n netlist.cir -i model.geo -th -M
```

8.8.4 Magnetic flux density

Magnetic flux density visualization over a defined volume or one or more planes is activated with the `-MB` switch during inductance calculation.

A typical execution command is:

```
inductex layout.ixi -l process.ldf -n netlist.cir -i model.geo -th -MB
```



9. Output Files

InductEx creates several output files. The format of some of these files is presented here to aid the interpretation of results. File names can be prepended with the string defined with the `-w` command line parameter.

9.1 Solution output

The simulation result is always printed to the standard output as well as written to the file `sol.txt`.

9.1.1 Solution result

The simulation output file lists the results for a component such as impedance (each entry of which represents a single branch inductance and resistance) in a block that contains two head lines and the component entries in the following format:

`Inductance` [*unit*]
`Name` `Design` `Extracted` `AbsDiff` `PercDiff`
Elementname *DesignVal* *ExtractedVal* *Absdiff* *Percdiff*

InductEx lists the followings:

Elementname
: is the name of the netlist element.

DesignVal
: is the inductance design value if inductance is listed in the netlist. If no value is provided in the netlist, "`--`" is printed.

ExtractedVal
: is the extracted inductance value in the units specified in the header line. If the extracted value is infinite (an open circuit), "`--`" is printed.

Absdiff
: is the absolute difference between the extracted and design values of the element. If no design value is provided in the netlist, *Absdiff* is equal to the extracted value of the element.

Percdiff

is the percentage difference between the extracted and design values of the element. If no design value is provided in the netlist, "--%" is printed.

Outputs for resistance, mutual inductance and Josephson junction area have similar structure.

9.1.2 Mesh quality

The cuboid mesh quality when filament meshing is used is listed as a set of values in the line:

```
Mesh info: dx = Mindx - Maxdx; dy = Mindy - Maxdy;
   Worst L/W ratios: min = Minlw; max = Maxlw
```

where:

Mindx

is the minimum dimension of any segment in the x direction, specified in units.

Maxdx

is the maximum dimension of any segment in the x direction, specified in units.

Mindy

is the minimum dimension of any segment in the y direction, specified in units.

Maxdy

is the maximum dimension of any segment in the y direction, specified in units.

Minlw

is the minimum length-to-width ratio of any mesh segment.

Maxlw

is the maximum length-to-width ratio of any mesh segment.

A high quality mesh is one for which the segment length-to-width ratios stay close to unity. If the mesh is of poor quality, the accuracy of current distribution calculations and subsequently the extracted element values degrades, and stability of the solver can be compromised.

If the length-to-width ratio falls outside the range 0.1 to 10, strongly consider smaller values for the `SegmentSize` parameter or careful grid snapping of layout object vertices.

9.1.3 Quality of the singular value decomposition for inductive networks

InductEx calculates the extracted values for inductance, resistance and mutual inductance in inductive networks with the singular value decomposition technique. Feedback on the quality of the solution is provided as:

```
SVD solution information (A*x = b):
 - Unknowns = unknowns; rank = rank.
 - Condition number:                K
 - Max error (Max(|A*x - b|)):      maxerror
 - Relative error (||A*x - b||/||b||): relerror
```

The quality indicators are:

K is the result of dividing the largest singular value by the smallest one.

unknowns

is the number of unknown element values to be solved, which is the sum of all inductance and resistance components and all mutual inductances.

rank

is rank of the branch current matrix.

maxerror

is the worst error in any port voltage calculated with the extracted element values and the reference currents found with *TetraHenry*. For standard inductance calculation, unless the amplitude of any voltage source has been specified in the netlist file, the maximum value of any element in **b** is 1, and the maximum error should be only a small fraction of 1.

relerror

is the norm of the error vector (extracted element values multiplied by the branch current matrix) divided by the norm of the loop voltage vector **b**, expressed as a percentage.

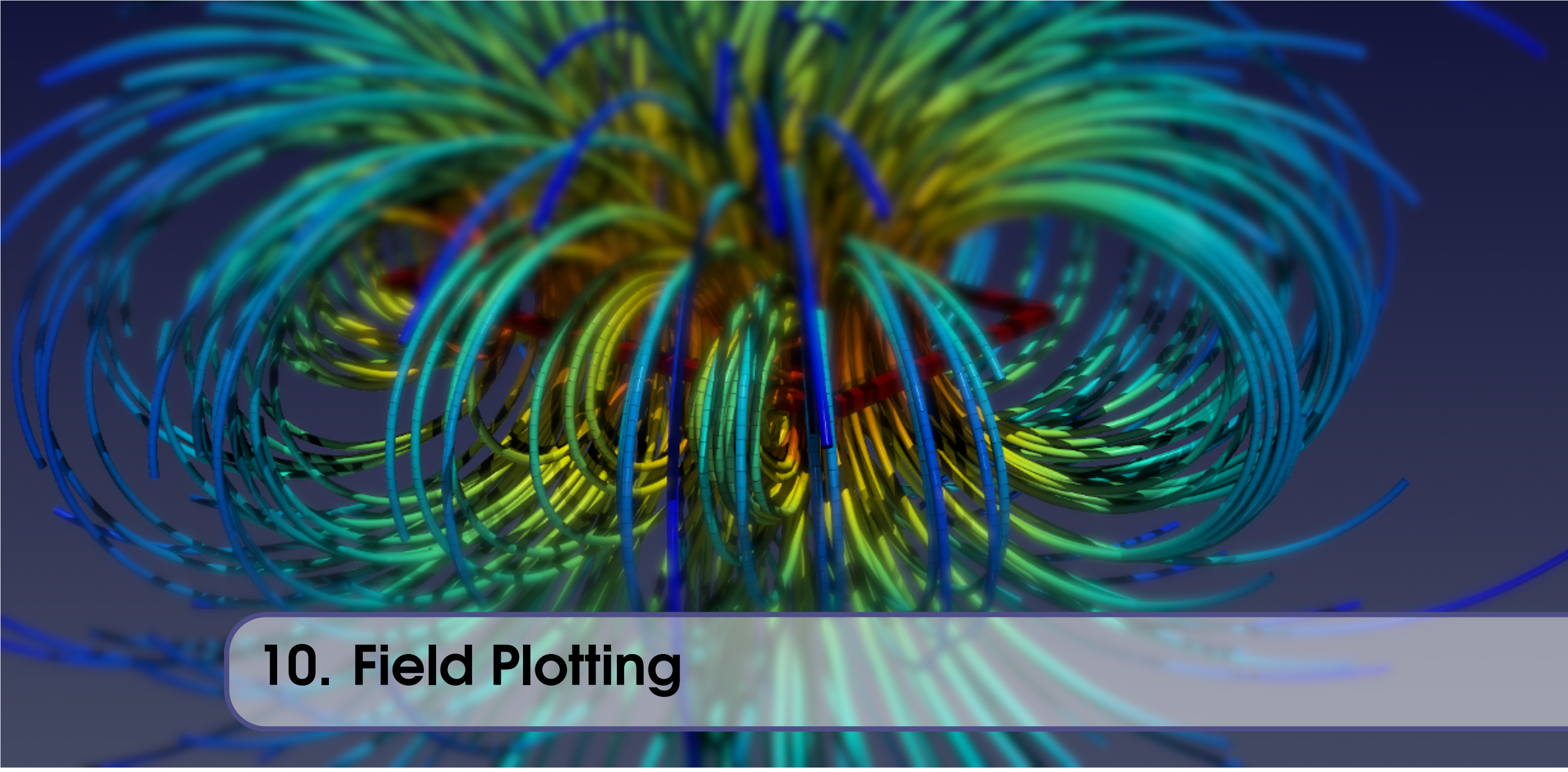
For solutions with high precision, K should be close to unity. The precision of all solutions degrades by one significant digit for every order of magnitude in the value of K. **If *K* exceeds 1000**, consider that the circuit model (netlist) may be wrong. **If *K* approaches 1E6 or above, treat all results as suspect.**

If *rank* is lower than *unknowns*, treat all results as suspect and consider altering the circuit netlist, unless fluxon or external magnetic field coupling is enabled.

Maxerror does not provide any information on the error or loss of accuracy in any given extracted element value. It only shows the largest error in the voltage at any port between what was calculated with the numerical engine (and thus used in the singular value decomposition) and that which is found from multiplying branch currents with the extracted element values. *Maxerror* should be much smaller than 1 (below 0.05, typically) for purely inductive networks that have no mutual inductances to indicate a high quality solution. In some voltage loops with much lower total impedance than other loops, *Maxerror* can be inflated.

The relative error *relerror* is more useful. *Relerror*, expressed as a percentage, should be at most 5%. If it exceeds about 10%, then it is highly likely that parasitic coupling or some strong mutual inductance is present in the layout that was not accounted for in the netlist. There is no clear limit for *relerror* above which results cannot be trusted, but for circuits without significant mutual inductance it is strongly advised that all results be treated as suspect if *relerror* exceeds about 15 % to 20 % (the higher value for more complex circuits with larger distributions between largest and smallest inductance). The error can mostly be reduced significantly by the inclusion of mutual inductance elements in the circuit netlist, or it might indicate a bad agreement between the netlist and the layout.

For a good match between netlist and layout, *relerror* should be comfortably below 1%.



10. Field Plotting

InductEx allows the plotting of magnetic and electric fields when volumes or planes are specified.

A field plot can be generated when *InductEx* is executed in inductance extraction or capacitance extraction mode with the *TTH* engine.

In inductance calculation mode, the `-M` command line switch enables a magnetic field intensity plot in A/m, while the `-MB` switch enables a magnetic flux density plot in Tesla. Both plots can be requested simultaneously.

In capacitance calculation mode, the `-E` command line switch enables an electric field strength plot in V/m.

10.1 Volumes

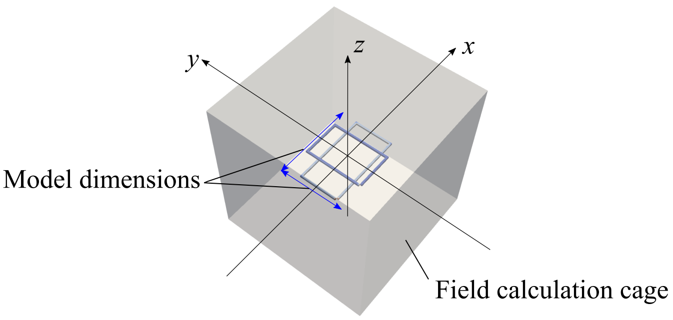


Figure 10.1: Field calculation volume around an extraction model.

The default field plot is generated over a cuboid volume that surrounds the extraction model, as shown in Fig. 10.1. The volume is centered around the centre of the extraction model, with side lengths in the x and y axes equal to the largest dimension of the extracion model multiplied by the scaling factor `FieldCageScale`. The cuboid volume's height in the z direction is derived from the side length in the xy plane multiplied by the vertical scaling factor `FieldCageZScale`.

The cuboid volume is meshed and field values are calculated at every node. The mesh size is specified with `FieldMeshSize`, which has default value of 2 Units.

Electric field results are written to `E_field.vtk`, magnetic field intensity results to `H_field.vtk` and magnetic flux density results to `B_field.vtk`.

10.2 Planes

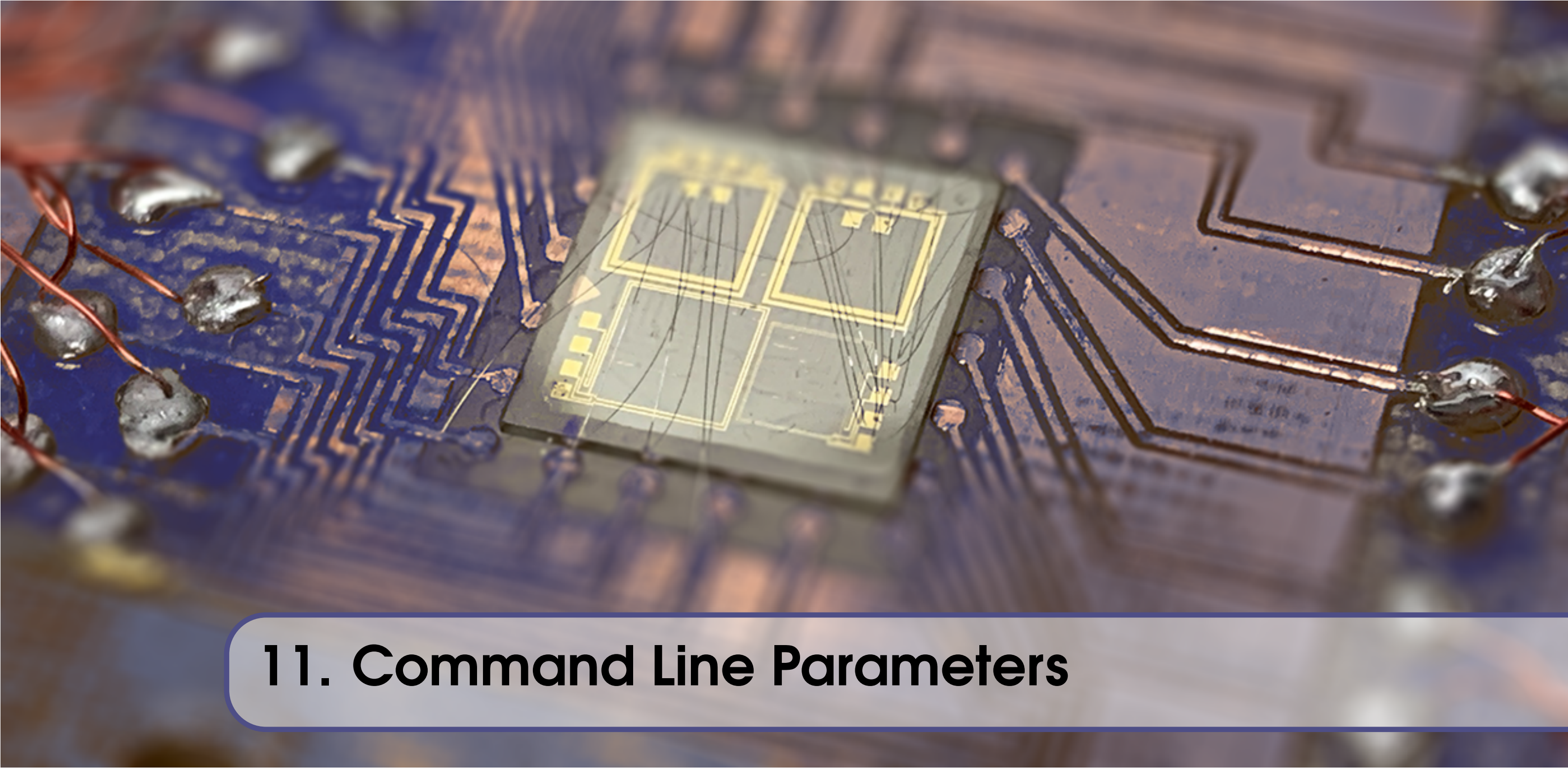
Field calculation over a volume is computationally expensive. For quick analysis or symmetrical structures it is often sufficient to view the field in a plane that slices through the space that contains an extraction model.

A plane can only be defined in the IXI input file as an FPlane record (described in Section 4.1.3). A plane is defined as a rectangle between two opposing corners in the *xy* plane, after which it is translated and rotated as specified.

Multiple planes can defined for one extraction model. Fields are then calculated over each of the planes. The mesh size for all planes is specified with `FieldMeshSize`.

When any plane is defined, it overrides the field volume calculation.

The electric field results for the first plane are written to `E_field_plane_0.vtk`, and the counter in the name is increased by one for each additional plane. The same counting method is applied to magnetic field intensity and magnetic flux density results, with the results for the first plane of each calculation type written to `H_field_plane_0.vtk` and `B_field_plane_0.vtk` respectively.



11. Command Line Parameters

11.1 *InductEx* command line paramters

InductEx uses command line parameters and switches to enable CAD tool integration.

The first command line parameter must be the layout input file, which can be Calma GDSII or IXI format.

The other parameters or switches can be organized at random, and many are optional.

-a Adds edge segments with width equalt to the penetration depth to superconductor objects with tetrahadral and triangular meshes for high fidelity mutual inductance extraction, and sets GMRES AbsTol = Tol = 1×10^{-6}.

-b Add a magnetic field analysis for external fields.

-c *n* Specifies n as the GMRES iteration limit for *TetraHenry*. InductEx uses a default of 400 iterations.

-D Dumps the circuit netlist as a graph to `netlistgraph.gv`. GraphViz (https://graphviz.org) must be installed to compile the graph to `netlistgraph.pdf`.

-d *filename.gds*
Instructs *InductEx* to write the geometry input to a GDS file after flattening all cell hierarchies and applying mask-wafer bias adjustments. If the optional output file name is omitted, InductEx will add the prefix `inductex` to name of the geometry input file and use this as the output file name. This switch is useful to verify what *InductEx* thinks it read, and to view mask-wafer bias effects. *InductEx* also writes a file with slicing and node number information for debugging purposes.

-E Enables electric field plot generation when capacitance is calculated with *TetraHenry*. The field plot mesh element size and field cage size are controlled with the parameters `FieldMeshSize` and `FieldCageSize` in the layer definition file.

-e Equalises grid slicing, irrespective of positions of individual vertices, when filament meshing is used. This option is only useful for very large structures with non-Manhattan polygon structures (such as spiral inductors) that create intractably large segment counts (such as millions of segments). The SegmentSize value for each layer is then used to set segment size. Because there is no alignment of segments above vias, ZSegsToEC is overrided to TRUE.

-eg ***x***

Equalises grid slicing and snaps all edges to a grid defined by the floating point value *x* when filament meshing is used. This option is and extension of the equal segment slicing option -e, but grid-locked slicing allows vertical segments. The global SegmentSize value is used as default value for *x*.

-fh Selects inductance extraction with filament meshing.

-freq ***x***

Overrides frequency at which impedances are calculated, where the floating point value *x* is the new frequency.

-h Prints help blurb to screen.

-i ***filename.inp***

Specifies the name of either the meshed cuboid filament file generated by *InductEx* (if the .inp suffix is used, as in `filename.inp`), or the geometry file for the third-party meshing software Gmsh if the .geo suffix is used (e.g. `filename.geo`). If this switch is omitted, *InductEx* will use the value assigned to the environment variable IXINP. Failing that, the default name ix.inp or ix.geo is used.

-id Lets *InductEx* write out the system identifier information to `IX_sysID.txt`.

-j Instructs *InductEx* to process the current density results for all ports and build a current density visualization file in VTK format.

-k Disables file cleanup after *InductEx* execution. This leaves the current density files for post-processing.

-L `path\licensefilename.txt` – specifies the full path to the licence file name for the MS Windows version of *InductEx*. This overrides all other licence file identifiers. Spaces are allowed in folder and file names, provided that the string is enclosed in double quotation marks ("...").

-l ***filename.ldf***

Specifies the layer definition file from which process parameters are read for the main layout structure. This switch can only be omitted if the layout input is read from an IXI file without a `$STRUCT` object.

-leg Instructs *InductEx* to use legacy IXI format (before version 6.0).

-M Enables magnetic field strength (**H** in A/m) plot generation when inductance is calculated. The field plot mesh element size and field cage size are controlled with the parameters `FieldMeshSize`, `FieldCageScale` and `FieldCageZScale` in the layer definition file.

-MB Enables magnetic flux density (**B** in Tesla) plot generation when inductance is calculated. The field plot mesh element size and field cage size are controlled with the parameters `FieldMeshSize`, `FieldCageScale` and `FieldCageZScale` in the layer definition file.

-n ***filename.cir***

Specifies the name of the Spice netlist file used by *InductEx* to find the inductance network. If this switch is omitted, InductEx will try to read the netlist from the file with the same name as the geometry input file (in GDSII or other format). For example, if the input file is `jtl.gds`, then *InductEx* will search for `jtl.cir` and `jtl.js` (in that order) to read the netlist from.

-noshadows

Disables shadow casting for tetra/triangle meshes, leading to faster but less accurate solutions.

-nosvd

Disables SVD solution, so that the user must use `matrix_I.txt` and `matrix_v.txt` to solve inductance.

-O Activates mesh optimization (remeshing) for triangular or tetrahedral mesh models.

-p *filename.txt*
Specifies the name of the port label text file used by *InductEx* to override port labels in the layout. The ASCII text file may have any extension, but `.txt` is preferable.

-pdump
Writes a port label text file for every layout input.

-qt Selects capacitance extraction. This overrides inductance calculation. The option does not require a circuit netlist file.

-S *startfrequency stopfrequency numberofsteps sourceimpedance loadimpedance*
Enables S-parameter extraction. This overrides inductance calculation.

-scube
Sets sparsity pattern of preconditioner in TTH to cube.

-t *filename*
Specifies the name of the text-based terminal definition file that declares ports/terminals in Lmeter-compatible format. If the filename or the switch is omitted, *InductEx* defaults to `lmeter.term` for compatibility with Cadence-based Lmeter scripts.

-th Selects inductance extraction with tetrahedral and/or triangular meshing. This overrides the -fh switch. If the switch is omitted, only the `.geo` geometry file is created, and the output from the last execution of *TetraHenr*y is used to solve the impedance network. If a change to the circuit netlist is made without any change to the layout, leaving out the -th switch lets *InductEx* solve the network without recalculating current distribution.

-top *name*
Specifies the top structure name in the layout input file. The default is the structure defined last in the input file.

-u Enables energy density visualization file in VTK format.

-v *n* Verbose mode on, with level *n*. With this switch set, *InductEx* prints more information to the screen and the output file `sol.txt`.

-w *name*
Writes all native output files with `name_` as prefix (yielding `name_sol.txt`, `name_ix.cur`, `name_fastout.out` and `name_a.txt`). This allows *InductEx* to execute without overwriting previous output files, and enables multiple instances of the programme to execute in the same working directory.

-X Enables text label propagation from lower to higher order structures in layout. The default behaviour, without this switch, is for *InductEx* to read only text labels from the top level structure.

-x *n* Specifies *n* as the number of processor cores to use with the GMRES solver in *TetraHenry*. The default is the system maximum.

-Z Selects impedance extraction. This overrides inductance calculation and the -fh switch, as well as capacitance calculation and the qt switch. The option does not require a circuit netlist file, as it assumes that a layout is defined with two ports: Z1 and Z2.

-z When tetrahedral meshing is used, reduces all conductive objects to sheets for current modelling (overrides layer definition file and zoff switch).

-zoff When tetrahedral meshing is used, forces all conductive objects to be modelled as volumes (overrides layer definition file).

11.2 Extra command line parameters to *TTH* and Gmsh

For advanced users it is sometimes necessary to specify parameters fot the mesher Gmsh or the calculation engine *TetraHenry*. *InductEx* can be instructed to append parameters to the string that it passes to each of these tools.

To do so, an ASCII text file that contains all parameters as a single line string per tool is fed to *InductEx* with the `-y` switch.

The file syntax is:

```
$\textit{toolname string}
```

The supported tool names are `TTH` and `GMSH`.

Tool names are case insensitive, but the parameter strings are case sensitive.
Examples are:

```
$gmsh -v 5
```

12. Exit Codes

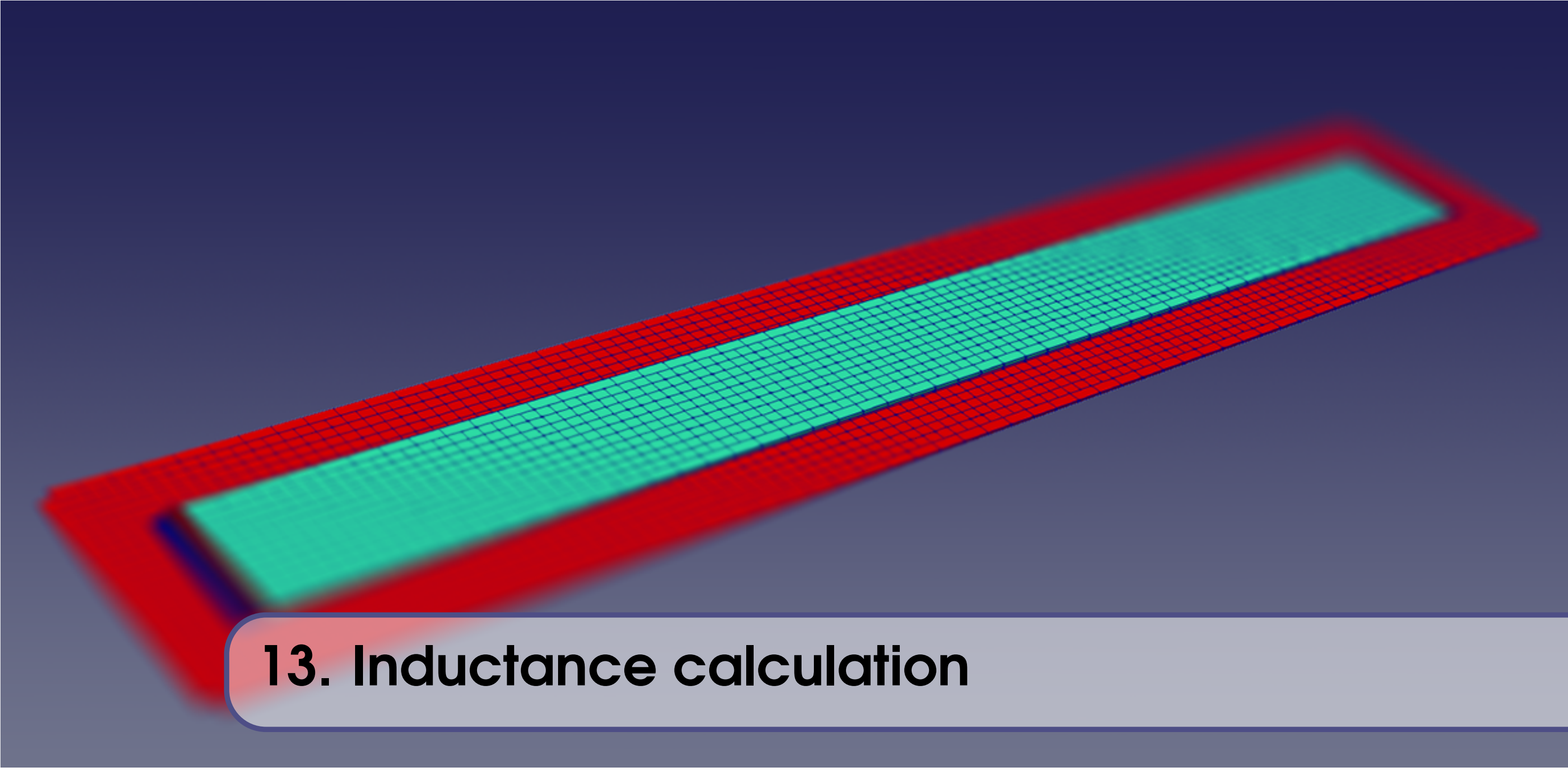
Exit codes are printed in the exit message to the standard output. The comprehensive list, with some solutions where applicable, is:

1 Error while trying to open layout input file (GDS or IXI). The file does not exist, or cannot be read.
2 Error while trying to open port label text file. The file does not exist, or cannot be read.
4 Error while trying to open process (layer definition, LDF) file, or process file not defined.
5 Error while trying to open netlist file.
6 Error while trying to open *FFH/TTH* port current or capacitance result file. Engine did not execute correctly (examine `ffh.log` or `tth.log`) or directory has write restrictions.
7 Error while trying to open mesh file.
8 Error reading solver output file `solver_out.txt`. Check if solver is in the path.
9 The input file format is not supported anymore (CIF or DXF).
10 *InductEx* cannot create an output file. Check of working directory has write permission.
20 Cannot resolve all inductor (branch) currents when solving inductance; no solution possible.
30 No parameter block defined in LDF file, leaving modelling parameters undefined. Check if parameter block starts with `$Parameters`.
31 Invalid input format or syntax in LDF file. `$End` statement found without an appropriate opening statement.
32 Duplicate layer number in LDF file, or no layer number defined in layer block. Layer number ambiguity could cause modelling errors and must be corrected.
33 Process uses a ground plane, but ground plane layer not identified. Add the parameter `gplayer=`*n* to the LDF file, where *n* is the ground plane layer's GDS number.
34 Capacitance conductor layer definition invalid, or does not support conductors (should be FilmType `S` or `R`).
35 Duplicate layer name in LDF file, or no layer name defined in layer block. Layer name ambiguity could cause modelling errors and must be corrected.
40 Component (such as inductor or port) name duplicated in circuit netlist file. Remove ambiguity by renaming one of the components (case-insensitive).

41 Error in port definition in netlist, such as a missing negative terminal.
42 Cannot link positive or negative terminal of a port to a layout figure. Check layer names in port labels or in LDF file.
43 Port positive or negative terminal layer definition invalid, or does not support terminals (should be FilmType `S` or `R`).
44 Cannot identify port surface while building Gmsh geometry file.
45 Component in parallel with another of its kind (inductor-inductor or port-port). This cannot be resolved for extraction.
46 No cycles in netlist, or bad cycle found. Check for series inductors in a branch, or a branch with a port but no inductance.
47 There is a mismatch between the number of ports (including magnetic field sources) in the netlist, layout and current output file (`ix.cur` or `ixth.cur`) when inductance is solved.
48 A port misses a terminal (in cuboid mesh).
49 Number of port/terminal text labels do not correspond to number of ports/terminals found in layout.
50 Layout object (e.g. path, poly, sref) outside of Struct record in IXI file. Structure cannot be resolved.
51 Coordinates not specified for a layout object in IXI file.
52 Non-unity SREF magnification in layout input file not supported by *InductEx*. Model geometry will not match layout.
53 Collapsed polygon in layout.
54 No objects in layout, either because layout file is empty or objects are all deleted by cut layers or operators, or were declared on layers not defined in the LDF file. A 3D extraction model cannot be built.
55 Duplicate terminal labels in layout at different locations - ambiguity cannot be resolved.
56 Invalid record length in GDS file - read-in aborted.
57 Structure name in GDS file exceeds 127 characters.
58 Port terminal defined on layer with no objects.
59 Record or command not recognised in IXI file.
60 No structures read from input an IXI layout input file. This mostly happens when an IXI file from before v6.00 is used, and no STRUCT records are defined.
61 Virtual cut terminal cannot slice conductor object into two separate polygons (it either does stretch edge-to-edge, or cuts conductor polygon multiple times).
62 Layout cannot be identified in multiple-layout IXI input or for linking bonds.
70 Abnormal termination or failed execution of external module (such as *TTH* or Gmsh).
82 Incompatible format in Solver output file `solver_out.txt`. Use the solver version packaged with this version of *InductEx*.
90 Different capacitance conductors linked to the same conductive structure.
91 No capacitance conductors identified.
110 Licence flavour does not support requested functionality (as listed in the exit message).
111 Licence error.
113 Invalid licence.
120 Mesh file not found. Gmsh did not complete mesh. Cannot execute *TTH* without a mesh.
190 No valid edge found in fundamental circuit construction. Check netlist.
191 Positive and negative node the same for an edge (component) in fundamental circuit construction. Check netlist.
192 Missing node declaration for an edge (component) in fundamental circuit construction. Check netlist.
200 Component lookup returned invalid dynamic array index - will result in memory error.

210 Syntax error in expression string (such as missing parenthesis).
211 Cannot parse variable in expression string. Variable identification symbol might be absent, or variable not defined.
220 No I_C read for Josephson junction in JoSIM netlist.
230 No filament info found in *FFH* mesh file - check `ffh.log`.
231 Exception – positive or negative terminal of a port has no nodes. This is an *InductEx* node-building error for cuboid meshes and should be reported if encountered.
240 Construct construct type not defined in LDF file.

II Calculation Examples



13. Inductance calculation

13.1 Microstrip

Example works with all licence tiers

This basic example, available in `examples/inductance/microstrip/`, demonstrates:

- Negative mask ground plane.
- Cuboid filaments.
- Edge terminals.
- GDSII layout input file.
- Inductance in Henry.

Consider the calculation of the inductance of a single, straight inductor above a ground plane (a microstrip line, which is easily solved analytically [12] but used here due to its simplicity). A cross-sectional view of such a microstrip line above a ground plane is shown in Fig. 13.1.

For this example, we use a microstrip line of length 100 µm and width (W) 10 µm. Line thickness (t_1) is 0.25 µm, ground plane thickness (t_2) is 0.2 µm and dielectric isolation is thickness (h) is 0.15 μm. The analytical solution, assuming an infinite ground plane, is 3.9 pH.

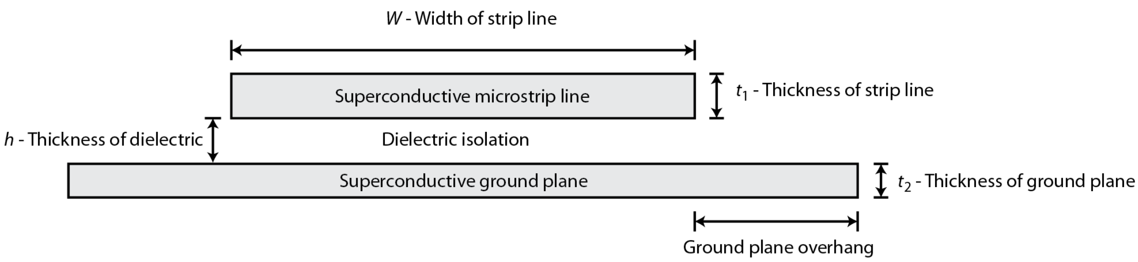


Figure 13.1: Cross-sectional view of superconductive microstrip line and ground plane.

Calculating the same inductance with *InductEx* requires us to draw a 100 µm × 10 µm line in a CAD tool capable of generating a GDS file.

Four layers are defined: M0 as the ground plane, I0 as the dielectric isolation layer, M1 as the conductor, and TERM as the terminal layer. The microstrip line is drawn as a box in M1, and the terminals are added as paths of width 2 µm at the furthest edges of the microstrip line. The width serves no purpose other than for visualisation, as the terminals are constructed at the path centre. Text labels are placed with their coordinates exactly on the edges of the microstrip line, inside the terminal paths, to identify the ports. The ground plane is not drawn, because the ground plane layer mask in this example is negative – it thus exists everywhere except where it is drawn. A sketch of the layout is shown in Fig. 13.2. The layout is then saved as a GDS file. For this example, we use `microstrip.gds`.

A simple circuit netlist with a single inductor and two ports describes the model. It is shown in Netlist 13.1.1.

Figure 13.2: Layout of microstrip in layer M1 with text labels identifying the terminal objects in layer TERM.

■ Netlist 13.1.1 — microstrip.cir.

```
* Title: Netlist for InductEx user manual - microstrip line
//--- Inductors ---
L1      1   2    3.9E-12
//----- Ports -----
P1      1   0
P2      2   0
.end
```

Finally, we need a layer definition file that defines layer dimensions and simulation variables. *InductEx* will accept any file extension, but we use `.ldf` by default. The layer definition file `process.ldf` is shown in Layer Definition File 15.1.2.

■ Layer Definition File 13.1.2 — process.ldf.

```
* --------------------------
* PARAMETERS
* --------------------------
$Parameters
  Units                  =  1e-6
  SegmentSize            =  2.0
  AbsMin                 =  0.025
  GPOverhang             =  5.0
  ProcessHasGroundPlane = TRUE
  GPLayer                =  17
  TermLayer              =  19
  TextLayer              =  18
  HFilaments             =  1
  TerminalInRange        =  1.0
  UnitL                  =  1
$End
* --------------------------
* LAYERS
* ----------- M0 ------------
$Layer
  Number     =    17
  Name       =    M0
  Thickness  =    0.2
  Lambda     =    0.09
  Order      =    0
  Mask       =   -1
  Filmtype   =    S
  HFilaments =    2
$End
* ----------- I0 ------------
$Layer
  Number     =    2
  Name       =    I0
  Thickness  =    0.15
  Order      =    1
  Mask       =   -1
  Filmtype   =    I
$End
* ----------- M1 ------------
$Layer
  Number     =    5
  Name       =    M1
  Thickness  =    0.25
  Lambda     =    0.09
  Order      =    2
  Mask       =    1
  Filmtype   =    S
  HFilaments =    3
$End
* ---------- TERM -----------
* TERM
$Layer
  Number     =    19
  Name       =    TERM
  Order      =    3
  Mask       =   -4
$End
```

The parameters are discussed in Section 3.2.3 and 3.2.4. Note that the ground plane uses a negative mask, and that the TERM layer uses a mask value of -4.

It is not strictly necessary to include the TERM layer definition block. The terminal layer is already specified by the TermLayer global parameter, but inclusion of the layer block allows LayoutEditor to recognise the terminal layer.

Since we cannot model an infinite ground plane, the ground plane overhang (`GPOverhang`) is specified as 5 µm. Given the dimensions of the problem, this is sufficient.

InductEx is executed from the command prompt for inductance calculation with a filament mesh with:

```
inductex microstrip.gds -l process.ldf -i microstrip.inp -fh
```

The screen output under Windows is shown below. Under Linux and OS X, *TetraHenry* is executed in the same terminal and the output is more verbose. The output is also written to `sol.txt`.

Command Prompt

```
InductEx v7.0 (16 September 2022). Copyright 2003-2022 Coenrad Fourie

Parameter string: microstrip.gds -l process.ldf -i microstrip.inp -fh
  Netlist input:
+================+===========+======+========+====================+
| Netlist file   | #Inductors | #M/k | #Ports | #Fundamental loops |
+----------------+------------+------+--------+--------------------+
| microstrip.cir | 1          | 0    | 2      | 1                  |
+================+===========+======+========+====================+
  Layout input files processed:
+===+================+=============+===========+========+========+=============+============+
| n | Layout file    | LDF         | Top-level  | #Polys | #Terms | TecUnit (m) | DBUnit (m) |
+---+----------------+-------------+------------+--------+--------+-------------+------------+
| 0 | microstrip.gds | process.ldf | microstrip | 2      | 2      | 1E-06m      | 1E-09m     |
+===+================+=============+===========+========+========+=============+============+
  Ports:
+====+======+======+===========+
| Nr | Name | Type | Amplitude |
+----+------+------+-----------+
| 1  | p1   | v    | 1         |
| 2  | p2   | v    | 1         |
+====+======+======+===========+
  Terminals:
+======+========+=======+===================+========+
| Name | Layout | Layer | Geometry          | Object |
+------+--------+-------+-------------------+--------+
| p1+  | 0      | m1    | Line/Edge Surface | 1      |
| p1-  | 0      | m0    | Line/Edge Surface | 0      |
| p2+  | 0      | m1    | Line/Edge Surface | 1      |
| p2-  | 0      | m0    | Line/Edge Surface | 0      |
+======+========+=======+===================+========+
  Mesh info:
+===========+==============+==============+==========================+
| #Filaments | X-width range | Y-width range | Worst L/W ratios         |
+------------+---------------+---------------+--------------------------+
| 1878       | 1.666 - 1.961 | 1.666 - 1.667 | min = 0.8496; max = 1.177 |
+===========+==============+==============+==========================+
  Time allocation (s):
+===============+===============+===============+======+=======+
| Process layout | Slice surfaces | Build 3D model | Mesh | TTH   |
+----------------+----------------+----------------+------+-------+
| 0              | 0.004          | 0.022          | 0    | 1.423 |
+===============+===============+===============+======+=======+

SVD solution information (A*x = b):
 - Unknowns = 1; rank = 1.
 - Condition number:             1
```

```
 - Max error (Max(|A*x - b|)):          8.09928E-07
 - Relative error (||A*x - b||/||b||): 8.09928E-05%

Inductance [H]
Name       Design        Extracted     AbsDiff       PercDiff
L1         3.9E-12       3.89587E-12   -4.1257E-15   -0.10579%
```

The calculated inductance compares very well to the analytical result, but is a strong function of segmentation size, filament numbers and ground plane overlap. This is shown when sweep plots are discussed.

The three-dimensional model built by *InductEx* can be written to a three-dimensional format for viewing purposes. To write a VTK format file that can be viewed with ParaView, use:

```
inp2vtk microstrip.inp microstrip.vtk
```

The result is shown in Fig. 13.3. segments connected to positive port terminals are coloured dark red, and segments connected to negative port terminals are coloured blue.

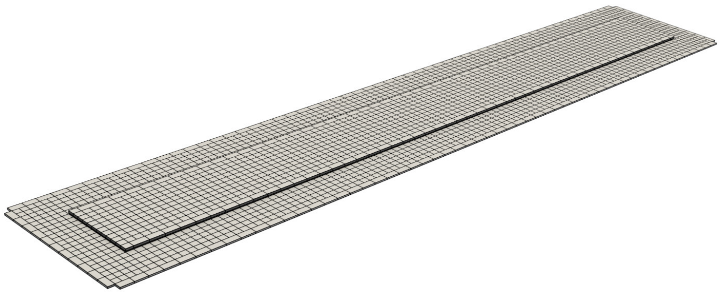


Figure 13.3: Interleaved cuboid filament model as built by *InductEx*.

Example works with Professional and Super licence tiers

A triangular mesh can be used by executing *InductEx* with:

```
inductex microstrip.gds -l process.ldf -i microstrip.geo -th
```

or

```
inductex microstrip.gds -l process.ldf -i microstrip.geo
```

It is not necessary to include the `-th` switch, because triangular meshing is the default option.

InductEx builds a geometry file of the layout structure, which is meshed with Gmsh. The resultant triangular mesh can be viewed directly with Gmsh, and is shown in Fig. 13.4. The solution is shown below.

Command Prompt

```
Inductance [H]
Name       Design        Extracted     AbsDiff       PercDiff
L1         3.9E-12       3.76489E-12   -1.3511E-13   -3.4644%
```

InductEx can also be instructed to solve the inductance from a tetrahedral mesh with:

```
inductex microstrip.gds -l process.ldf -i microstrip.geo -zoff
```

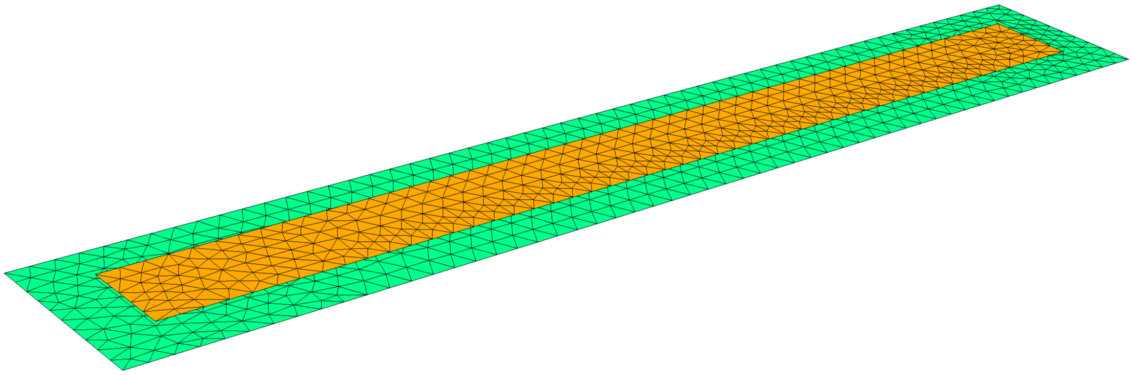


Figure 13.4: Triangular mesh model.

Note that the results are slightly higher than for the triangular mesh, due to current flow inside the tetra volumes.

Command Prompt

```
Inductance [H]
Name      Design      Extracted     AbsDiff      PercDiff
L1        3.9E-12     4.09953E-12   +1.9953E-13  +5.1161%
```

Modelling mistakes

If the microstrip inductance extraction model is defined to have one port (P_1) in parallel with an inductor L_1, and the positive and negative terminals of P_1 are defined at both ends of the conductor as shown in Fig. 13.5, the inductance of the line is calculated with a current return path at infinity. This is not correct.

The calculation result is L_1 = 58.33 pH, which is very far from the correct result.

Figure 13.5: Incorrect definition of ports for microstrip inductance calculation.

13.2 Washer

Example works with all licence tiers

This basic example, available in `examples/inductance/washer/`, demonstrates:

- Single conductor layer without gound plane.
- Alphanumeric port and inductor names.
- IXI input file.
- Virtual cut terminal.
- Current density plot.
- Flux ports.

Consider the calculation of the inductance of a square washer without a ground plane, as described in [13] and shown in Fig. 13.6 Instead of using a GDS input file, the washer geometry can be defined in an IXI file.

The washer polygon is created to have a hole with diameter 10 µm and material width of 10 µm. The outer dimensions are thus 30 µm × 30 µm.

The washer must be broken open to add a port for current injection and extraction. This can be done by drawing a polygon that has a thin channel that connects the hole to the outside of the washer, and then placing a positive and negative terminal on the opposing sides of the channel.

However, it is easier just to add a virtual cut to the washer. To do this, a path object is drawn on the TERM layer across the washer to connect the hole to the outside. This port, P_{one}, is defined as a path with non-zero width as shown in Fig. 13.7. The port is labelled `Pone/ ybco +y`. The / after the name indicates a virtual port. The second parameter is the layer name, YBCO, and the third parameter indicates that positive current injection is in the positive y direction.

The IXI file is `washer.ixi`.

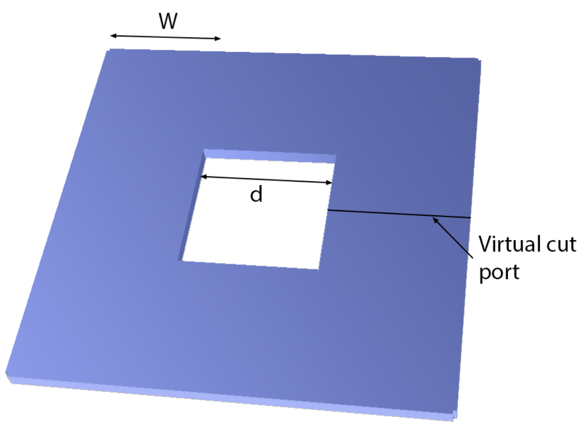


Figure 13.6: Square washer with arm width W and hole diameter d.

Ground plane filling is disabled with the parameter `ProcessHasGroundPlane = False`.
There is only one layer material (with order = 0), so that `LastDieLayerOrder = 0`.

■ Netlist 13.2.1 — washer.cir.

```
* Title: Netlist for InductEx user manual - monolayer washer
//--- Inductors ---
Lwasher 1   0   2E-11
//----- Ports -----
Pone    1   0
```

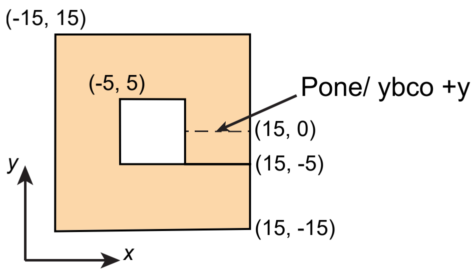


Figure 13.7: Geometry of a washer layout in layer YBCO.

```
.end
```

IXI File 13.2.2 — washer.ixi.

```
$STRUCT
  Name washer
  $POLY
    Layer ybco
    XY
    (  15 ,  -5 )
    (  15 , -15 )
    ( -15 , -15 )
    ( -15 ,  15 )
    (  15 ,  15 )
    (  15 ,  -5 )
    (   5 ,  -5 )
    (   5 ,   5 )
    (  -5 ,   5 )
    (  -5 ,  -5 )
    (  15 ,  -5 )
  $END
  $PATH
    Layer term
    Width 0
    XY
    (  5 ,   0 )
    ( 15 ,   0 )
  $END
  $TEXT
    Text "Pone/ ybco +y"
    XY
    (  10 ,   0 )
  $END
$END
```

Layer Definition File 13.2.3 — monolayer.ldf.

```
* ------------------
* PARAMETERS
* ------------------
$Parameters
  Units                  =  1e-6
  SegmentSize            =  0.5
  AbsMin                 =  0.1
  ProcessHasGroundPlane = FALSE
  TermLayer              =  19
  TextLayer              =  18
  TerminalInRange        =  1.0
  VirtualGap             =  0.25
$End
* ------------------
* LAYERS
* ---- YBCO --------
$Layer
  Number      =      1
  Name        =      YBCO
  Thickness   =      0.2
  Lambda      =      0.24
  Order       =      0
  Mask        =      1
  Filmtype    =      S
$End
```

Execute *InductEx* with:

```
inductex washer.ixi -l monolayer.ldf -i washer.inp -fh
```

An excerpt of the calculation result is shown below as it is printed to screen in the terminal or to the result output file `sol.txt`.

Command Prompt

```
  Ports:
+====+======+======+===========+
| Nr | Name | Type | Amplitude |
+----+------+------+-----------+
| 0  | pone | v    | 1         |
+====+======+======+===========+
  Terminals:
+=======+========+=======+===================+========+
| Name  | Layout | Layer | Geometry          | Object |
+-------+--------+-------+-------------------+--------+
```

```
 | pone+ | 0      | ybco  | Line/Edge Surface | 0      |
 | pone- | 0      | ybco  | Line/Edge Surface | 0      |
+=======+========+=======+==================+========+
  Mesh info:
+============+===============+===============+==========================+
| #Filaments | X-width range | Y-width range | Worst L/W ratios         |
+------------+---------------+---------------+--------------------------+
| 6868       | 0.476 - 0.477 | 0.368 - 0.488 | min = 0.7715; max = 1.908 |
+============+===============+===============+==========================+
  Time allocation (s):
+================+================+================+======+=======+
| Process layout | Slice surfaces | Build 3D model | Mesh | TTH   |
+----------------+----------------+----------------+------+-------+
| 0              | 0.001          | 0.014          | 0    | 1.398 |
+================+================+================+======+=======+

SVD solution information (A*x = b):
 - Unknowns = 1; rank = 1.
 - Condition number:                  1
 - Max error (Max(|A*x - b|)):        3.63095E-06
 - Relative error (||A*x - b||/||b||): 0.000363095%

Inductance [H]
Name      Design        Extracted     AbsDiff       PercDiff
LWASHER   2E-11         2.00874E-11   +8.7423E-14   +0.43711%
```

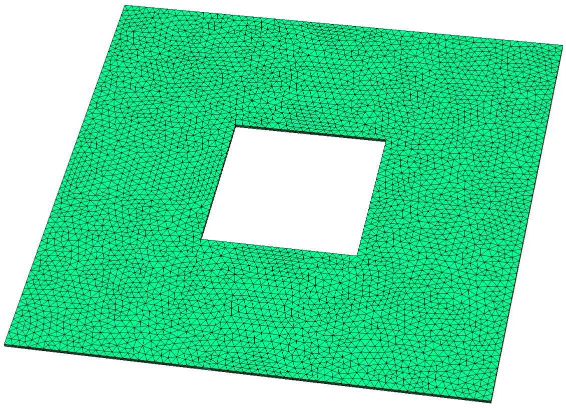


Figure 13.8: Meshed washer with virtual cut *InductEx* for port P_{one}.

Example works with Professional and Super licence tiers

To use a more precise tetrahedral mesh for inductance calculation, and to generate a plot of the current density distribution, execute *InductEx* with:

```
inductex washer.ixi -l monolayer.ldf -i washer.inp -th -zoff -j
```

With triangular and tetrahedral meshes, the virtual cut port is defined in the mesh without InductEx having to slit open the structure. A three-dimensional rendering of the meshed washer can be viewed with Gmsh, and is shown in Fig. 13.8.

An excerpt of the calculation result with a tetrahedral mesh is shown below.

Command Prompt

```
Name      Design        Extracted     AbsDiff       PercDiff
LWASHER   2E-11         2.02057E-11   +2.0567E-13   +1.0284%
```

The current density throughout the model is plotted to `/output/J_washer.vtk`. It can be visualised with ParaView, as shown in Fig. 13.9.

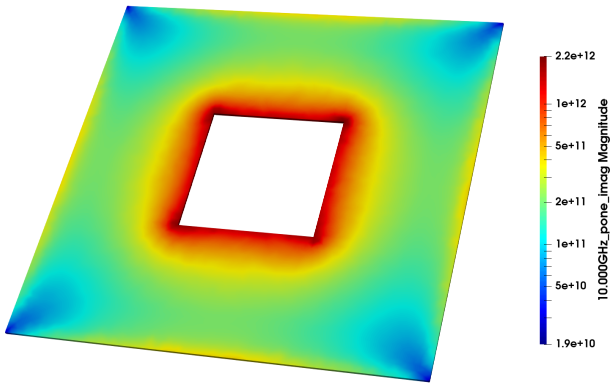


Figure 13.9: Current density over washer when port P_{one} is excited with 1 V at 10 GHz.

A more elegant solution to the inductance of the washer loop, provided that it is superconducting, involves no terminals on the structure. A flux port in the hole can be defined with a label anywhere inside the hole, as shown in IXI file `washerhole.ixi` (13.2.4).

■ IXI File 13.2.4 — washerhole.ixi.

```
$STRUCT
  Name washer
  $POLY
    Layer ybco
    XY
    (  15 ,  -5 )
    (  15 , -15 )
    ( -15 , -15 )
    ( -15 ,  15 )
    (  15 ,  15 )
    (  15 ,  -5 )
    (   5 ,  -5 )
    (   5 ,   5 )
    (  -5 ,   5 )
    (  -5 ,  -5 )
    (  15 ,  -5 )
  $END
  $TEXT
    Text "Fone"
    XY
    (  0 ,  0 )
  $END
$END
```

No circuit netlist is used for the solution. Use the `-noports` switch to let *InductEx* know that there are no voltage or current ports on the layout, and thus no netlist. Execute *InductEx* with:

```
inductex washerhole.ixi -l monolayer.ldf -i washerhole.geo -th -noports
```

The result is shown below:

Command Prompt

```
Inductance [H]
Name      Design        Extracted     AbsDiff       PercDiff
LFONE     --            2.00994E-11   +2.0099E-11   --%
```

The current density in the washer when a flux port through the hole is used for inductance extraction is shown in Fig. 13.10. Note that the washer remains intact.

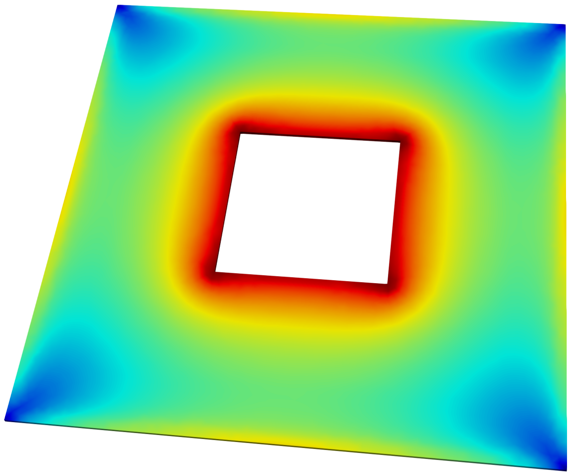


Figure 13.10: Current density over washer when phase around the hole is 2π.

13.3 Coplanar waveguide

Example works with Professional and Super licence tiers

This example, available in `examples/inductance/cpw/`, demonstrates:

- Ports with multiple terminals of the same polarity.
- Variables and parameterised dimensions in IXI file.
- Nested structures in IXI file.
- Paths in IXI file.
- Parameter list entry through IXI file.
- Terminal labelling directly on polygons and paths in IXI file.
- Tetrahedral meshes and *TetraHenry*.
- Parameter sweeps.

For this example, only the inductance of a coplanar waveguide structure is calculated. A rendering of the structure with dimensions and the required port connections is shown in Fig. 13.11.

For inductance calculation, return current must flow through both ground plane sections. Two ports need to be declared; one at each end of the conductor. If the positive terminal of each port is connected to the signal line, then the negative terminal of each port must be connected to both ground plane sections.

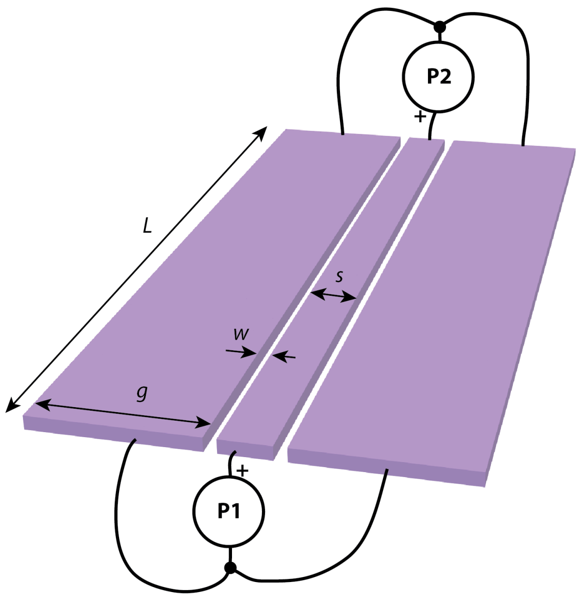


Figure 13.11: Coplanar waveguide structure with two ports for inductance calculation.

The netlist contains one inductor and two ports.

The parameter string is defined in the IXI file in the `$PARAMSTRING` record, so that only the IXI file name needs to be entered at the command line. The geometry is parameterised in terms of the variables declared in the IXI file. For demonstration purposes, the ground section is defined as a structure that is referenced twice through the `$SREF` record when the main geometry is defined.

Terminal objects are defined in the main structure as polygons or paths on the edges of the conductor and ground plane sections. Labels added with the `Text` parameter to these polygons and paths define the ports and terminals.

Note that the negative terminals of both ports P_1 and P_2 are defined twice; once each for the left hand side and right hand side ground sections.

Netlist 13.3.1 — cpw.cir.

```
* Title: Netlist for InductEx user manual - CPW
//--- Inductors ---
L1      1   2
//----- Ports -----
P1      1   0
P2      2   0
.end
```

IXI File 13.3.2 — cpw.ixi.

```
$PARAMSTRING
  -l process_cpw.ldf -i cpw.inp
  -fh   // Parameter string can
  -n cpw.cir    // be broken up
$END
// ------------------
//      VARIABLES
// ------------------
$VARIABLE
  Name  L    // Line length
  Value 100
$END
$VARIABLE
  Name  s    // Line width
  Value 6
$END
$VARIABLE
  Name  w    // Gap width
  Value 1.5
$END
$VARIABLE
  Name  g    // Ground width
  Value 20
$END
// ------------------
//      STRUCTURES
// ----- Ground -----
$STRUCT
  Name Ground
  $POLY
    Layer M0
    XY
    ( 0.0   ,  %g/2 )
    ( 0.0   , -%g/2 )
    ( %L    , -%g/2 )
    ( %L    ,  %g/2 )
    ( 0.0   ,  %g/2 )
  $END
$END
// --- Main (Top) ---
$STRUCT
  Name Main
  $SREF  // Add ground one side
    Name   Ground
      Reflect False
      Angle   0
      XY
      ( 0.0 , %s/2 + %w +%g/2 )
  $END
  $SREF  // Add opposite ground
    Name   Ground
      Reflect False
      Angle   0
      XY
      ( 0.0 , -%s/2 -%w -%g/2 )
  $END
  $POLY
    Layer M0 // Conductor
    XY
    ( 0.0 ,  %s/2 )
    ( 0.0 , -%s/2 )
    ( %L  , -%s/2 )
    ( %L  ,  %s/2 )
    ( 0.0 ,  %s/2 )
  $END
  $POLY   // Port P1 + terminal
    Layer term
    Text  "P1+ m0" // Label
    XY
    ( 0.0 , -%s/2 )
    ( 0.0 ,  %s/2 )
  $END
  $POLY
    Layer term
    Text  "P2+ m0"
    XY
    ( %L , -%s/2 )
    ( %L ,  %s/2 )
  $END
  $PATH
    Layer term
    Text  "P1- m0"
    XY
    ( 0.0 , %s/2 + %w )
    ( 0.0 , %s/2 + %w + %g )
  $END
  $PATH  // Path also works
    Layer term
    Text  "P1- m0"
    XY
    ( 0.0 , -%s/2 - %w - %g )
    ( 0.0 , -%s/2 - %w )
  $END
  // Second negative terminal
  // of port P1; electrically
  // equivalent to first.
  $PATH
    Layer term
    Text  "P2- m0"
    XY
    ( %L , %s/2 + %w )
    ( %L , %s/2 + %w + %g )
  $END
  $PATH
    Layer term
    Text  "P2- m0"
    XY
    ( %L , -%s/2 - %w - %g )
    ( %L , -%s/2 - %w )
  $END
$END
```

Execute *InductEx* with:

```
inductex cpw.ixi
```

An excerpt of the calculation result is shown as it is printed to screen in the terminal or to the result output file `sol.txt`. From the output, it is seen that both ports P_1 and P_2 have two negative terminals in layer `M0`.

■ **Layer Definition File 13.3.3 — process_cpw.ldf.**

```
* ------------------
* PARAMETERS
* ------------------
$Parameters
  Units               =  1e-6
  SegmentSize         =  1.0
  AbsMin              =  0.025
  ProcessHasGroundPlane = TRUE
  GPLayer             =  17
  TermLayer           =  19
  TextLayer           =  18
  Lambda              =  0.09
  Sigma               =  10
  HFilaments          =  1
  TerminalInRange     =  1.0
$End
* ------------------
* LAYERS
* ------- M0 -------
$Layer
  Number      =     17
  Name        =     M0
  Bias        =     0
  Thickness   =     0.17
  Lambda      =     0.085
  Order       =     0
  Mask        =     1 // Positive
  Filmtype    =     S
  HFilaments  =     1
  Colour      =     130
$End
* ------ TERM ------
$Layer
  Number      =     19
  Name        =     TERM
  Order       =     1
  Mask        =    -4
$End
```

Command Prompt

```
Inductance [H]
Name      Design       Extracted     AbsDiff       PercDiff
L1        --           3.48454E-11   +3.4845E-11   --%
```

The parameterised layout allows geometry sweeps. Sweep commands can be added to the netlist file, as shown in Netlist File 13.3.4. For this example, two sweeps are defined. The first sweep statement sweeps the value of global SegmentSize from 1 to 4 in increments of 0.5 (all units in µm). The second sweep statement is a double sweep of two variables: gap width w from 0.5 to 1.5 in steps of 0.25, and line width s from 2 to 6 in steps of 2 (all units in µm).

■ **Netlist 13.3.4 — cpwsweep.cir.**

```
* Title: Netlist for InductEx user manual - CPW
// -----------------
//   CONTROL BLOCK
// -----------------
.control
  SWEEP SegmentSize 1 0.5 3.5
  SWEEP variable w 0.5 0.25 1.5 variable s 2 2 6
.endc
//--- Inductors ----
L1      1   2
//----- Ports ------
P1      1   0
P2      2   0
.end
```

After execution of *InductEx*, the sweep results are written in Matlab script format to files `sweep0_view.m` and `sweep1_view.m`. The plots generated by Matlab are shown in Fig. 13.12 and Fig. 13.13.

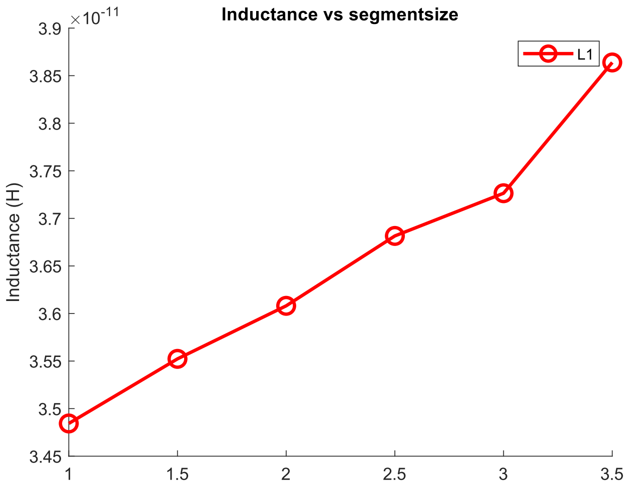


Figure 13.12: Inductance as a function of model segment size for a coplanar waveguide structure.

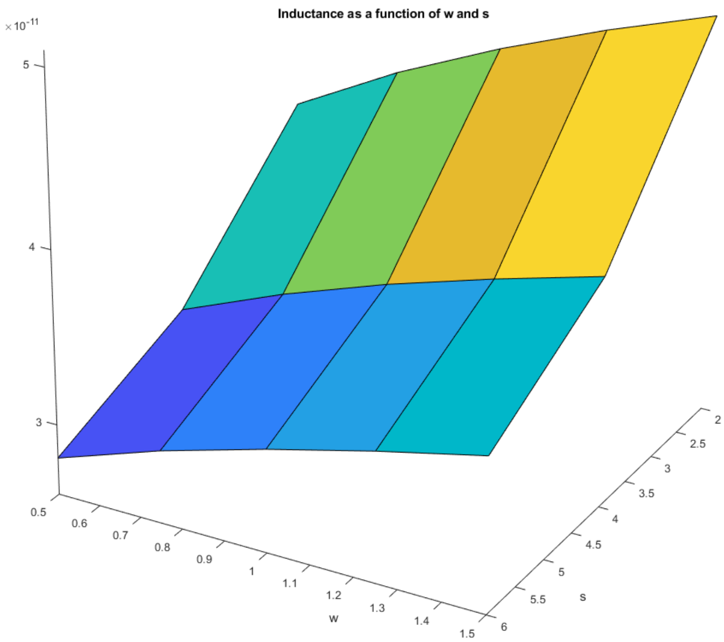


Figure 13.13: Inductance as a function of line width s and gap width w for a CPW structure.

13.4 SQUID with mutual inductance

Example works with Professional and Super licence tiers

This example, available in `examples/inductance/squid/`, demonstrates:
- Mutual inductance.
- Coupling.

For this example, the inductance of a SQUID loop and a coupling coil are calculated, as well as the mutual inductance between the inductive loops. The example is based on a SQUID implementing a gradiometer for a microcalorimeter [14].

The circuit schematic is shown in Fig. 13.14 and the corresponding netlist with mutual inductances defined as K elements (similar to SPICE) is shown in Netlist 13.4.1.

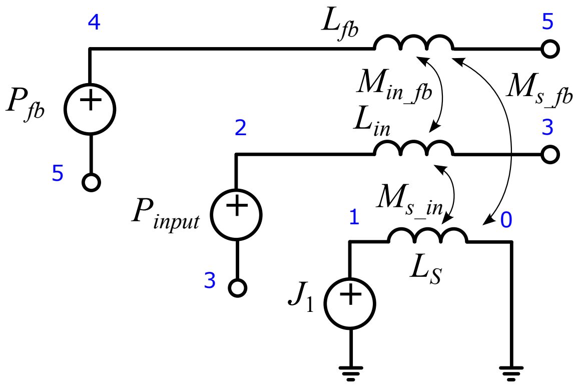


Figure 13.14: Netlist of the self and mutual inductances and port placement of the SQUID for this example.

Netlist 13.4.1 — squid.cir.

```
* Title: Netlist for SQUID gradiometer
Ls      1   0   45p
Lin     2   3   1.8n
Lfb     4   5   675p
J1      1   0
Pinput  2   3
Pfb     4   5
Ks_in   Ls  Lin
Ks_fb   Ls  Lfb
Kin_fb  Lin Lfb
.end
```

The SQUID model with its port declarations is shown in Fig. 13.15.

Execute *InductEx* with:

```
inductex squid.gds -l process.ldf -i squid.geo -th
```

■ **Layer Definition File 13.4.2 — process.ldf.**

```
$Parameters
  Units                = 1e-6
  AbsMin               = 0.01
  SegmentSize          = 3.0
  ProcessHasGroundPlane = FALSE
  GPLayer              = 1
  TermLayer            = 15
  TextLayer            = 64
  Lambda               = 0.09
  Sigma                = 10e6
  HFilaments           = 1
  DecimationDistance = 0.05
$End
* ----------- NB1 -----------
* NB1 - lower Niobium layer
$Layer
  Number     =     4
  Name       =     NB1
  Thickness  =     0.25
  Lambda     =     0.09
  Order      =     0
  Mask       =     1
  Filmtype   =     S
  HFilaments =     3
$End
* ----------- JJ ------------
$Layer
  Number     =     6
  Name       =     JJ
  Thickness  =     0.143
  Order      =     1
  Mask       =     0
  Filmtype   =     A
$End
* ----------- ISO -----------
$Layer
  Number     =     7
  Name       =     BC
  Thickness  =     0.43
  Order      =     2
  Mask       =    -1
  Filmtype   =     I
$End
* ----------- A1 ------------
// A1 - Gold layer for Rshunt
$Layer
  Number     =     12
  Name       =     A1
  Thickness  =     0.24
// Sputtered Au sigma [S/um]
  Sigma      =     42
// Plated Au = 35 S/um
  Order      =     3
  Mask       =     1
  Filmtype   =     N
  ViaBypass  =     TRUE
  Colour     =     34
$End
* ----------- NB2 -----------
* NB2 - upper Niobium layer
$Layer
  Number     =     8
  Name       =     NB2
  Thickness  =     0.5
  Lambda     =     0.09
  Order      =     4
  Mask       =     1
  Filmtype   =     S
  HFilaments =     3
$End
```

Command Prompt

```
Inductance [H]
Name      Design        Extracted     AbsDiff       PercDiff
LS        4.5E-11       4.78038E-11   +2.8038E-12   +6.2306%
LIN       1.8E-09       1.84387E-09   +4.3874E-11   +2.4374%
LFB       6.75E-10      7.22508E-10   +4.7508E-11   +7.0382%

Mutual Inductance [H]                                             Coupling factor
Name      Design        Extracted     AbsDiff       PercDiff             k
MS_IN     ---           -1.6342E-10   -1.6342E-10   --%           -0.55044
MS_FB     ---           -5.6754E-11   -5.6754E-11   --%           -0.30538
MIN_FB    ---           +9.4983E-11   +9.4983E-11   --%           +0.082292
```

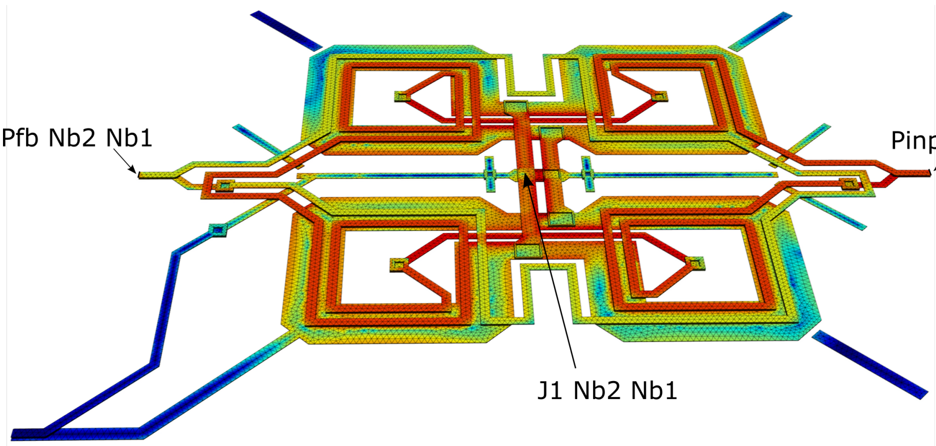


Figure 13.15: The meshed model of the SQUID gradiometer for a microcalorimeter, with port declarations and their positions. Current density is shown when port Pinput is excited.

13.5 RSFQ DC-SFQ circuit inductance

Example works with all licence tiers

This example, available in `examples/inductance/rsfq_dcsfq/`, demonstrates:

- Multi-port inductance network.
- Ports on Josephson junctions (or vias).
- Layer definition construction for a real fabrication process.
- Mask-to-wafer bias on layers.
- Junction area and critical current calculation.
- Unit size scaling (PSCAN dimensionless units).
- Layer subtraction.
- Resistance.

For this example an RSFQ DC-SFQ converter laid out for the Hypres 1 µm mask aligner lithography-based 4 layer Niobium process [1] (4.5 kA/cm^2) is demonstrated. The circuit schematic of the JoSIM [15] simulation model is shown in Fig. 13.16, and the layout is shown in Fig. 13.18.

For the first inductance extraction, resistance will be ignored. The netlist for *InductEx* is shown in Fig. 13.17. All Josephson junctions are replaced with ports (the polarity is important), while the input and output terminals are connected to ground with ports P_{in} and P_{out}. The bias current inputs are also replaced with ports I_{b1} and I_{b2}.

This example uses PSCAN [16] dimensionless units[2]. The dimensionless units are set up in Layer Definition File 13.6.2 as `UnitL = 2.64E-12`, `UnitR = 3.48` and `UnitI = 0.125E-3`.

The critical current density of layer A1, which defines Josephson junction area, is set to 45 µA/µm^2 with `IDensity = 45E-6`. This parameter is defined in A/(unit area). *InductEx* will then multiply the critical current density with the area of any port that uses this layer to calculate the critical current of an associated Josephson junction.

Layers R2 and R3 are defined with `FilmType = N`. Objects on these layers will not be meshed, but will be taken into account when the height of objects on higher layers is calculated. Since no objects remain on these layers, vias from layer M2 to R2 will short all the way down to M1. To prevent this, `LayerSUB = 9` is added to the I1B layer definition. This instructs *InductEx* to subtract objects in layer 9 (R2) from objects in layer I1B, which effectively removes vias that terminate in resistors.

In the layout, it is only necessary to draw terminal objects for P_{in}, P_{out}, I_{B1} and I_{B2}. In this example, paths are used to define the terminals. Resistive objects will not be meshed, so that IB1 and IB2 have to be placed between the resistors and the circuit. The inductance of the bias branches is not important in this example, so that the location of the bias branch ports is irrelevent.

If a port label is placed inside a via on an isolation layer, *InductEx* uses the smallest via in the stack to insert a terminal. It is thus not necessary to draw terminal objects for the Josephson junctions. Junction port labels are just placed at coordinates inside the respective junction objects.

Also note that, for this example, mask-to-wafer bias is defined for most of the layers. For layer M2 with its negative bias, for example, every object will shrink by 0.2 µm on every edge (thus, every edge moves 0.2 µm inwards) from mask (layout) to wafer (meshed calculation model).

[1] SeeQC, Inc. acquired the Hypres fabrication process. Information at https://seeqc.com/chip-foundry-services/chip-fabrication

[2] See http://www.physics.sunysb.edu/Physics/RSFQ/Lib/units.html and http://pscan2sim.org/documentation.html

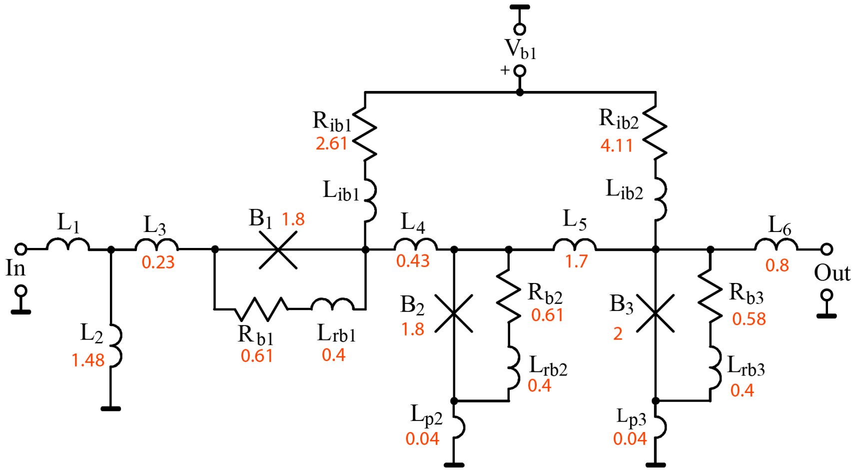


Figure 13.16: JoSIM schematic of DC-SFQ circuit.

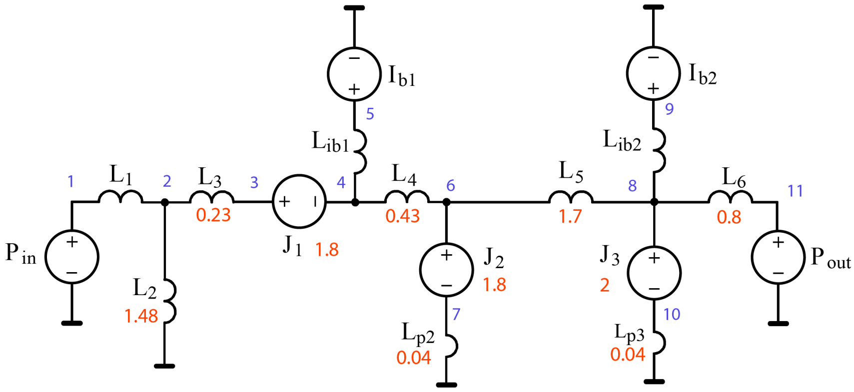


Figure 13.17: *InductEx* netlist for DC-SFQ inductance extraction without resistive branches.

■ **Netlist 13.5.1 — dcsfq.cir.**

```
* Title: Netlist for DC-SFQ
//  PSCAN dimensionless units
//  UnitL  =  2.64  pH
//  UnitR  =  3.48  Ohm
//  UnitI  =  0.125 mA
//--- Inductors ---
L1     1   2   0.1
L2     2   0   1.48
L3     2   3   0.23
L4     4   6   0.43
L5     6   8   1.7
L6     8   11  0.8
Lp2    7   0   0.04
Lp3    10  0   0.04
LIB1   4   5
LIB2   8   9
//----- Ports -----
Pin    1   0
J1     3   4    1.8
Ib1    5   0
J2     6   7    1.8
J3     8   10   2
Ib2    9   0
Pout   11  0
.end
```

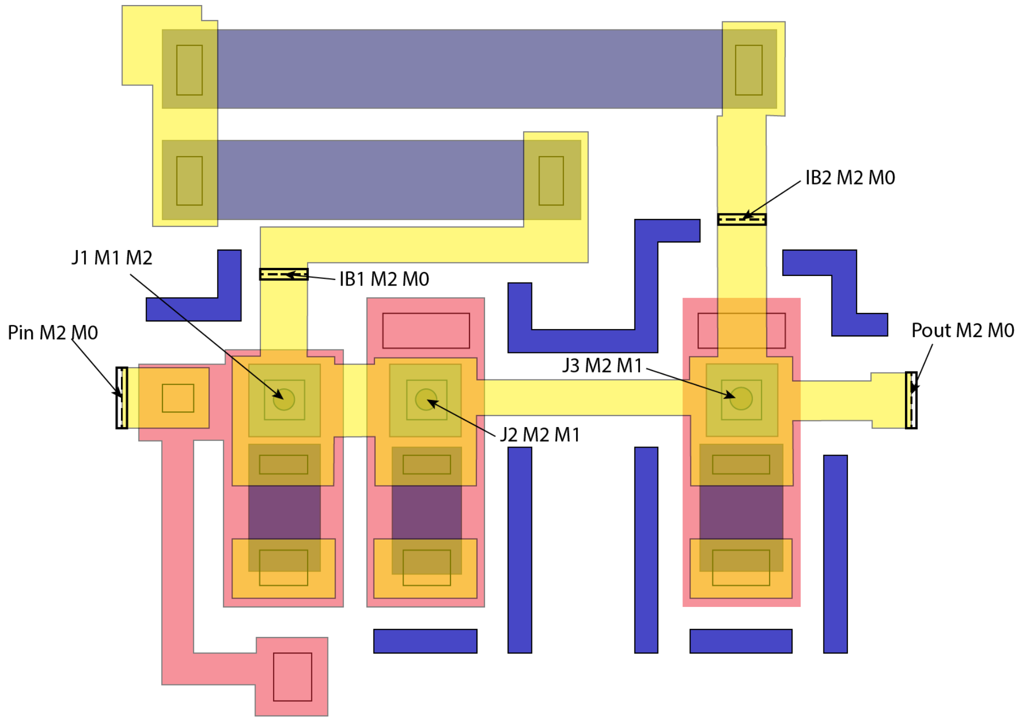


Figure 13.18: Layout of an RSFQ DC-SFQ converter.

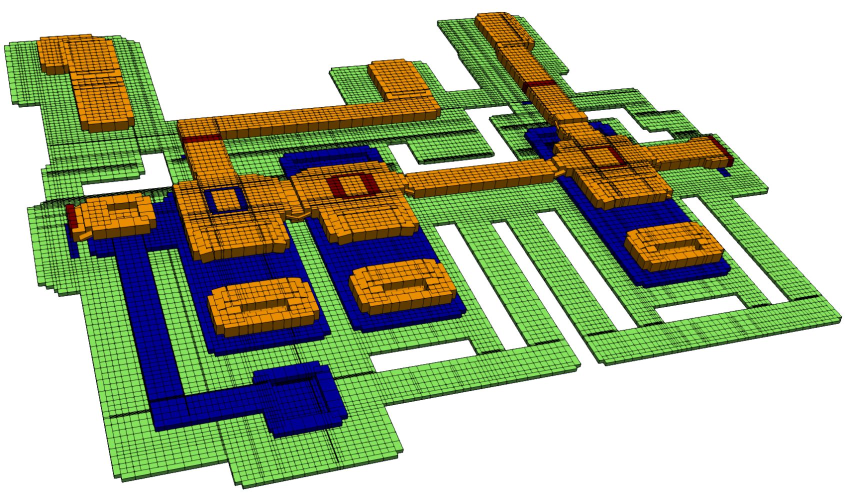


Figure 13.19: Three-dimensional rendering of DC-SFQ inductance extraction model.

■ **Layer Definition File 13.5.2 — h4k5.ldf.**

```
$Parameters
  Units          =  1e-6  // um
  AbsMin         =  0.001
  SegmentSize    =  2.0
  GPOverhang     =  5
  ProcessHasGroundPlane = TRUE
  GPLayer        =  30
  TermLayer      =  63
  TextLayer      =  64
  Lambda         =  0.09
  HFilaments     =  1
  UnitL          =  2.64E-12
  UnitR          =  3.48
  UnitI          =  0.125e-3
  TerminalInRange =  1.0
  DecimationDistance = 0.05
$End
* ------ M0 --------
$Layer
  Number      =  30
  Name        =  M0
  Bias        =  0.2
  Thickness   =  0.1
  Lambda      =  0.09
  Order       =  0
  Mask        = -1
  Filmtype    =  S
  HFilaments  =  1
$End
* ------ I0 --------
$Layer
  Number      =  31
  Name        =  I0
  Bias        =  0.2
  Thickness   =  0.15
  Order       =  1
  Mask        = -1
  Filmtype    =  I
$End
* ------ M1 --------
$Layer
  Number      =  1
  Name        =  M1
  Bias        =  0
  Thickness   =  0.135
  Lambda      =  0.09
  Order       =  2
  Mask        =  1
  Filmtype    =  S
  HFilaments  =  2
$End
* ------ I1C -------
$Layer
  Number      =  4
  Name        =  I1C
  Bias        =  0
  Thickness   =  0.05
  Order       =  3
  Mask        =  0
  IDensity    =  45e-6
  Filmtype    =  A
$End
* ------ A1 --------
$Layer
  Number      =  5
  Name        =  A1
  Thickness   =  0.04
  Order       =  4
  Mask        =  0
  Filmtype    =  A
$End
* ------ R2 --------
$Layer
  Number      =  9
  Name        =  R2
  Bias        =  0
  Thickness   =  0.07
  Sigma       =  10e6
  Order       =  5
  Mask        =  1
  Filmtype    =  N
$End
* ------ I1B -------
$Layer
  Number      =  3
  Name        =  I1B
  Bias        = -0.1
  Thickness   =  0.2
  Order       =  6
  Mask        = -1
  Filmtype    =  I
  LayerSUB    =  9
$End
* ------ M2 --------
$Layer
  Number      =  6
  Name        =  M2
  Bias        = -0.2
  Thickness   =  0.3
  Lambda      =  0.09
  Order       =  7
  Mask        =  1
  Filmtype    =  S
  HFilaments  =  3
$End
* ------ I2 --------
$Layer
  Number      =  8
  Name        =  I2
  Bias        =  0.2
  Thickness   =  0.5
  Order       =  8
  Mask        = -1
  Filmtype    =  I
$End
* ------ M3 --------
$Layer
  Number      =  10
  Name        =  M3
  Bias        = -0.4
  Thickness   =  0.6
  Lambda      =  0.09
  Order       =  9
  Mask        =  1
  Filmtype    =  S
  HFilaments  =  3
$End
* ------ R3 --------
$Layer
  Number      =  11
  Name        =  R3
  Thickness   =  0.35
  Order       =  10
  Mask        =  1
  Filmtype    =  N
$End
```

Execute *InductEx* with:

```
inductex dcsfq.gds -l h4k5.ldf -i dcsfq.inp -fh
```

The output is shows all the extracted inductances. Undefined design values are printed as "--". Note that the inductance, resistance and junction critical current units are printed as "[--]" to indicate that these are not standard metric units.

Command Prompt

```
SVD solution information (A*x = b):
 - Unknowns = 10; rank = 10.
 - Condition number:              7.895
 - Max error (Max(|A*x - b|)):    0.0303099
 - Relative error (||A*x - b||/||b||): 1.38023%
```

```
Inductance [--]
Name       Design        Extracted     AbsDiff       PercDiff
L1         0.1           0.224577      +0.12458      +124.58%
L2         1.48          1.53025       +0.050251     +3.3953%
L3         0.23          0.190452      -0.039548     -17.195%
L4         0.43          0.370492      -0.059508     -13.839%
L5         1.7           1.67187       -0.028132     -1.6548%
L6         0.8           0.775175      -0.024825     -3.1031%
LP2        0.04          0.0403521     +0.00035211   +0.88027%
LP3        0.04          0.0276436     -0.012356     -30.891%
LIB1       --            0.489846      +0.48985      --%
LIB2       --            0.655001      +0.655        --%

Junction   Critical current [--]
Name       Design        Extracted     AbsDiff       PercDiff
J1         1.8           1.7874        -0.0126       -0.7%
J2         1.8           1.7874        -0.0126       -0.7%
J3         2             1.9998        -0.0002       -0.01%
```

A three-dimensional rendering can be constructed with *Inp2vtk*, with the command

```
inp2vtk dcsfq.inp dcsfq.vtk -h 5
```

The three-dimensional rendering is stretched vertically by a factor of five for visual clarity. It is shown in Fig. 13.19.

It can be noted that the port terminals can be identified by colour: red for all segments connected to positive terminal nodes, and blue for all segments connected to negative terminal nodes. Ground plane cropping proceeds to 5 µm from any non-groundplane structure (as set with `CropGP` in the `h4k5.ldf`, except where a moat overlaps the cropped ground plane. In those cases, the ground plane wraps around the moat in a band with the same width as `CropGP`.

The resistive layers are defined with `FilmType = N`, which denotes normal conductors that are not meshed for inclusion in the extraction model. Structures on these layers do however influence the height of layers above. The effect of the unmeshed resistors can thus be seen in the three-dimensional meshed structure.

Modelling mistakes

The polarity of a port is very important. Note that port J_3 is defined with its positive terminal connected to L_3 in the Netlist 13.5.1. In the layout, that corresponds to the terminal of the Josephson junction on layer M1, so that the port label on the layout is "`J1 M1 M2`" If the port label is erroneously defined as "`J1 M2 M1`", the calculation result is wrong:

Command Prompt

```
Impedance    Inductance [--]         Resistance [--]       AbsDiff      PercDiff
Name       Design      Extracted    Design      Extracted    (L only)     (L only)
L1         0.1         2.48534      --          --           +2.3853      +2385.3%
L2         1.48        -1.12448     --          --           -2.6045      -175.98%
L3         0.23        2.5976       --          --           +2.3676      +1029.4%
L4         0.43        0.00740269   --          --           -0.4226      -98.278%
L5         1.7         -0.0346786   --          --           -1.7347      -102.04%
L6         0.8         0.481071     --          --           -0.31893     -39.866%
LP2        0.04        0.393604     --          --           +0.3536      +884.01%
LP3        0.04        0.199185     --          --           +0.15919     +397.96%
LIB1       --          0.445954     --          --           +0.44595     --%
LIB2       --          0.395247     --          --           +0.39525     --%
```

13.6 RSFQ DC-SFQ circuit with resistance

Example works with all licence tiers

This example, available in `examples/resistance/rsfq_dcsfq_res/`, demonstrates:

- Resistance setup in layer definition file.
- Addition of resistance as parameter to inductors in netlist.
- Current density plotting when the sheet current method is used.

This example uses the same RSFQ DC-SFQ converter circuit as in Section 13.5, except that resistance is now included in the calculation.

The extraction netlist now has to be expanded to include the shunt resistance branches. More ports are needed to allow current calculation in each branch. As a rule, a port is added to each shunt resistance branch.

The bias resistors R_{ib1} and R_{ib2} will appear in series during extraction unless a further port (P_{dc} in this case) is added. An inductance, L_{dc} is necessary to prevent a branch with only a port. Every resistor has some inductance, to that L_{ib1} and L_{ib2} are placed in series with the two bias resistors. The resulting schematic for the extraction netlist is shown in Fig. 13.20. In this netlist, there is no inductance for the branches that contain the shunt resistors R_{s1}, R_{s2} and R_{s3}. *InductEx* will automatically add inductors in series with these resistors and name these L_{rs1_series}, L_{rs2_series} and L_{rs3_series} respectively.

Layer R2 is now defined with `FilmType = R`. Object on this layers will now be meshed, and have a conductivity as specified with `sigma`.

For this example, note the polarity of ports I_{b1} and I_{b2}. Port label placement is shown on the layout in Fig. 13.21.

The layer definition file is adapted to handle the resistance layer buried inside the isolation layer I1B. For this, a non-process layer is added for modelling purposes. This layer, I1BL implements the lower part of the I1B isolation below R2. Note that the layer order for layers above this one is stepped up.

For I1BL, all objects from layer I1B are added, after which all objects that overlap with objects on R2 are subtracted. This way, I1B vias from layer M2 downward will etch stop at R2 or M1. The `ViaBypass = True` parameter for layer R2 enables vias to drill past R2 when no conducting objects are present.

The conductivity of layer R2 is 2.1 Ω/□, and the layer thickness is 70 nm. The conductivity `sigma` in $\Omega^{-1}\,\mathrm{Units}^{-1}$ $\Omega^{-1}\mathrm{Units}^{-1}$ is thus $2.1^{-1}0.07^{-1}\Omega^{-1}\mu\mathrm{m}^{-1} = 6.803\,\Omega^{-1}\mu\mathrm{m}^{-1}$.

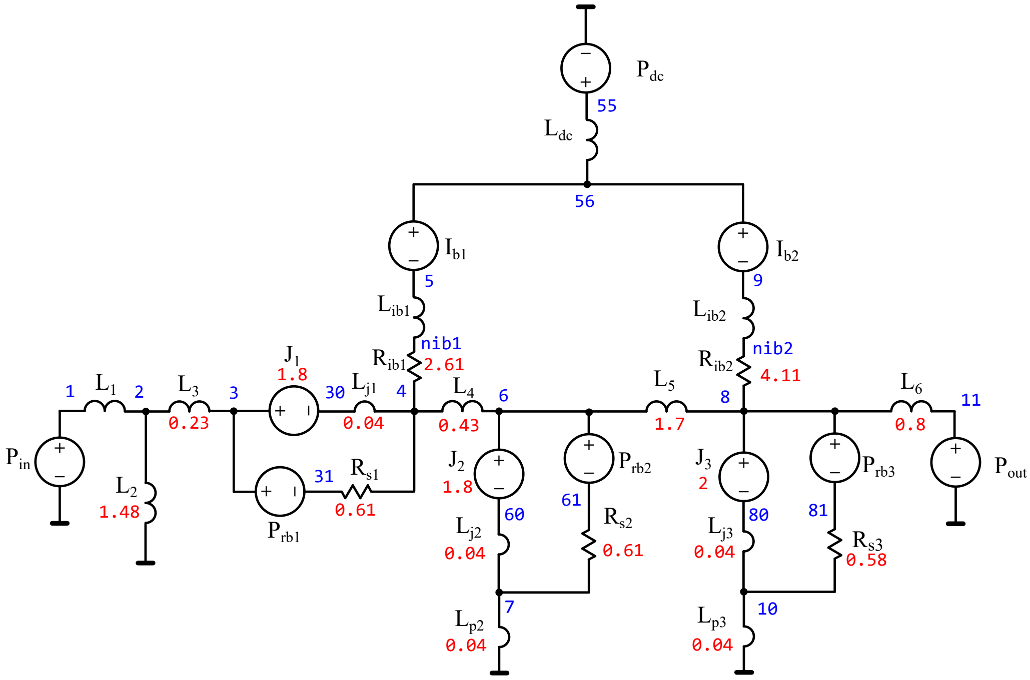


Figure 13.20: *InductEx* netlist for DC-SFQ inductance extraction with resistive branches.

■ Netlist 13.6.1 — dcsfqres.cir.

```
* Title: Netlist for DC-SFQ
//  UnitL  =  2.64  pH
//  UnitR  =  3.48  Ohm
//  UnitI  =  0.125 mA
//--- Inductors ---
L1      1     2     0.1
L2      2     0     1.48
L3      2     3     0.23
L4      4     6     0.43
L5      6     8     1.7
L6      8     11    0.8
Lj1     30    4     0.04
Lj2     60    7     0.04
Lp2     7     0     0.04
Lj3     80    10    0.04
Lp3     10    0     0.04
LIB1    5     nib1
LIB2    9     nib2
LDC     55    56
```

```
//--- Resistors ---
Rs1     31    4     0.61
Rs2     61    7     0.61
Rs3     81    10    0.58
Rib1    nib1  4     2.61
Rib2    nib2  8     4.11
//----- Ports -----
Pin     1     0
J1      3     30    1.8
Prb1    3     31
Ib1     56    5
J2      6     60    1.8
Prb2    6     61
J3      8     80    2
Prb3    8     81
Ib2     56    9
Pout    11    0
PDC     55    0
.end
```

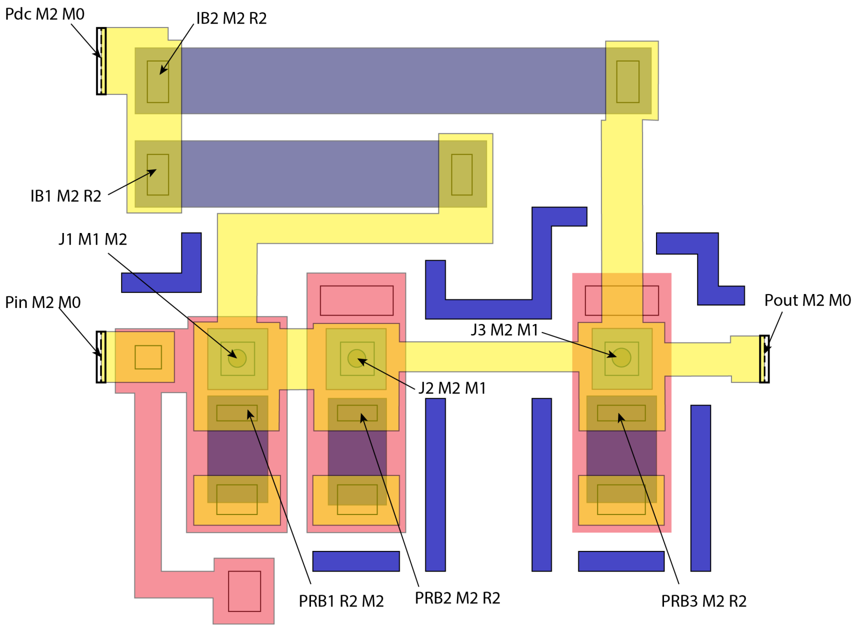


Figure 13.21: Layout of an RSFQ DC-SFQ converter with ports labelled or inductance and resistance extraction.

■ **Layer Definition File 13.6.2 — h4k5.ldf.**

```
$Parameters
  Units          =  1e-6  // um
  AbsMin         =  0.001
  SegmentSize    =  2.0
  GPOverhang     =  5
  ProcessHasGroundPlane = TRUE
  GPLayer        =  30
  TermLayer      =  63
  TextLayer      =  64
  Lambda         =  0.09
  UnitL          =  2.64E-12
  UnitR          =  3.48
  UnitI          =  0.125e-3
  TerminalInRange =  1.0
$End
* ------------------
*     LAYERS
* ------- M0 -------
$Layer
  Number     =  30
  Name       =  M0
  Bias       =  0.2
  Thickness  =  0.1
  Lambda     =  0.09
  Order      =  0
  Mask       = -1
  Filmtype   =  S
  HFilaments =  1
$End
* ------- I0 -------
$Layer
  Number     =  31
  Name       =  I0
  Bias       =  0.2
  Thickness  =  0.15
  Order      =  1
  Mask       = -1
  Filmtype   =  I
$End
* ------- M1 -------
$Layer
  Number     =  1
  Name       =  M1
  Thickness  =  0.135
  Lambda     =  0.09
  Order      =  2
  Mask       =  1
  Filmtype   =  S
  HFilaments =  2
$End
* ------- I1C ------
$Layer
  Number     =  4
  Name       =  I1C
  Bias       =  0
  Thickness  =  0.05
  Order      =  3
  Mask       =  0
  IDensity   =  4.5e-5
  Filmtype   =  A
$End
* ------- A1 -------
$Layer
  Number     =  5
  Name       =  A1
  Thickness  =  0.04
  Order      =  4
  Mask       =  0
  Filmtype   =  A
$End
* ------ I1BL ------
$Layer
  // Artifical layer
  // to allow R2 to be
  // buried in isolation
  Number     =     59
  Name       =     I1BL
  Bias       =    -0.1
  Thickness  =     0.1
  Order      =     5
  Mask       =    -1
  Filmtype   =     I
  LayerADD   =     3
  LayerSUB   =     9
  // Add layer 3 (IB1) to
  // create vias all the way
  // through isolation; then
  // subtract layer 9 (R2) to
  // prevent vias from M2 to
  // M1 drilling through
  // resistors
$End
* ------- R2 -------
$Layer
  Number     =     9
  Name       =     R2
  SegmentSize =    1.0
  Thickness  =     0.07
  Sigma      =     6.803
  Order      =     6
  Mask       =     1
  Filmtype   =     R
  ViaBypass  =     TRUE
$End
* ------- I1B ------
$Layer
  Number     =     3
  Name       =     I1B
  Bias       =    -0.1
  Thickness  =     0.1
  Order      =     7
  Mask       =    -1
  Filmtype   =     I
$End
* ------- M2 -------
$Layer
  Number     =     6
  Name       =     M2
  Bias       =    -0.2
  Thickness  =     0.3
  Lambda     =     0.09
  Order      =     8
  Mask       =     1
  Filmtype   =     S
  HFilaments =     3
$End
* ------- I2 -------
$Layer
  Number     =     8
  Name       =     I2
  Bias       =     0.2
  Thickness  =     0.5
  Order      =     9
  Mask       =    -1
  Filmtype   =     I
$End
* ------- M3 -------
$Layer
  Number     =     10
  Name       =     M3
  Bias       =    -0.4
  Thickness  =     0.6
  Lambda     =     0.09
  Order      =     10
  Mask       =     1
  Filmtype   =     S
  HFilaments =     3
$End
* ------- R3 -------
$Layer
  Number     =     11
  Name       =     R3
  Thickness  =     0.35
  Order      =     11
  Mask       =     1
  Filmtype   =     N
$End
* ------ TERM ------
$Layer
  Number     =     63
  Name       =     TERM
  Order      =     12
  Mask       =    -4
$End
```

Execute *InductEx* with:

```
inductex dcsfqres.gds -l h4k5res.ldf -i dcsfqres.inp -fh
```

The output shows all the extracted inductances. Only non-zero resistance values are printed. The resistance value for L_{dc} is an artefact.

Command Prompt

```
SVD solution information (A*x = b):
 - Unknowns = 22; rank = 22.
 - Condition number:                    508.9
 - Max error (Max(|A*x - b|)):          0.0311575
 - Relative error (||A*x - b||/||b||): 1.02554%

Inductance [--]
Name          Design        Extracted     AbsDiff       PercDiff
L1            0.1           0.226099      +0.1261       +126.1%
L2            1.48          1.53191       +0.051915     +3.5077%
L3            0.23          0.17545       -0.05455      -23.718%
L4            0.43          0.366266      -0.063734     -14.822%
L5            1.7           1.6628        -0.037205     -2.1885%
L6            0.8           0.774464      -0.025536     -3.192%
LJ1           0.04          0.0163177     -0.023682     -59.206%
LJ2           0.04          0.0151787     -0.024821     -62.053%
LP2           0.04          0.0271852     -0.012815     -32.037%
LJ3           0.04          0.0121733     -0.027827     -69.567%
LP3           0.04          0.0278019     -0.012198     -30.495%
LIB1          --            3.20568       +3.2057       --%
LIB2          --            2.35887       +2.3589       --%
LDC           --            0.233237      +0.23324      --%
LRS1_SERIES   --            0.249663      +0.24966      --%
LRS2_SERIES   --            0.253325      +0.25333      --%
LRS3_SERIES   --            0.24653       +0.24653      --%

Resistance [--]
Name          Design        Extracted     AbsDiff       PercDiff
RS1           0.61          0.702284      +0.092284     +15.129%
RS2           0.61          0.70192       +0.09192      +15.069%
RS3           0.58          0.631894      +0.051894     +8.9472%
RIB1          2.61          2.66605       +0.056049     +2.1475%
RIB2          4.11          4.13379       +0.023793     +0.57891%

Junction       Critical current [--]
Name          Design        Extracted     AbsDiff       PercDiff
J1            1.8           1.8           -0            -0%
J2            1.8           1.8           -0            -0%
J3            2             2.0016        +0.0016       +0.08%
```

A three-dimensional rendering can be constructed with *Inp2vtk*, with the command

```
inp2vtk dcsfq.inp dcsfq.vtk -h 5
```

The three-dimensional rendering is shown in Fig. 13.22. The resistors can now be seen as part of the meshed model. As can be seen, the cuboid filament method respects local change in layer height for a process (here, layer M2 clearly dips when no M1 is present below).

Example works with Professional and Super licence tiers

In order to support layer height variation when tetrahedral or triangular meshes are used, *InductEx* post-processes the mesh produced by Gmsh and adjusts node height as required.

To extract parameters for a triangular mesh with sheet current modelling, and to plot the current density, run *InductEx* with:

```
inductex dcsfqres.gds -l h4k5res.ldf -i dcsfqres.inp -th -j
```

The output is shown below.

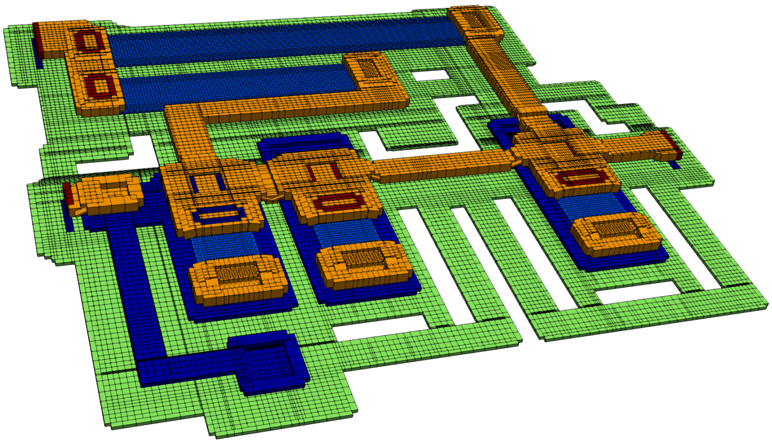


Figure 13.22: Three-dimensional rendering of cuboid filament mesh of DC-SFQ inductance extraction model with resistors included. Vertical dimension is multiplied by 5 for clarity.

Command Prompt

```
Inductance [--]
Name          Design        Extracted     AbsDiff       PercDiff
L1            0.1           0.21086       +0.11086      +110.86%
L2            1.48          1.45075       -0.029248     -1.9762%
L3            0.23          0.168337      -0.061663     -26.81%
L4            0.43          0.360825      -0.069175     -16.087%
L5            1.7           1.62764       -0.072357     -4.2563%
L6            0.8           0.752411      -0.047589     -5.9486%
LJ1           0.04          0.0181647     -0.021835     -54.588%
LJ2           0.04          0.0177175     -0.022282     -55.706%
LP2           0.04          0.0212999     -0.0187       -46.75%
LJ3           0.04          0.0142474     -0.025753     -64.382%
LP3           0.04          0.0224355     -0.017565     -43.911%
LIB1          --            3.10668       +3.1067       --%
LIB2          --            2.32087       +2.3209       --%
LDC           --            0.227575      +0.22758      --%
LRS1_SERIES   --            0.222075      +0.22207      --%
LRS2_SERIES   --            0.22357       +0.22357      --%
LRS3_SERIES   --            0.2168        +0.2168       --%

Resistance [--]
Name          Design        Extracted     AbsDiff       PercDiff
RS1           0.61          0.689351      +0.079351     +13.008%
RS2           0.61          0.688966      +0.078966     +12.945%
RS3           0.58          0.618109      +0.038109     +6.5706%
RIB1          2.61          2.64423       +0.03423      +1.3115%
RIB2          4.11          4.11876       +0.0087644    +0.21324%

Junction       Critical current [--]
Name          Design        Extracted     AbsDiff       PercDiff
J1            1.8           1.8           -0            -0%
J2            1.8           1.8           -0            -0%
J3            2             2.0016        +0.0016       +0.08%
```

Although triangles are used for the mesh, *TetraHenry* models sheet currents at the top and bottom of each conducting layer while respecting layer thickness. A 3D visualisation of the current density, with an arbitary scale when I_{B1} is excited, is shown in Fig. 13.23. As seen here, current density is plotted over layer volume. Height variation on layer M2 is also visible due to non-planarization.

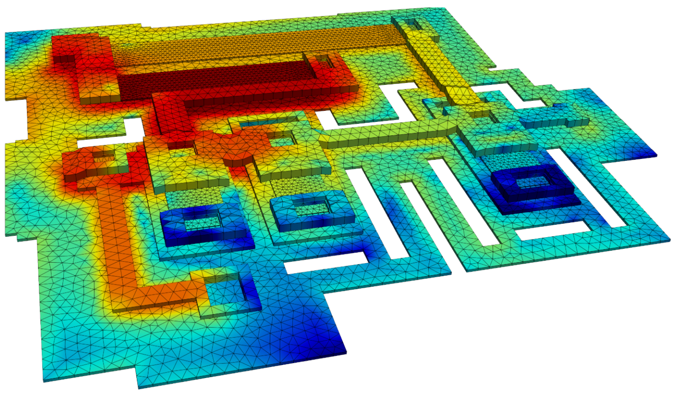


Figure 13.23: Three-dimensional rendering of current distribution in DC-SFQ inductance extraction model with resistors included when layers are modelled with triangles and sheet current flow. Vertical dimension is multiplied by 5 for clarity.

13.7 RSFQ SFQ-DC circuit with CUT layer

Example works with all licence tiers

This example, available in `examples/inductance/rsfq_sfqdc/`, demonstrates:

- Circuit with loops in netlist.
- Cut layer.
- Gaps in layer order.

This example uses the RSFQ SFQ-DC converter developed at Ilmenau University of Technology[3] for the Leibniz-IPHT FLUXONICS foundry [4].

The simplified extraction netlist that disregards resistive branches is shown in Fig. 13.24. The layout, with terminals and labels placed, is shown in Fig. 13.25.

This layout has a short from M1 to M0 for anodization purposes during fabrication. After anodization, the area marked by the object in the CUT layer is ablated away by laser to remove the M1-M0 short.

To extract inductance from the fabrication-ready circuit layout, it is necessary to instruct *InductEx* to remove the short. This is done by setting the CUT layer parameter `FilmType = C` as is shown in Layer Definition File 13.7.2. *InductEx* will now "etch" away all structures underneath objects in the CUT layer.

Also, note that the layer order in `i1k.ldf` has gaps to make room for another isolation layer between I0B and M1 or auxiliary layers above R2. *InductEx* compresses the layer order when it detects gaps in the layer order sequence. The resulting layer order is printed to the output, with the last die layer order indicated with square brackets (`[11 R2]` in this case).

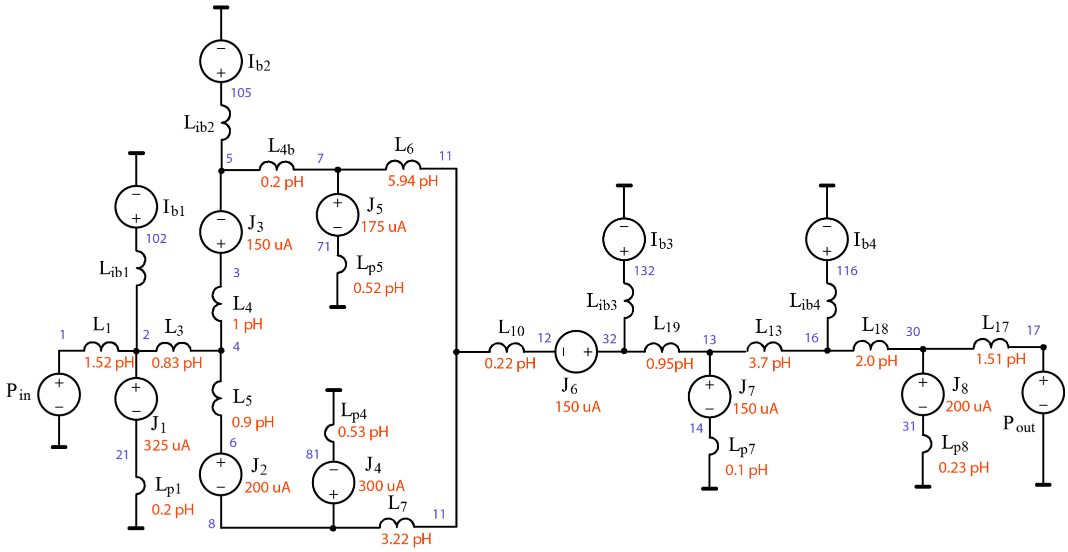


Figure 13.24: *InductEx* netlist for SFQ-DC circuit inductance extraction.

The extraction netlist is shown in Netlist 13.7.1. The bias branch inductors (L_{IB1} - L_{IB4}) have no values specified because these are *don't care* inductors. For the Josephson junctions, the critical current design values are specified. Standard JoSIM or SPICE notation is used, so that parameter values can be entered with the standard suffixes (such as "p", "n", "u", "m", etc.) to

[3] Cell library at https://www.tu-ilmenau.de/en/university/departments/department-of-electrical-engineering-and-information-technology/profile/institutes-and-groups/advanced-electromagnetics-group/research/cryoelectronics/rapid-single-flux-quantum-cell-library/cells

[4] FLUXONICS foundry design rules at: https://www.fluxonics.de/fluxonics-foundry/

denote magnitude.

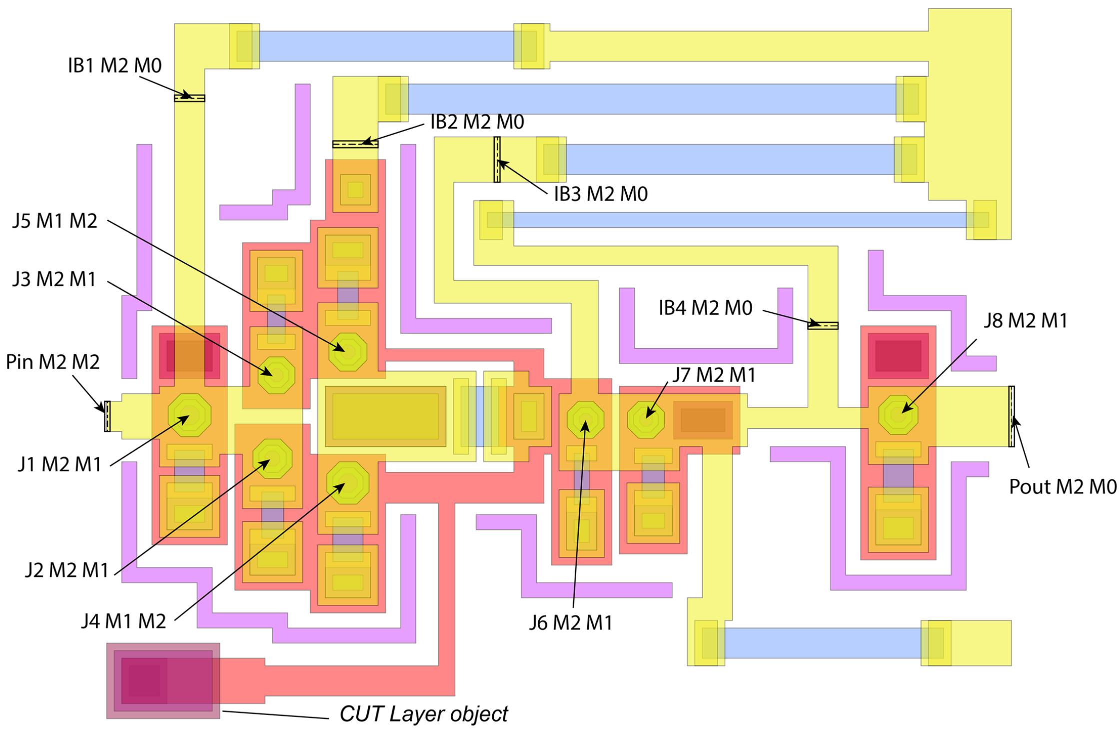


Figure 13.25: Layout of SFQ-DC converter with terminals and port labels added.

■ Netlist 13.7.1 — sfqdc.cir.

```
* Title: Netlist for SFQ-DC
//--- Inductors ---
L1      1    2    1.52p
L3      2    4    0.83p
L4      4    3    0.84p
L5      4    6    0.9p
L4b     5    7    0.2p
L6      7    11   5.94p
L7      8    11   3.22p
L10     11   12   0.22p
L19     32   13   0.95p
L13     13   16   3.7p
L18     16   30   2.0p
L17     30   17   1.51p
Lp1     21   0    0.2p
Lp4     81   0    0.53p
Lp5     71   0    0.52p
Lp7     14   0    0.1p
Lp8     31   0    0.23p
LIB1    102  2
LIB2    105  5
LIB3    132  32
LIB4    116  16
//----- Ports -----
Pin      1    0
J1       2    21   325E-6
J2       6    8    0.000200
J3       3    5    0.000150
J4       8    81   0.000300
J5       7    71   0.175m
J6       32   12   150u
J7       13   14   150u
J8       30   31   200u
IB1      102  0
IB2      105  0
IB3      132  0
IB4      116  0
Pout     17   0
.end
```

Layer Definition File 13.7.2 — i1k.ldf.

```
$Parameters
  Units              =  1e-6
  SegmentSize        =  2.5
  AbsMin             =  0.025
  GPOverhang         =  2.5
  ProcessHasGroundPlane = TRUE
  Relief             =  TRUE
  GPLayer            =  17
  TermLayer          =  19
  TextLayer          =  18
  Lambda             =  0.09
  Sigma              =  10
  TerminalInRange =  1.0
$End
* ----------------
*   LAYERS
* ------ M0 ------
$Layer
  Number     =  17
  Name       =  M0
  Thickness  =  0.2
  Lambda     =  0.09
  Order      =  0
  Mask       = -1
  Filmtype   =  S
  HFilaments =  2
$End
* ------ I0A ------
$Layer
  Number     =  2
  Name       =  I0A
  Thickness  =  0.05
  Order      =  1
  Mask       = -1
  Filmtype   =  I
$End
* ------ I0B ------
$Layer
  Number     =  3
  Name       =  I0B
  Thickness  =  0.2
  Order      =  2
  Mask       = -1
  Filmtype   =  I
$End
* ------ M1 ------
$Layer
  Number     =  5
  Name       =  M1
  Thickness  =  0.25
  Lambda     =  0.09
  Order      =  4
  Mask       =  1
  Filmtype   =  S
  HFilaments =  3
$End
* ------ T1 ------
$Layer
  Number     =  6
  Name       =  T1
  Thickness  =  0.09
  Order      =  5
  Mask       =  0
  Filmtype   =  A
$End
* ------ I1A ------
$Layer
  Number     =  7
  Name       =  I1A
  Thickness  =  0.07
  Order      =  6
  Mask       = -1
  IDensity   =  10e-6
  Filmtype   =  I
$End
* ------ CUT ------
$Layer
  Number     =  8
  Name       =  CUT
  Order      =  7
  Mask       =  0
  Filmtype   =  C
$End
* ------ I1B ------
* I1B
$Layer
  Number     =  9
  Name       =  I1B
  Thickness  =  0.15
  Order      =  8
  Mask       = -1
  Filmtype   =  I
$End
* ------ R1 ------
$Layer
  Number     =  10
  Name       =  R1
  Bias       =  0
  Thickness  =  0.08
  Sigma      =  12.5
  Order      =  9
  Mask       =  1
  Filmtype   =  N
$End
* ------ I2 ------
$Layer
  Number     =  11
  Name       =  I2
  Thickness  =  0.15
  Order      =  10
  Mask       = -1
  Filmtype   =  I
$End
* ------ M2 ------
* M2
$Layer
  Number     =  12
  Name       =  M2
  Thickness  =  0.35
  Lambda     =  0.09
  Order      =  11
  Mask       =  1
  Filmtype   =  S
  HFilaments =  3
$End
* ------ R2 ------
$Layer
  Number     =     13
  Name       =     R2
  Thickness  =     0.5
  Sigma      =     10
  Order      =     12
  Mask       =     1
  Filmtype   =     N
$End
* ----- TERM -----
$Layer
  Number     =  19
  Name       =  TERM
  Order      =  16
  Mask       = -4
$End
```

Execute *InductEx* with:

```
inductex sfqdc.gds -l i1k.ldf -i sfqdc.inp -fh
```

The output shows all the extracted inductances in Henry, and all the critical current values in Ampere.

Command Prompt

```
Collapsed layout order to sequential set.
(0 M0) (1 I0A) (2 I0B) (3 M1) (4 T1) (5 I1A) (6 CUT) (7 I1B) (8 R1) (9 I2) (10 M2) [11 SM]
 (12 TERM) (13 )
...
 Mesh info:
+============+===============+===============+=========================+
| #Filaments | X-width range | Y-width range | Worst L/W ratios        |
+------------+---------------+---------------+-------------------------+
| 39405      | 0.15 - 2.5    | 0.4 - 2.5     | min = 0.07029; max = 50 |
+============+===============+===============+=========================+

SVD solution information (A*x = b):
 - Unknowns = 21; rank = 21.
 - Condition number:                  33.31
 - Max error (Max(|A*x - b|)):        0.0747954
 - Relative error (||A*x - b||/||b||): 2.65787%

Inductance [H]
Name       Design        Extracted     AbsDiff       PercDiff
L1         1.52E-12      1.5626E-12    +4.2601E-14   +2.8027%
L3         8.3E-13       7.81451E-13   -4.8549E-14   -5.8493%
L4         8.4E-13       9.3378E-13    +9.378E-14    +11.164%
L5         9E-13         9.27896E-13   +2.7896E-14   +3.0995%
L4B        2E-13         1.75709E-13   -2.4291E-14   -12.146%
L6         5.94E-12      6.03977E-12   +9.9771E-14   +1.6796%
L7         3.22E-12      3.24784E-12   +2.7843E-14   +0.86469%
L10        2.2E-13       2.44951E-13   +2.4951E-14   +11.341%
L19        9.5E-13       9.68572E-13   +1.8572E-14   +1.955%
L13        3.7E-12       4.09923E-12   +3.9923E-13   +10.79%
L18        2E-12         2.16425E-12   +1.6425E-13   +8.2126%
L17        1.51E-12      1.60322E-12   +9.3215E-14   +6.1732%
LP1        2E-13         1.85156E-13   -1.4844E-14   -7.422%
LP4        5.3E-13       4.78442E-13   -5.1558E-14   -9.7279%
LP5        5.2E-13       5.02045E-13   -1.7955E-14   -3.4529%
LP7        1E-13         4.73874E-14   -5.2613E-14   -52.613%
LP8        2.3E-13       2.02073E-13   -2.7927E-14   -12.142%
LIB1       --            7.34796E-12   +7.348E-12    --%
LIB2       --            1.79278E-12   +1.7928E-12   --%
LIB3       --            1.34255E-11   +1.3426E-11   --%
LIB4       --            2.22421E-12   +2.2242E-12   --%

Junction   Critical current [A]
Name       Design        Extracted     AbsDiff       PercDiff
J1         0.000325      0.0003247     -3E-07        -0.092308%
J2         0.0002        0.0002009     +9E-07        +0.45%
J3         0.00015       0.0001511     +1.1E-06      +0.73333%
J4         0.0003        0.0002999     -1E-07        -0.033333%
J5         0.000175      0.0001759     +9E-07        +0.51429%
J6         0.00015       0.0001511     +1.1E-06      +0.73333%
J7         0.00015       0.0001511     +1.1E-06      +0.73333%
J8         0.0002        0.0002009     +9E-07        +0.45%
```

A three-dimensional rendering of the meshed extraction model is shown in Fig. 13.26. It can be seen that the anodization bridge has been cut out. Tight wrapping of the ground plane up to the `GPOverHang` distance can also be seen.

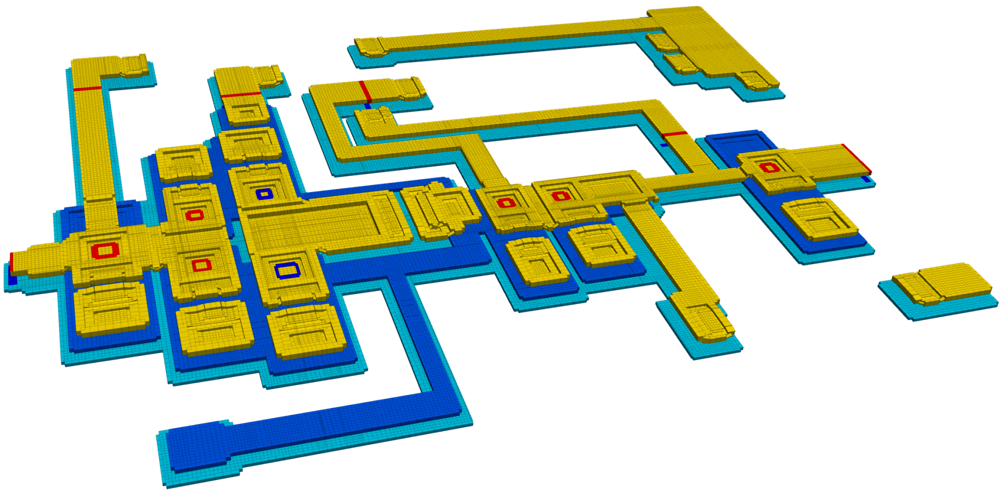


Figure 13.26: Three-dimensional rendering of SFQ-DC inductance extraction model.

13.8 AQFP buffer circuit with mutual inductance

Example works with all licence tiers

This example, available in `examples/coupling/aqfp_buffer/`, demonstrates:

- Mutual inductance.
- Use of the `M` and `k` elements.
- Resistive layer setup where there are different via drawing layers for the same isolation.
- Ground plane not cropped.

This example demonstrates mutual inductance calculation in a highly coupled Adiabatic Quantum-Flux-Parametron (AQFP) buffer cell as presented in [17]. The layout, with ports marked for extraction, is shown in Fig. 13.27. The extraction netlist, with only some of the mutual coupling, is shown in Fig. 13.28.

The circuit netlist contains definitions for the mutual inductance or coupling. *InductEx* accepts both the `M` element for mutual inductance (this does not conform to SPICE syntax, because the symbol is reserved for MOSFET transistors in SPICE), or the JoSIM/SPICE compatible `k` element for coupling factor.

The layer definition file, `process.ldf`, is shown in Layer Definition File 13.8.2. This process description only loosely resembles the AIST SDP 2.5 kA/cm^2 process, and is used here purely as a demonstration example.

Netlist 13.8.1 — aqfpbuffer.cir.

```
* Title: Netlist - AQFP buffer
// -- Inductors --
Lin     1   2
L1      2   3    1.6
L2      4   2    1.6
Lp1     5   0
Lp2     6   0
Ls1     11  0
Ls2     12  0
Lq      2   0
Lout    0   7
Lx      8   10   8
Ld      18  20
// - Coupling/mutual ind. -
Kout    Lq  Lout
* Can specify design value of k
Kx1     Lx  L1   -0.15
* Or specify design value of M
Mx2     Lx  L2   -0.6
Kd1     Ld  L1
Kd2     Ld  L2
Kxq     Lx  Lq
Kdq     Ld  Lq
Kdx     Ld  Lx
Kxout   Lx  Lout
Kdout   Ld  Lout
// ---- Ports ----
Pin     1   0
Pout    7   0
Pj1     3   5
Pj2     4   6
Ps1     3   11   50 // 50 uA
Ps2     4   12   50 // 50 uA
Px1     8   0
Px2     10  0
Pd1     18  0
Pd2     20  0
.end
```

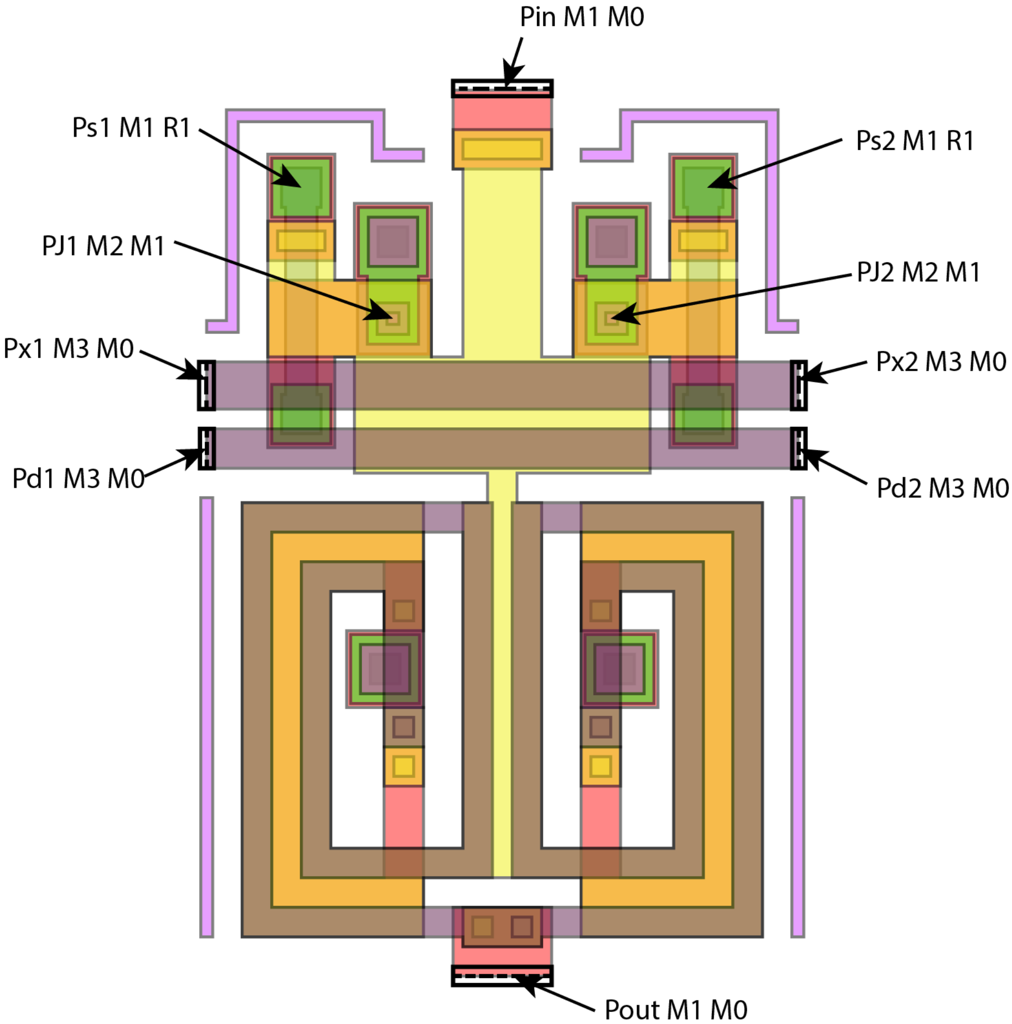


Figure 13.27: Layout of AQFP buffer cell with ports labelled for extraction.

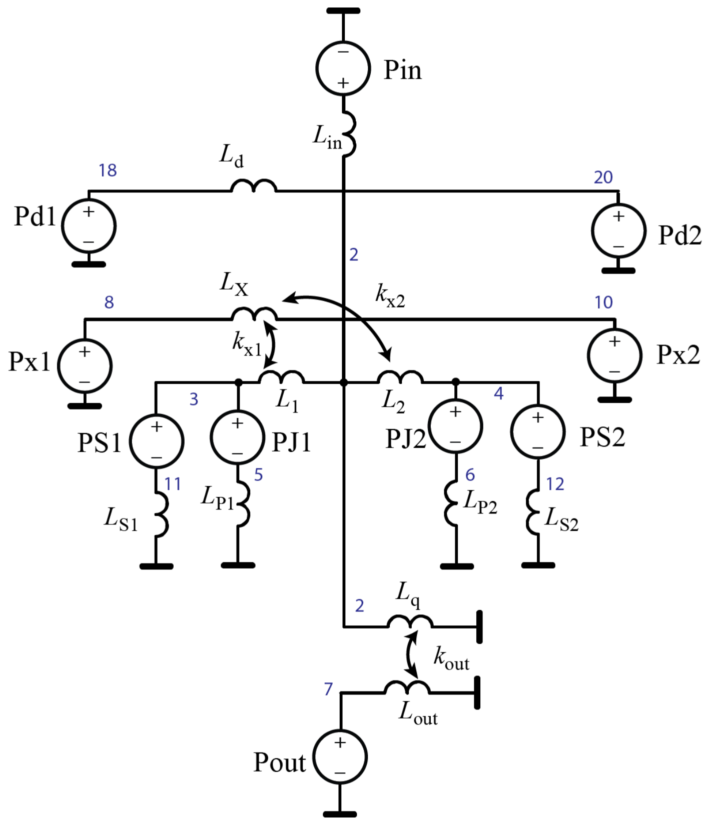


Figure 13.28: *InductEx* netlist for AQFP buffer circuit inductance extraction.

■ **Layer Definition File 13.8.2 — process.ldf.**

```
$Parameters
  Units        =  1e-6
  AbsMin       =  0.01
  SegmentSize =  0.75
  GPOverhang  =  5
  CropGP       =  FALSE
  ProcessHasGroundPlane = TRUE
  GPLayer      =  1
  TermLayer    =  15
  TextLayer    =  64
  UnitL        =  1E-12
  UnitI        =  1E-6
$End
* ----- M0 -----
$Layer
  Number      =  1
  Name        =  M0
  Thickness   =  0.3
  Lambda      =  0.08
  Order       =  0
  Mask        = -1
  Filmtype    =  S
  HFilaments =  3
$End
* ----- I0 -----
$Layer
  Number      =  2
  Name        =  I0
  Thickness   =  0.25
  Order       =  1
  Mask        = -1
  Filmtype    =  I
$End
* ----- R1 -----
$Layer
  Number      =  3
  Name        =  R1
  Thickness   =  0.1
  Sigma       =  10
  Order       =  2
  Mask        =  1
  Filmtype    =  R
  ViaBypass   =  TRUE
$End
* ----- RC -----
$Layer
  Number      =  9
  Name        =  RC
  Thickness   =  0.2
  Order       =  3
  Mask        = -1
  Filmtype    =  I
  LayerADD    =  2
// Add I0 to create vias to M0
$End
* ----- M1 -----
$Layer
  Number      =  4
  Name        =  M1
  Thickness   =  0.3
  Lambda      =  0.08
  Order       =  4
  Mask        =  1
  Filmtype    =  S
  HFilaments =  3
$End
* ----- JP -----
$Layer
  Number      =  5
  Name        =  JP
  Thickness   =  0.01
  Order       =  5
  Mask        =  0
  Filmtype    =  A
$End
* ----- JJ -----
$Layer
  Number      =  6
  Name        =  JJ
  Bias        = -0.1
  Thickness   =  0.15
  Order       =  6
  Mask        =  0
  Filmtype    =  A
  IDensity    =  2.5e-5
$End
* ----- JC -----
$Layer
  Number      =  10
  Name        =  JC
  Thickness   =  0.01
  Order       =  7
  Mask        =  0
  Filmtype    =  A
$End
* ----- I1 -----
$Layer
  Number      =  7
  Name        =  I1
  Thickness   =  0.4
  Order       =  8
  Mask        = -1
  Filmtype    =  I
  LayerADD    =  10
$End
* ----- M2 -----
$Layer
  Number      =  8
  Name        =  M2
  Thickness   =  0.4
  Lambda      =  0.08
  Order       =  9
  Mask        =  1
  Filmtype    =  S
  HFilaments =  3
$End
* ----- I2 -----
$Layer
  Number      =  11
  Name        =  I2
  Thickness   =  0.480
  Order       =  10
  Mask        = -1
  Filmtype    =  I
$End
* ----- M3 -----
$Layer
  Number      =  12
  Name        =  M3
  Bias        = -0.06
  Thickness   =  0.5
  Lambda      =  0.08
  Order       =  11
  Mask        =  1
  Filmtype    =  S
  HFilaments =  3
$End
* ---- TERM ----
$Layer
  Number      =  15
  Name        =  TERM
  Order       =  12
  Mask        = -4
$End
```

Execute *InductEx* with:

```
inductex aqfpbuffer.gds -l process.ldf -i aqfpbuffer.inp -fh
```

Note that all the inductance values are scaled to pH (`UnitL = 1E-12`) and all the current values are scaled to µA. *InductEx* lists the mutual inductance results in terms of M, which is also scaled to pH in this example. The design value for M_{x2}, -0.6, is listed directly. For M_{x1} the design value of the coupling factor was specified in `aqfpbuffer.cir`, so that the design value printed in the output is calculated from the coupling factor, L_1 and L_2 as -0.53666.

A rendering of the meshed structure is shown in Fig. 13.29. The vertical dimension has been stretched to accentuate height.

Command Prompt

```
Inductance [pH]
Name      Design        Extracted     AbsDiff       PercDiff
LIN       --            2.45691       +2.4569       --%
L1        1.6           1.59583       -0.0041693    -0.26058%
L2        1.6           1.59642       -0.0035804    -0.22378%
LP1       --            0.0229019     +0.022902     --%
LP2       --            0.0229302     +0.02293      --%
LS1       --            2.53607       +2.5361       --%
LS2       --            2.53628       +2.5363       --%
LQ        --            7.34288       +7.3429       --%
LOUT      --            28.7892       +28.789       --%
LX        8             9.42528       +1.4253       +17.816%
LD        --            10.1751       +10.175       --%

Resistance [Ohm]
Name      Design        Extracted     AbsDiff       PercDiff
RS1       --            6.39957       +6.3996       --%
RS2       --            6.39969       +6.3997       --%

Mutual Inductance [pH]                                             Coupling factor
Name      Design        Extracted     AbsDiff       PercDiff              k
MOUT      ---           -7.4366       -7.4366       --%           -0.51148
MX1       -0.53666      -0.68641      +0.14975      +27.905%      -0.17699
MX2       -0.6          -0.68638      +0.086383     +14.397%      -0.17695
MD1       ---           -0.51226      -0.51226      --%           -0.12712
MD2       ---           -0.51197      -0.51197      --%           -0.12703
MXQ       ---           -0.00037743   -0.00037743   --%           -4.537E-5
MDQ       ---           -0.00033958   -0.00033958   --%           -3.929E-5
MDX       ---           +2.7572       +2.7572       --%           +0.28155
MXOUT     ---           +0.00070471   +0.00070471   --%           +4.278E-5
MDOUT     ---           +0.0019185    +0.0019185    --%           +1.121E-4

Junction   Critical current [uA]
Name      Design        Extracted     AbsDiff       PercDiff
PJ1       50            49            -1            -2%
PJ2       50            49            -1            -2%
```

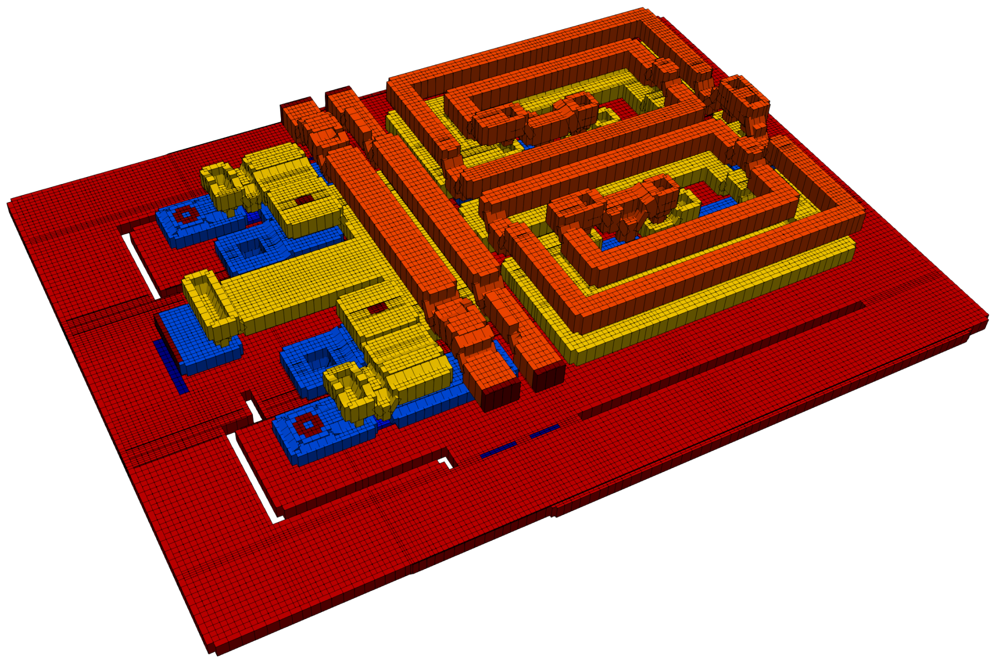


Figure 13.29: Three-dimensional rendering of AQFP buffer inductance extraction model.

13.9 Circuit in planarized multilayer process with ground and skyplane

Example works with all licence tiers

This example, available in `examples/inductance/rsfq_jtl/`, demonstrates:

- Tetrahedral meshes.
- Layer-specific segment size.
- Port terminals connected to multiple layers.
- Planarization.
- Skyplane connection.
- The SET command.

This example demonstrates inductance extraction for a circuit layout in a multilayer process where a skyplane is utilised. The circuit is a standard two-junction Josephson transmission line. The schematic of the extraction netlist is shown in Fig. 13.31.

The circuit netlist contains a control block with two `SET` statements. The first is used to set `SegmentSize` for layer M7 to 0.75 µm, while the second sets the London penetration depth for layer M4 to 85 nm. These commands override the values in the layer definition file. This shows how parameters can be manipulated without altering the layer definition file.

The layer definition file, `multilayer.ldf`, is shown in Layer Definition File 13.9.2. This process description only loosely resembles the MIT Lincoln Laboratories SFQ5ee 10 kA/cm^2 process [18], and is used here purely as a demonstration example. In this example, `SegmentSize` for layer M6 is defined directly in the layer description. This overrides the global value defined in the parameter block.

The use of a skyplane (M7) along with a ground plane (M4) provides a second current return path, specifically because the two layers are connected at regular intervals. This requires careful modelling. For inductors that form part of branches connecting to the edges of the layout cell (L_1, L_4 and L_{dc}), placement in a larger chip environment will cause return current for a signal on these inductors to flow below the conductors in both layers M4 and M7. **The ports at the input to these inductors must thus provide contact to layers M4 and M7**.

This is achieved by defining the ports P_{in}, P_{out} and P_{dc} to have the positive terminal on layer M6 and the negative terminal on both layers M4 and M7. Port labelling is shown in Fig. 13.30.

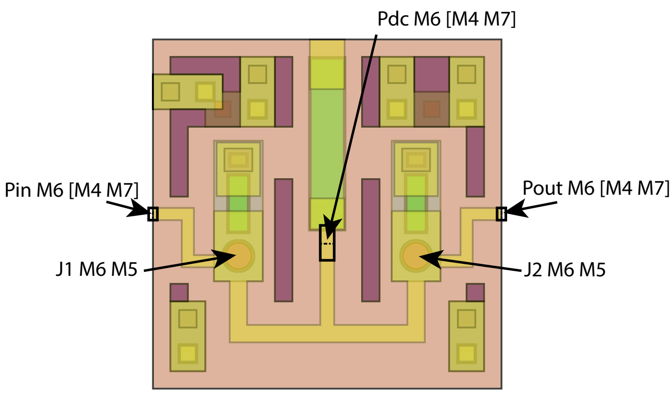


Figure 13.30: Layout of JTL in multilayer process.

When cuboid filament meshes are used, planarization has to be specified. In `mulitlayer.ldf`, each isolation layer has the parameter `PlanarModel = 1`. This enforces planarization of the conducting layer just above the corresponding isolation layer. To use a cuboid filament mesh, execute *InductEx* with:

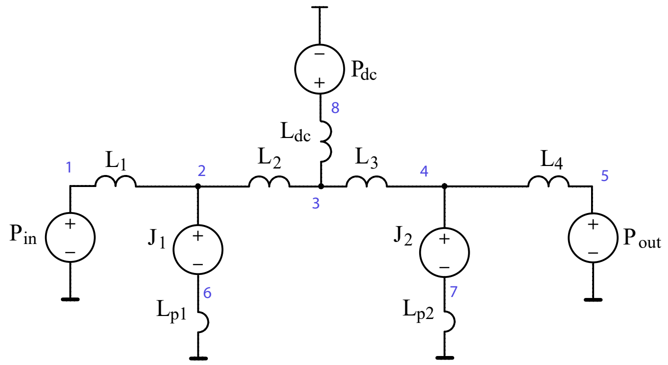


Figure 13.31: *InductEx* netlist for AQFP buffer circuit inductance extraction.

```
inductex jtl.gds -l multilayer.ldf -i jtl.inp -fh
```

Example works with Professional and Super licence tiers

To use a tetrahedral mesh model, execute *InductEx* with:

```
inductex jtl.gds -l multilayer.ldf -i jtl.geo -th -zoff
```

The `-zoff` parameter switches off the default modelling option of current sheets (planes), allowing *InductEx* to build volumetric tetrahedral meshes.

Note that all the inductance values are scaled to pH (`UnitL = 1E-12`).

Netlist 13.9.1 — jtl.cir.

```
* Title: Netlist for JTL
.control
  SET layer M7 segmentsize = 0.75
  SET layer M4 lambda = 0.085
.endc
L1    1  2  2
L2    2  3  2
L3    3  4  2
L4    4  5  2
Lp1   6  0  0.2
Lp2   7  0  0.2
Ldc   8  3
Pin   1  0
Pout  5  0
Pdc   8  0
J1    2  6
J2    4  7
.end
```

■ Layer Definition File 13.9.2 — multilayer.ldf.

```
$Parameters
  Units          =  1e-6
  SegmentSize    =  0.5
  AbsMin         =  0.025
  CropGP         =  FALSE
  GPLayer        =  40
  TermLayer      =  19
  TextLayer      =  18
  TerminalInRange =  1.0
  DecimationDistance = 0.05
  UnitL          =  1E-12
$End
$Layer  // ---- M0 ----
  Number      =     1
  Name        =     M0
  Bias        =     0
  Thickness   =     0.200
  Lambda      =     0.091
  Order       =     0
  Mask        =     1
  Filmtype    =     S
  HFilaments  =     2
$End
$Layer ---- I0 ----
  Number      =     2
  Name        =     I0
  Thickness   =     0.2
  Order       =     3
  Mask        =    -1
  Filmtype    =     I
  PlanarModel =     1
$End
$Layer ---- M1 ----
  Number      =     10
  Name        =     M1
  Thickness   =     0.2
  Lambda      =     0.09
  Order       =     4
  Mask        =     1
  Filmtype    =     S
  HFilaments  =     2
$End
$Layer ---- I1 ----
  Number      =     11
  Name        =     I1
  Thickness   =     0.2
  Order       =     5
  Mask        =    -1
  Filmtype    =     I
  PlanarModel =     1
$End
$Layer ---- M2 ----
  Number      =     20
  Name        =     M2
  Thickness   =     0.2
  Lambda      =     0.09
  Order       =     6
  Mask        =     1
  Filmtype    =     S
  HFilaments  =     2
$End
$Layer ---- I2 ----
  Number      =     21
  Name        =     I2
  Thickness   =     0.2
  Order       =     7
  Mask        =    -1
  Filmtype    =     I
  PlanarModel =     1
$End
$Layer ---- M3 ----
  Number      =     30
  Name        =     M3
  Thickness   =     0.2
  Lambda      =     0.09
  Order       =     8
  Mask        =     1
  Filmtype    =     S
  HFilaments  =     2
$End
$Layer ---- I3 ----
  Number      =     31
  Name        =     I3
  Thickness   =     0.2
  Order       =     9
  Mask        =    -1
  Filmtype    =     I
  PlanarModel =     1
$End
$Layer ---- M4 ----
  Number      =     40
  Name        =     M4
  Thickness   =     0.2
  Lambda      =     0.09
  Order       =     10
  Mask        =     1
  Filmtype    =     S
  HFilaments  =     2
$End
$Layer ---- I4 ----
  Number      =     41
  Name        =     I4
  Thickness   =     0.2
  Order       =     11
  Mask        =    -1
  Filmtype    =     I
  PlanarModel =     1
$End
$Layer ---- M5 ----
  Number      =    50
  Name        =    M5
  Thickness   =    0.2
  Lambda      =    0.09
  Order       =    12
  Mask        =    1
  Filmtype    =    S
  HFilaments =    1
$End
$Layer ---- J5 ----
  Number      =     51
  Name        =     J5
  Bias        =    -0.02
  Thickness   =     0.001
  Order       =     13
  Mask        =     0
  Filmtype    =     A
  IDensity   =      100e-6
$End
$Layer ---- R5 ----
  Number      =     52
  Name        =     R5
  Thickness   =     0.001
  Order       =     14
  Mask        =     1
  Filmtype    =     N
$End
$Layer ---- C5J ----
  Number      =     55
  Name        =     C5J
  Thickness   =     0.001
  Order       =     15
  Mask        =     0
  Filmtype    =     A
$End
$Layer ---- I5 ----
  Number      =     54
  Name        =     I5
  Thickness   =     0.3
  Order       =     16
  Mask        =    -1
  Filmtype    =     I
  PlanarModel =     1
  LayerADD    =     51
$End
$Layer ---- M6 ----
  Number      =    60
  Name        =    M6
  Thickness   =    0.2
  Lambda      =    0.09
  Order       =    17
  Mask        =    1
  Filmtype    =    S
  SegmentSize =    0.35
  HFilaments =    2
$End
$Layer ---- I6 ----
  Number      =     61
  Name        =     I6
  Thickness   =     0.2
  Order       =     18
  Mask        =    -1
  Filmtype    =     I
  PlanarModel =     1
$End
$Layer ---- M7 ----
  Number      =    70
  Name        =    M7
  Thickness   =    0.2
  Lambda      =    0.09
  Order       =    19
  Mask        =    1
  Filmtype    =    S
  HFilaments =    2
$End
```

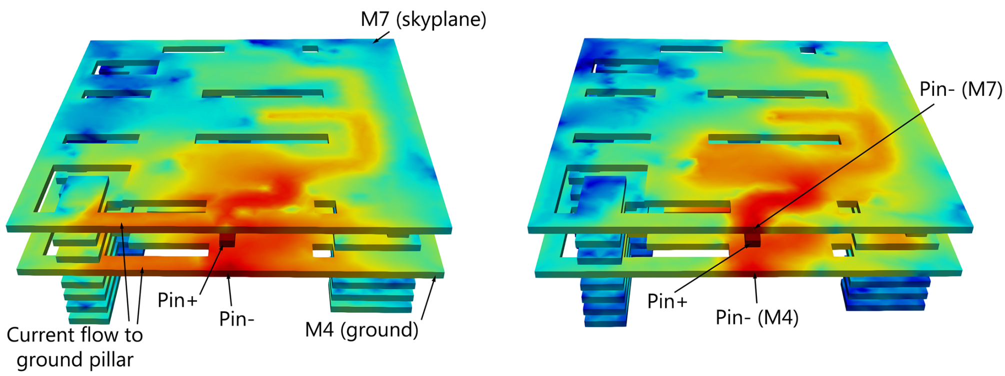


Figure 13.32: Three-dimensional rendering of JTL in multilayer process with current distribution when Port P_{in} is excited. If input/output ports are connected with negative terminals only on the lower ground plane M4, return current in the upper skyplane M7 is diverted to the nearest M4-M7 via stacks (this is shown on the left) with the result that calculated inductance is too high. A more accurate model is shown on the right, where input/output ports have the negative terminal connected to both M4 and M7.

Command Prompt

```
Tetrahedral mesh:
Inductance [pH]
Name       Design         Extracted     AbsDiff       PercDiff
L1         2              1.87282       -0.12718      -6.3588%
L2         2              2.05219       +0.052192     +2.6096%
L3         2              2.05359       +0.053592     +2.6796%
L4         2              1.87228       -0.12772      -6.3859%
LP1        0.2            0.449333      +0.24933      +124.67%
LP2        0.2            0.450814      +0.25081      +125.41%
LDC        --             1.40936       +1.4094       --%

Cuboid filament mesh:
Inductance [pH]
Name       Design         Extracted     AbsDiff       PercDiff
L1         2              1.84408       -0.15592      -7.796%
L2         2              2.0379        +0.037899     +1.895%
L3         2              2.04364       +0.043645     +2.1822%
L4         2              1.84327       -0.15673      -7.8366%
LP1        0.2            0.429973      +0.22997      +114.99%
LP2        0.2            0.429967      +0.22997      +114.98%
LDC        --             1.4069        +1.4069       --%
```

Modelling mistakes

If the port labels are defined as:

```
Pin M6 M4
Pout M6 M4
Pdc M6 M4
```

the inductances are calculated too high, as shown below. The reason is visible in Fig. 13.32. The rendering on the left shows the current distribution in the layout structure when Port P_{in} is excited. Return current in the skyplane (M7) diverts to the nearest M4-M7 via stacks to reach the negative terminal in M4. The result is a larger current loop area and inflated inductance results.

With the negative terminals connected to both layers M4 and M7, the current distribution is correct as shown in the rendering on the right in Fig. 13.32.

Command Prompt

```
Inductance [pH]
Name       Design         Extracted     AbsDiff       PercDiff
L1         2              2.27705       +0.27705      +13.852%
L2         2              2.14615       +0.14615      +7.3074%
L3         2              2.14508       +0.14508      +7.2542%
L4         2              2.27115       +0.27115      +13.557%
LP1        0.2            0.351051      +0.15105      +75.526%
LP2        0.2            0.351576      +0.15158      +75.788%
LDC        --             1.41085       +1.4108       --%
```

13.10 Weak coupling between coplanar input line an qubit structure

Example works with Professional and Super licence tiers

This example, available in `examples/coupling/qubit/`, demonstrates:

- High-fidelity meshing of edges to improve weak mutual inductance results.

This example demonstrates inductance extraction of the mutual inductance between a coplanar input line and a SQUID structure in a hole in a monolayer process such as those used for qubit structures.

Models such as this give better results when a substantial part of the ground plane is included, and when the input lne is sufficiently long to prevent currents on the ground plane edge from interfering with the coupling during extraction. Return current from the input line flows in the ground plane right next to the input line, so that the input port's negative terminals should be short and not span the entire width of the ground plane. This is demonstrated in the layout file available in the example directory.

Due to the size of the ground plane, the segment size is made large. The fidelity of the solution is then increased by using edge segments as wide as the penetration depth for all superconductor objects. *InductEx* adds such edge segments when the `-A` switch is used.

The layer definition file, `process.ldf`, is shown in Layer Definition File 13.10.2..

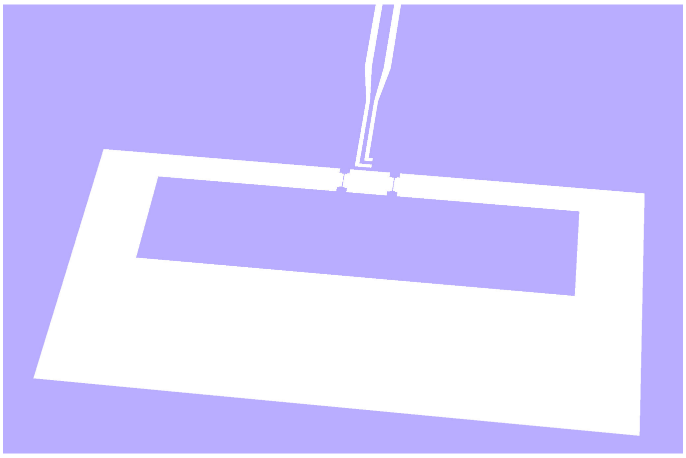


Figure 13.33: A coplanar input line coupled to a SQUID structure in a hole.

To use a triangular filament mesh – the most economical mesh for large thin-film structures, execute *InductEx* with:

```
inductex squidcoupling.gds -l process.ldf -i mesh.geo -th -a
```

■ Netlist 13.10.1 — squidcoupling.cir.

```
Lloop   1   0   100E-11
Linput  2   0   2E-11
K1   Lloop  Linput
* Ports
Psquid    1   0
Pin       2   0
.end
```

■ **Layer Definition File 13.10.2 — process.ldf.**

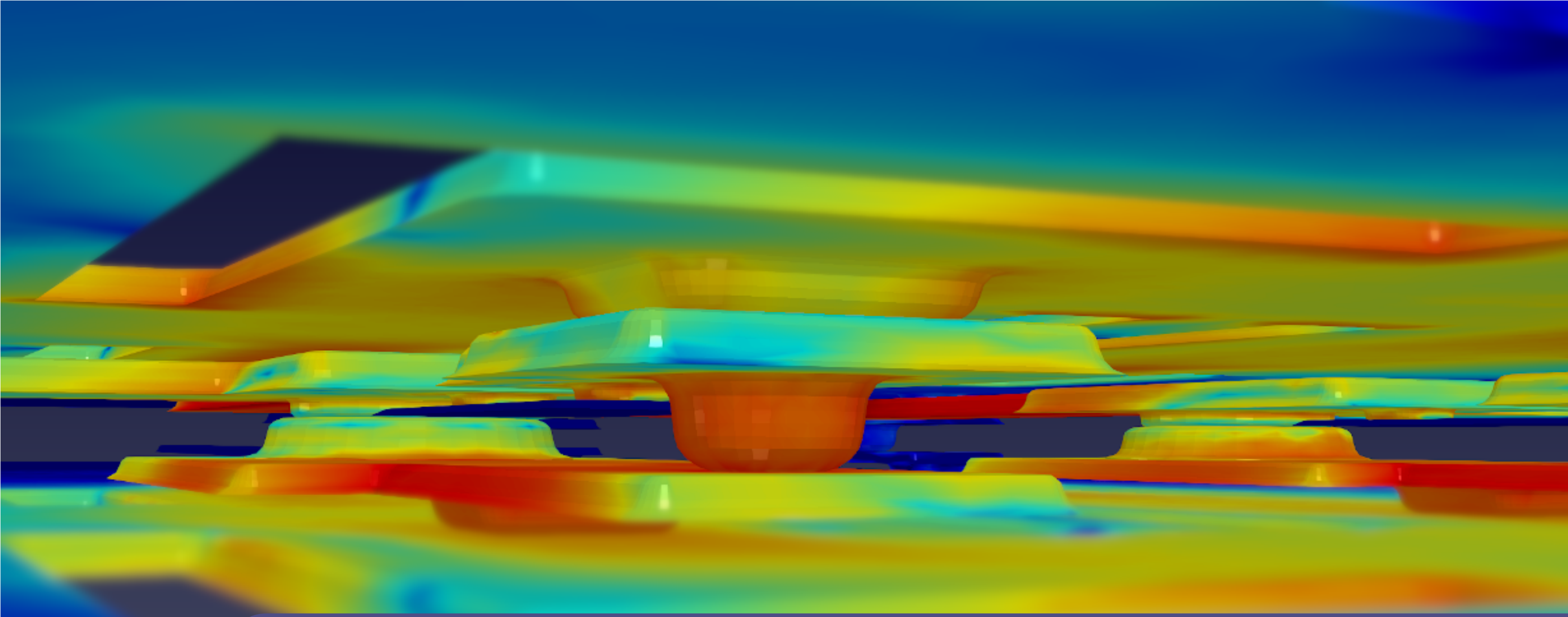
```
$Parameters
  Units              =  1e-6
  SegmentSize        =  15
  AbsMin             =  0.01
  ProcessHasGroundPlane = FALSE
  ExtractBoundLayer = 200
  TermLayer          =  19
  TextLayer          =  20
$End
$Layer
  Number      =   2
  Name        =   NB
  Thickness   =   0.2
  Lambda      =   0.09
  Order       =   0
  Mask        =   1
  Filmtype    =   S
$End
```

Command Prompt

```
Inductance [H]
Name      Design       Extracted     AbsDiff       PercDiff
LLOOP     1E-09        1.08843E-10   -8.9116E-10   -89.116%
LINPUT    2E-11        1.54909E-09   +1.5291E-09   +7645.4%

Mutual Inductance [H]                                             Coupling factor
Name      Design       Extracted     AbsDiff       PercDiff              k
M1        +4E-12       +4.0335E-12   +3.3464E-14   +0.83659%       +9.823E-3
```

Note that the mutual inductance is 4 pH, compared to the SQUID loop inductance of almost 110 pH, and is very sensitive to inaccurate modelling.



14. Capacitance calculation

14.1 Parallel plates

Example works with Professional and Super licence tiers

This easy example, available in `examples/capacitance/parallel_plates/`, demonstrates:
- Capacitance conductor setup.

A set of parallel plates are defined by way of an IXI file. The dimensions are parameterised, and for this example the conductors are resistive and the dielectric is free space.

■ **IXI File 14.1.1 — parallelplates.ixi.**

```
$PARAMSTRING
  -l capprocess.ldf
  -qt
  -zoff
$END
$VARIABLE
  Name   d
  Value  3
$END
$STRUCT
  Name cap
  $POLY
    Layer  m0
    XY
      ( -%d/2 , -%d/2 )
      (  %d/2 , -%d/2 )
      (  %d/2 ,  %d/2 )
      ( -%d/2 ,  %d/2 )
      ( -%d/2 , -%d/2 )
  $END
  $POLY
    Layer  m1
    XY
      ( -%d/2 , -%d/2 )
      (  %d/2 , -%d/2 )
      (  %d/2 ,  %d/2 )
      ( -%d/2 ,  %d/2 )
      ( -%d/2 , -%d/2 )
  $END
  $TEXT
    Text "C1 M0"
    Layer Text
    XY
    (   0, 0)
  $END
  $TEXT
    Text "C2 M1"
    Layer Text
    XY
    (   0, 0)
  $END
$END
```

InductEx models the conductor surfaces, which are meshed and passed to TetraHenry when the `-qt` switch is used. No netlist is required.

It is important to note that the relative permittivity below the lowest layer has to be set to 1 with `EpsilonRBelow = 1` in the process file, because *InductEx* as an integrated circuit extraction tool assumes a silicon substrate with $\varepsilon_r = 11.7$ as default. In free space, it is not necessary to specify `EpsilonRAbove = 1` because that is the default value.

■ **Layer Definition File 14.1.2 — capprocess.ldf.**

```
$Parameters
  Units               =  1e-6
  SegmentSize         =  0.1
  AbsMin              =  0.025
  ProcessHasGroundPlane = FALSE
  EpsilonRBelow       =  1
  EpsilonRAbove       =  1
  GPLayer             =  1
  TermLayer           =  63
  TextLayer           =  64
$End
* ------- M0 -------
$Layer
  Number      =     1
  Name        =     M0
  Thickness   =     0.2
  Order       =     0
  Mask        =     1
  Sigma       =     10
  Filmtype    =     R
$End
* ------- I0 -------
$Layer
  Number      =     2
  Name        =     I0
  Thickness   =     1
  Order       =     1
  Mask        =     -1
  Filmtype    =     I
  EpsilonR    =  1
$End
* ------- M1 -------
$Layer
  Number      =     3
  Name        =     M1
  Thickness   =     0.2
  Order       =     3
  Mask        =     1
  Sigma       =     10
  Filmtype    =     R
$End
```

Execute *InductEx* with:

```
inductex parallelplates.ixi
```

The Maxwell capacitance matrix is calculated, and results printed to the output.

Command Prompt

```
Capacitance results [F]
C1 C1              2.07997E-16
C1 C2             -1.22013E-16
C2 C1             -1.22013E-16
C2 C2              2.07902E-16
```

The capacitance between the plates is $-C_{12} = -C_{21} = 1.22\,\text{pF}$.

14.2 Multiple lines above ground plane with multiple dielectric layers

Example works only with Super licence tier

This example, available in `examples/capacitance/multiple_dielectrics/`, demonstrates:

- Multiple capacitance conductor setup.
- Multiple dielectric layers, including substrate and free space.
- The use of RELIEF = FALSE to disable elevation change, or enforce planarisation.

Two conductors above a ground plane are defined by way of an IXI file. The dimensions are parameterized, and for this example the conductors are superconductive. In this example, a die substrate with relative permittivity of 11.2 is modelled below an isolation layer (such as an oxide) IL with an arbitrary $\varepsilon_r = 5.5$. The ground plane layer M0 is located above this isolation layer, and is separated from conductors in M1 and M2 by isolation layers with different dielectric constants. Finally, an isolation layer I2 with $\varepsilon_r = 4.6$ above M2 separates the structure from free space. The relative permittivity for the die substrate is defined with `EpsilonRBelow`, while that of free space above the integrated circuit is defined with `EpsilonRAbove`.

To enforce planarisation, every isolation layer can be defined with PLANARMODEL = 1. Alternatively, global planarisation can be enforced with by setting the global parameter RELIEF = FALSE as shown in Layer Definition File 14.2.2.

The dielectric boundaries are modelled as surfaces that extend beyond the horizontal layout dimensions by the distance `DielectricOverhang` and are meshed to the size defined with `DielectricMeshSize`, which can be different from the mesh size of any of the conductor layers. A cross-section of the structure described here is shown as rendered from the meshed model in Fig. 14.1.

IXI File 14.2.1 — multilayercap.ixi.

```
$PARAMSTRING
  -l multidielectric.ldf
  -qt
  -zoff
$END
$VARIABLE
  Name    d
  Value   3
$END
$VARIABLE
  Name    w1
  Value   1
$END
$VARIABLE
  Name    w2
  Value   0.5
$END
$VARIABLE
  Name    s
  Value   0.2
$END
$STRUCT
  Name cap
  $POLY
    Layer  m0
    XY
    ( -%d/2 , -%d/2 )
    (  %d/2 , -%d/2 )
    (  %d/2 ,  %d/2 )
    ( -%d/2 ,  %d/2 )
    ( -%d/2 , -%d/2 )
  $END
  $PATH
    Layer  m1
    Width  %w1
    XY
    ( -%d/2 , %s/2+%w1/2 )
    (  %d/2 , %s/2+%w1/2 )
  $END
  $PATH
    Layer  m2
    Width  %w2
    XY
    ( -%d/2 , -%s/2-%w2/2 )
    (  %d/2 , -%s/2-%w2/2 )
  $END
  $TEXT
    Text "Cgnd M0"
    Layer Text
    XY
    (0, 0)
  $END
  $TEXT
    Text "C1 M1"
    Layer Text
    XY
    (0, %s/2+%w1/2)
  $END
  $TEXT
    Text "C2 M2"
    Layer Text
    XY
    (0, -%s/2-%w2/2)
  $END
$END
```

■ **Layer Definition File 14.2.2 — multidielectric.ldf.**

```
$Parameters
  Units              =  1e-6
  SegmentSize        =  0.1
  AbsMin             =  0.025
  ProcessHasGroundPlane = FALSE
  Relief             =  FALSE
  GPLayer            =  1
  TermLayer          =  63
  TextLayer          =  64
  DielectricOverhang = 1.5
  DielectricMeshSize = 0.25
  EpsilonR           =  4.6
  EpsilonRbelow      =  11.2
  EpsilonRabove      =  1
$End
* -- IL - Oxide ---
$Layer
  Number     =  30
  Name       =  IL
  Thickness  =  0.1
  Order      =  0
  Mask       = -1
  Filmtype   =  I
  EpsilonR   =  5.5
$End
* -- M0 - metal ---
$Layer
  Number     =  1
  Name       =  M0
  Thickness  =  0.2
  Order      =  1
  Mask       =  1
  Lambda     =  0.09
  Filmtype   =  S
$End
* I0 - Dielectric
$Layer
  Number     =  2
  Name       =  I0
  Thickness  =  0.2
  Order      =  2
  Mask       = -1
  Filmtype   =  I
  EpsilonR   =  3.9
$End
* -- M1 - metal ---
$Layer
  Number     =  3
  Name       =  M1
  Thickness  =  0.3
  Order      =  3
  Mask       =  1
  Lambda     =  0.09
  Filmtype   =  S
$End
* I1 - Dielectric
$Layer
  Number     =  4
  Name       =  I1
  Thickness  =  0.2
  Order      =  4
  Mask       = -1
  Filmtype   =  I
  EpsilonR   =  4.9
$End
* -- M2 - metal ---
$Layer
  Number     =  5
  Name       =  M2
  Thickness  =  0.4
  Order      =  5
  Mask       =  1
  Lambda     =  0.09
  Filmtype   =  S
$End
* I2 - Dielectric
$Layer
  Number     =  6
  Name       =  I2
  Thickness  =  0.15
  Order      =  6
  Mask       = -1
  Filmtype   =  I
  EpsilonR   =  4.6
$End
```

Execute *InductEx* with:

```
inductex multilayercap.ixi
```

The Maxwell capacitance matrix is calculated, and results printed to the output.

Command Prompt

```
Capacitance results [F]
C1 C1             1.21294E-15
C1 C2            -2.33391E-16
C1 CGND          -8.57379E-16
C2 C1            -2.33391E-16
C2 C2             6.06674E-16
C2 CGND          -2.2302E-16
CGND C1          -8.57379E-16
CGND C2          -2.2302E-16
CGND CGND         1.82694E-15
```

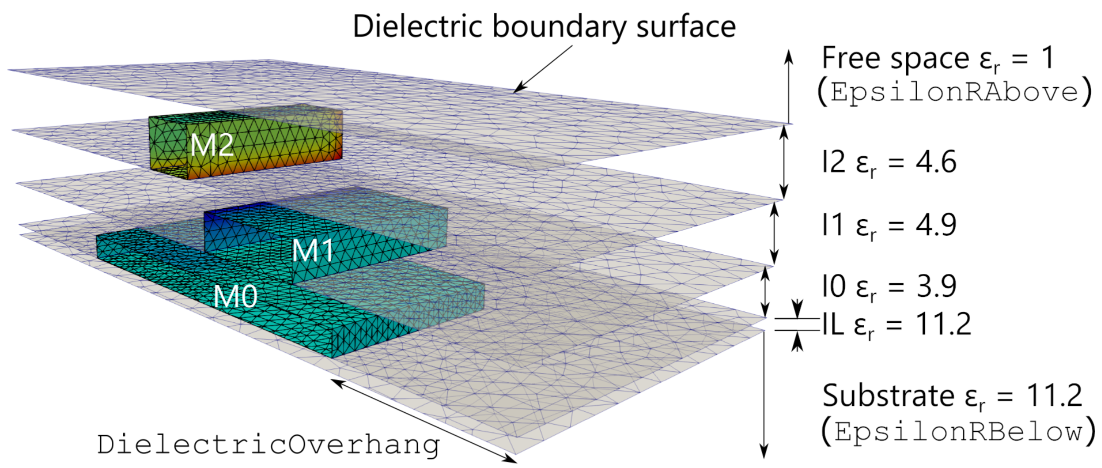
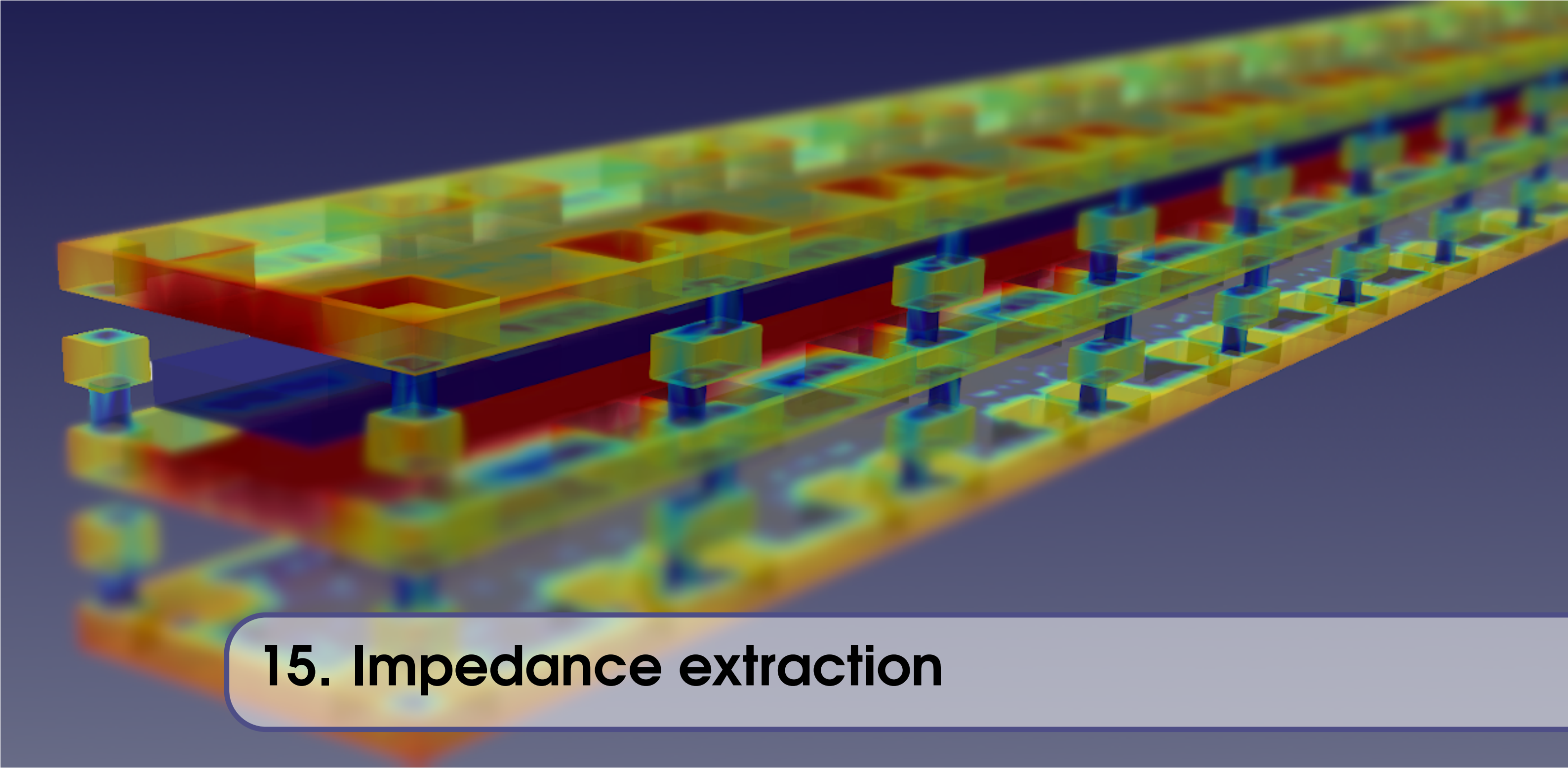


Figure 14.1: Cross-section of capacitance calculation model with multiple conductors and multiple dielectric layers.



15. Impedance extraction

15.1 Characteristic impedance of superconductor microstrip

Example works only with Super licence tier

This example, available in `examples/impedance/microstrip/`, demonstrates:
- Setup of characteristic impedance extraction.
- Dielectric layers.

This example demonstrates characteristic impedance calculation for a superconductor microstrip line over a finite ground plane. The finite ground plane structure sits on top of a dielectric material with relative permittivity of 11. The dielectric layer between the ground plane and the conductor has a relative permittivity of 4.6, while above the conductor there is free space. A cut-through of the meshed calculation model is shown in Fig. 15.1.

In process file 15.1.2, `microstrip.ldf`, the dielectric relative permittivity `EpsilonR` is set to 4.6 for the isolation layer. The global parameter `EpsilonRBelow = 11` defines the substrate below the lowest die layer and `EpsilonRAbove = 1` sets the free space above the last die layer.

■ IXI File 15.1.1 — microstrip.ixi.

```
$PARAMSTRING
  -l microstrip.ldf
  -zoff   // use tetras
  -Z      // find impedance
$END
$VARIABLE
  Name   wl // Line width
  Value  2.5
$END
$VARIABLE
  Name   wg // Ground wrap
  Value  2.0
$END
$VARIABLE
  Name   L  // Line length
  Value  15
$END
$STRUCT
  Name microstrip
  $POLY
    Layer  m1
    XY
    ( 0.0 , -%wl/2 )
    ( %L  , -%wl/2 )
    ( %L  ,  %wl/2 )
    ( 0   ,  %wl/2 )
    ( 0.0 , -%wl/2 )
  $END
  $POLY
    Layer  m0
    XY
    (-%wg    ,-(%wg+%wl/2) )
    ( %wg+%L,-(%wg+%wl/2) )
    ( %wg+%L, (%wg+%wl/2) )
    (-%wg    , (%wg+%wl/2) )
    (-%wg    ,-(%wg+%wl/2) )
  $END
  $POLY
    Layer term
    Text "Z1 m1 m0"
    XY
    ( 0 , -%wl/2 )
    ( 0 ,  %wl/2 )
  $END
  $POLY
    Layer term
    Text "Z2 m1 m0"
    XY
    ( %L , -%wl/2 )
    ( %L ,  %wl/2 )
  $END
$END
```

The dielectric interfaces are created as meshed rectangles that exceeds the furthest limits of the layout structure by `DielectricOverhang`, while the global dielectric surface mesh size is set with `DielectricOverhang`.

■ Layer Definition File 15.1.2 — microstrip.ldf.

```
$Parameters
  Units              =  1e-6
  SegmentSize        =  0.75
  AbsMin             =  0.025
  GPLayer            =  1
  DielectricOverhang = 10
  DielectricMeshSize = 1
  EpsilonRBelow      =  11
  EpsilonRAbove      =  1
  ProcessHasGroundPlane = TRUE
  Relief             =  FALSE
  TermLayer          =  63
  TextLayer          =  64
  TerminalInRange    =  1.0
$End
* --- M0 - METAL LAYER ---
$Layer
  Number      =     1
  Name        =     M0
  Thickness   =     0.15
  Order       =     0
  Mask        =     1
  Lambda      =     0.09
  Filmtype    =     S
$End
* - I0- DIELECTRIC LAYER -
$Layer
  Number      =     2
  Name        =     I0
  Thickness   =     0.5
  Order       =     1
  Mask        =     -1
  Filmtype    =     I
  EpsilonR    =     4.6
$End
* --- M1 - METAL LAYER ---
$Layer
  Number      =     3
  Name        =     M1
  Thickness   =     0.6
  Order       =     2
  Mask        =     1
  Lambda      =     0.09
 Filmtype    =     S
$End
```

Execute *InductEx* with:

```
inductex microstrip.ixi -l microstrip.ldf -Z
```

The resulting output is shown below. Z is the characteristic impedance, L is the total inductance between the ports, C is the capacitance between the conductor and ground, T is the transmission delay, and v is the phase velocity.

Command Prompt

```
Transmission line structure between ports Z1 and Z2

Z = 27.851 ohm
L = 3.1644E-12 H
C = 4.0795E-15 F
T = 0.0075745 ps/um
v = 132.02 um/ps
Distance = 15 um
```

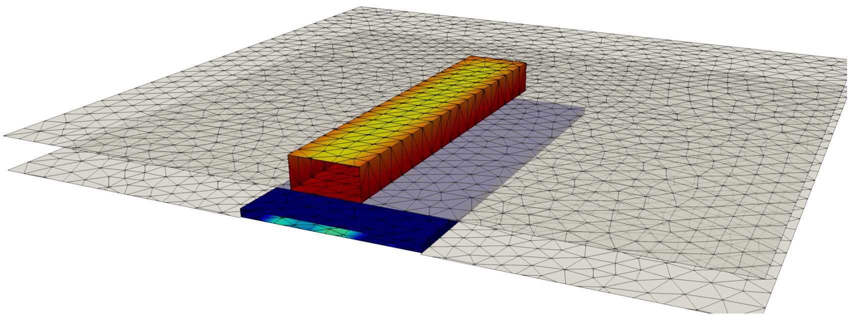


Figure 15.1: Cut-through of surface mesh for microstrip capacitance extraction as part of characteristic impedance calculation, with dielectric interfaces visible.

15.2 Characteristic impedance of stripline

Example works with Professional and Super licence tiers

This example, available in `examples/impedance/stripline/`, demonstrates:

- Multiple ground planes.
- Declaration of *InductEx* parameters inside IXI file.
- The use of structures in an IXI file to build up a layout.
- Port terminals connected to multiple layers.

This example demonstrates characteristic impedance calculation for a superconductor stripline with finite ground planes and a substrate below the lower ground with a relative permittivity of 11.7, and an upper half space with a relative permittivity of 4.6. The layout is based on a track routing architecture for RSFQ and AQFP circuits in a multilayer process [19].

The ground planes are seamed together with staggered vias, as is shown in Fig. 15.2.

In Layer Definition File 15.2.1, `multilayer.ldf`, the dielectric relative permittivity `EpsilonR` is set to 4.6 for the isolation layers, while the global parameter `EpsilonRBelow = 11.7` defines the substrate below the lowest die layer. In this example, the global parameter `EpsilonRAbove` must equal that of the isolation layers (`4.6`), otherwise *InductEx* will add a dielectric surface above the last layer in the layer stack.

Layer Definition File 15.2.1 — multilayer.ldf.

```
$Parameters
  Units              =  1e-6
  SegmentSize        =  0.5
  DielectricOverhang = 1.5
  DielectricMeshSize = 0.5
  AbsMin             =  0.025
  ProcessHasGroundPlane = TRUE
  Relief             =  FALSE
  GPLayer            =  40
  TermLayer          =  19
  TextLayer          =  18
  TerminalInRange    =  1.0
  EpsilonR           =  4.6
  EpsilonRBelow      =  4.6
  EpsilonRAbove      =  4.6
$End
$Layer
  Number      =     20
  Name        =     M2
  Thickness   =     0.200
  Lambda      =     0.090
  Order       =     6
  Mask        =     1
  Filmtype    =     S
$End
$Layer
  Number      =     21
  Name        =     I2
  Thickness   =     0.200
  Order       =     7
  Mask        =    -1
  Filmtype    =     I
  EpsilonR    =     4.6
$End
$Layer
  Number      =     30
  Name        =     M3
  Thickness   =     0.200
  Lambda      =     0.09
  Order       =     8
  Mask        =     1
  Filmtype    =     S
$End
$Layer
  Number      =     31
  Name        =     I3
  Thickness   =     0.200
  Order       =     9
  Mask        =    -1
  Filmtype    =     I
  EpsilonR    =     4.6
$End
$Layer
  Number      =     40
  Name        =     M4
  Thickness   =     0.200
  Lambda      =     0.09
  Order       =     10
  Mask        =     1
  Filmtype    =     S
$End
```

■ **IXI File 15.2.2 — stripline.ixi.**

```
$PARAMSTRING
  -l multilayer.ldf -Z
  -zoff
$END
$STRUCT
  Name routeblock
  $POLY
    Layer M2
    XY
    (  0.0, 0.0  )
    ( 10.0, 0.0  )
    ( 10.0, 10.0 )
    (  0.0, 10.0 )
    (  0.0, 0.0  )
  $END
  $POLY
    Layer M4
    XY
    (  0.0, 0.0  )
    ( 10.0, 0.0  )
    ( 10.0, 10.0 )
    (  0.0, 10.0 )
    (  0.0, 0.0  )
  $END
  $POLY
    Layer I3
    XY
    ( 9.05, 9.65 )
    ( 9.65, 9.65 )
    ( 9.65, 9.05 )
    ( 9.05, 9.05 )
    ( 9.05, 9.65 )
  $END
  $POLY
    Layer I3
    XY
    ( 0.35, 0.95 )
    ( 0.95, 0.95 )
    ( 0.95, 0.35 )
    ( 0.35, 0.35 )
    ( 0.35, 0.95 )
  $END
  $POLY
    Layer M3
    XY
    (  8.75, 1.25 )
    ( 10,    1.25 )
    ( 10,    0    )
    (  8.75, 0    )
    (  8.75, 1.25 )
  $END
  $POLY
    Layer M3
    XY
    ( 0,    1.25 )
    ( 1.25, 1.25 )
    ( 1.25, 0    )
    ( 0,    0    )
    ( 0,    1.25 )
  $END
  $POLY
    Layer M3
    XY
    (  8.75, 10    )
    ( 10,    10    )
    ( 10,     8.75 )
    (  8.75,  8.75 )
    (  8.75, 10    )
  $END
  $POLY
    Layer M3
    XY
    ( 0,    10    )
    ( 1.25, 10    )
    ( 1.25,  8.75 )
    ( 0,     8.75 )
    ( 0,    10    )
  $END
  $POLY
    Layer I2
    XY
    ( 9.05, 0.95 )
    ( 9.65, 0.95 )
    ( 9.65, 0.35 )
    ( 9.05, 0.35 )
    ( 9.05, 0.95 )
  $END
  $POLY
    Layer I2
    XY
    ( 0.35, 9.65 )
    ( 0.95, 9.65 )
    ( 0.95, 9.05 )
    ( 0.35, 9.05 )
    ( 0.35, 9.65 )
  $END
$END
$STRUCT
  Name ptlm3
  $PATH
    Layer M3
    Width 4.5
    XY
    (  0, 5 )
    ( 50, 5 )
  $END
  $SREF
    Name routeblock
      XY
      ( 40, 0 )
  $END
  $SREF
    Name routeblock
      XY
      ( 30, 0 )
  $END
  $SREF
    Name routeblock
      XY
      ( 20, 0 )
  $END
  $SREF
    Name routeblock
      XY
      ( 10, 0 )
  $END
  $SREF
    Name routeblock
      XY
      ( 0, 0 )
  $END
  $PATH
    Layer TERM
    Text  "Z1 [M3] [M4 M2]"
    XY
    ( 0, 7.25 )
    ( 0, 2.75 )
  $END
  $PATH
    Layer TERM
    Text  "Z2 [M3] [M4 M2]"
    XY
    ( 50, 7.250 )
    ( 50, 2.750 )
  $END
$END
```

Execute *InductEx* with:

```
inductex stripline.ixi
```

The output is shown below.

Command Prompt

```
Transmission line structure between ports Z1 and Z2

Z = 5.2684 ohm
L = 2.7493E-12 H
C = 9.9054E-14 F
T = 0.010437 ps/um
v = 95.813 um/ps
Distance = 50 um
```

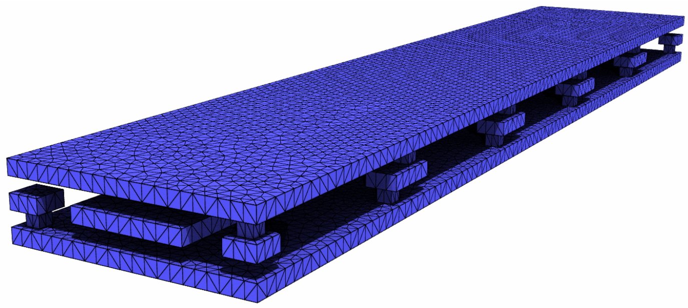


Figure 15.2: Rendering of stripline between finite ground planes with staggered seams, stretched in height for clarity.

15.3 S-parameters of microstrip filter

Example works only with Super licence tier

This example, available in `examples/sparams/hairpinline_bandpassfilter/`, demonstrates:

- S-parameter analysis.
- Ground boundaries.
- Units in millimetre.

For this example, a five-pole hairpin-line microstrip bandpass filter is analysed. A rendering of the filter model is shown in Fig. 15.3. The design is presented in [20], and has a centre frequency of 2 GHz. Substrate thickness is given as 1.27 mm and relative permittivity as 6.15. We assume values for the other material parameters and dimensions in this example.

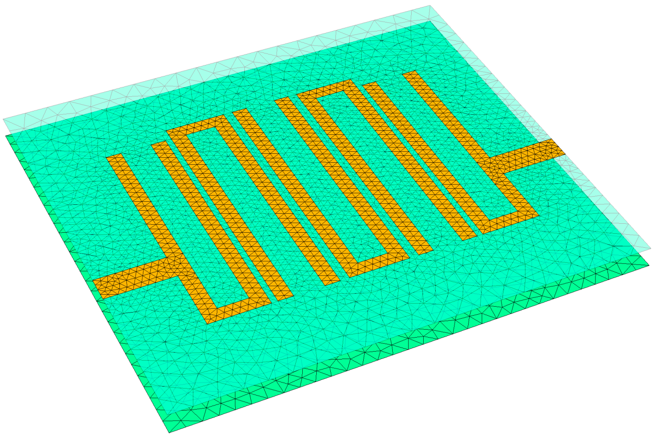


Figure 15.3: Rendering of hairpin-line microstrip bandpass filter.

The filter is constructed from 0.1 mm thick copper structures, with free space above the upper conductor. The ground conductor is defined as a perfect electrical conductor (`PEC = true`), and the ground plane edges at the north and south are electrically connected during simulation through the definition of a ground boundary (gbound1) inside the ground plane, close to each of the edges. The process is defined in Layer Definition File 15.3.1.

Layer Definition File 15.3.1 — microstripfilter.ldf.

```
$Parameters
  // Units in mm
  Units              = 1e-3
  AbsMin             = 0.025
  ProcessHasGroundPlane = TRUE
  GPLayer            = 1
  EpsilonR           = 6.15
  EpsilonRBelow      = 6.15
  EpsilonRAbove      = 1
  DielectricOverhang = 0.0
  DielectricMeshSize = 1.5
  TermLayer          = 19
  TextLayer          = 18
$End
* ------- M0 -------
$Layer
  Number      =     1
  Name        =     M0
  SegmentSize  =    1.00
  Thickness   =     0.1
  Sigma      =      58.7E3
  Order       =     0
  Mask        =     1
  Filmtype    =     R
  PEC   = true
$End
* ------- I0 -------
$Layer
  Number      =     2
  Name        =     I0
  Thickness   =     1.27
  Order       =     1
  Mask        =    -1
  EpsilonR    =     6.15
  Filmtype    =     I
$End
* ------- M1 -------
$Layer
  Number      =     3
  Name        =     M1
  SegmentSize  =    0.5
  Thickness   =     0.1
  // Cu conductivity in S/mm
  Sigma      =      58.7e3
  Order       =     2
  Mask        =     1
  Filmtype    =     R
$End
```

The filter geometry is described in parameterised IXI File 15.3.2.

IXI File 15.3.2 — hairpinline_bandpassfilter.ixi.

```
$PARAMSTRING
  -l microstripfilter.ldf
  -S LIN 1.2e9 2.8e9 200 50 50
$END
$VARIABLE
  Name wio
  Value 1.85
$END
$VARIABLE
  Name gwidth
  Value 30.0
$END

$STRUCT
  Name bpfilter
  $PATH
    Layer M1
Width %wio
    XY
    (  0   , 4.7 + %wio/2 )
    (  4.6, 4.7 + %wio/2 )
  $END
  $PATH
    Layer M1
Width %wio
    XY
    (  31.2, 4.7 + %wio/2 )
    (  26.6, 4.7 + %wio/2 )
  $END
  $PATH
    Layer M1
Width 1.0
    XY
    (  5.1, 17.4 )
    (  5.1, 0.5  )
(  8.1, 0.5  )
(  8.1, 17.4 )
  $END
  $PATH
    Layer M1
Width 1.0
    XY
    (  23.1, 17.4 )
    (  23.1, 0.5  )
(  26.1, 0.5  )
(  26.1, 17.4 )
  $END
  $PATH
    Layer M1
Width 1.0
    XY
    (  9.5 , 0.0  )
    (  9.5 , 17.9 )
(  12.5, 17.9 )
(  12.5, 0.0  )
  $END
  $PATH
    Layer M1
Width 1.0
    XY
    (  18.7, 0.0  )
    (  18.7, 17.9 )
(  21.7, 17.9 )
(  21.7, 0.0  )
  $END
  $PATH
    Layer M1
Width 1.0
    XY
    (  14.1, 18.4 )
    (  14.1, 0.5  )
(  17.1, 0.5  )
(  17.1, 18.4 )
  $END
  $POLY
    Layer M0
    XY
    ( 0    , 9.2 - %gwidth/2 )
    ( 31.2, 9.2 - %gwidth/2 )
( 31.2, 9.2 + %gwidth/2 )
    ( 0    , 9.2 + %gwidth/2 )
    ( 0    , 9.2 - %gwidth/2 )
  $END
  $PATH
    Layer TERM
    Text  "Z1 M1 M0"
    XY
    ( 0, 9.2 - %gwidth/2  )
    ( 0, 9.2 + %gwidth/2  )
  $END
  $PATH
    Layer TERM
    Text  "Z2 M1 M0"
    XY
    ( 31.2, 9.2 - %gwidth/2  )
    ( 31.2, 9.2 + %gwidth/2  )
  $END
  $TEXT
    text "gbound1 M0"
layer text
xy
(15.6, 9.2+%gwidth/2-0.1 )
  $END
  $TEXT
    text "gbound1 M0"
layer text
xy
    (15.6, 9.2-%gwidth/2+0.1 )
  $END
$END
```

For S-parameter extraction, *InductEx* is executed with the `-S` switch, followed by the start frequency, stop frequency, number of frequency steps and the source and load impedances. For this filter, source and load impedance is 50 Ω. The parameter string is defined in the IXI file, so that *InductEx* is simply executed with:

```
inductex hairpinline_bandpassfilter.ixi
```

The results are written directly to the file `sparams.m`, which can be plotted with Matlab to produce the graphs in Fig. 15.4. The screen output also contains plots, as shown below.

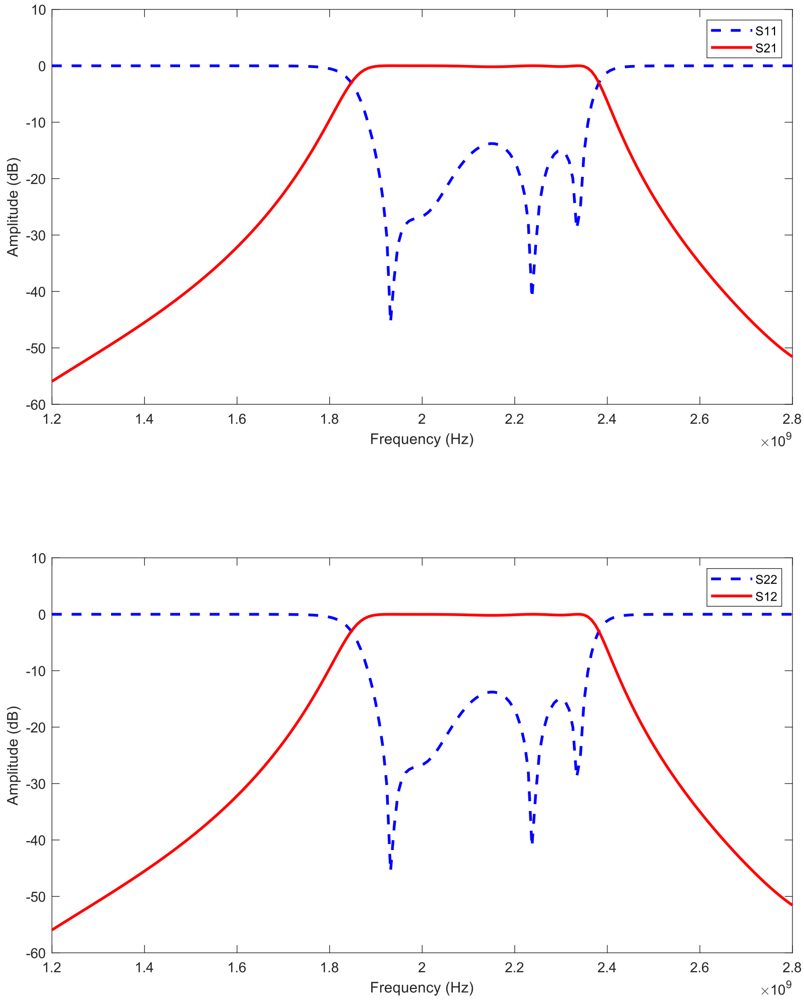


Figure 15.4: S-parameter plots for hairpin-line microstrip bandpass filter.

Command Prompt

```
A 1.65E-02  ************************-xxxxxxxxxxxxxxxxxxx***************** LEGEND
m           |                      *x                   xx              | S11 **
p           |                      x**                  *x              | S21 xx
l           |                     xx *                   xx             |
i           |                    xx  *                  * x             |
t           |                   xx    *       ****   ***   x            |
u           |                   x            **   * ***    xx           |
d           |                  x      *    **     * *  *    x           |
e           |                 x       *    *       **        x          |
            |                x          ***        *   *      x         |
(           |              xx          **              *       x        |
d           |             xx            *           *          xx       |
B           |            xx                                     xx      |
)           |          xx              *                         xx     |
            |        xxx                           *               xx   |
            |      xxx               *                              xx  |
            |    xxx                                                 xx |
            |  xxx                                                     xx
            |xxx                                                        x
  -5.59E+01 xx----------------------------------------------------------+
            1.20E+09                                             2.80E+09
            Frequency (Hz)

A 7.04E-04  ************************-xxxxxxxxxxxxxxxxxxx***************** LEGEND
m           |                      *x                   xx              | S22 **
p           |                      x**                  *x              | S12 xx
l           |                     xx *                  *xx             |
i           |                    xx  *                    x             |
t           |                   xx    *       ****   ***   x            |
u           |                   x            **   * ***    xx           |
d           |                  x      *    **     * *  *    x           |
e           |                 x       *    *       **        x          |
            |                x          ***        *   *      x         |
(           |              xx          **              *       x        |
d           |             xx            *           *          xx       |
B           |            xx                                     xx      |
)           |          xx              *                         xx     |
            |        xxx                           *               xx   |
            |      xxx               *                              xx  |
            |    xxx                                                 xx |
            |  xxx                                                     xx
            |xxx                                                        x
  -5.60E+01 xx----------------------------------------------------------+
            1.20E+09                                             2.80E+09
            Frequency (Hz)
```

15.4 S-parameters of microstrip filter with multilayer dielectrics

Example works only with Super licence tier

This example, available in `examples/sparams/multilayer_dielectrics/`, demonstrates:

- Dimensions in mil.
- Multilayer dielectrics.

For this example, a coupled microstrip filter geometry with top, ground and intermediate metal layers and multiple dielectric layers is analysed. A rendering of the filter model is shown in Fig. 15.5. The design is presented in [21], and has resonant frequencies at around 2.14 and 2.45 GHz. All dimensions are in mil.

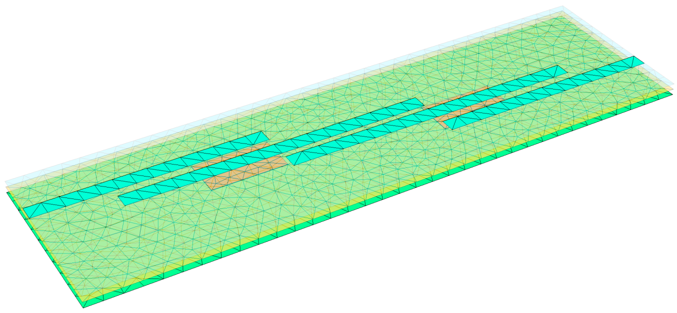


Figure 15.5: Rendering of a coupled microstrip filter with multiple layers and dielectrics.

The ground plane is modelled as a PEC, while other layers are copper. For the copper layers, we use bulk conductivity of 58.7 S/m and define a mil as 25.4 µm, which translates to `Sigma` = $58.7 \times 10^6 * 25.4 \times 10^{-6} = 1491$ siemens per mil. The ground plane edges at the north and south are electrically connected through the definition of a ground boundary (gbound1) inside the ground plane, close to each of the edges. The process is defined in Layer Definition File 15.4.1.

■ Layer Definition File 15.4.1 — process_multilayer.ldf.

```
$Parameters
  Units               =  25.4e-6
  AbsMin              =  0.025
  ProcessHasGroundPlane = FALSE
  EpsilonRBelow       =  3.78
  EpsilonRAbove       =  1
  DielectricOverhang = 0
  DielectricMeshSize = 100
  TermLayer           =  19
  TextLayer           =  18
$End
* ------- M0 -------
$Layer
  Number      =     1
  Name        =     M0
  SegmentSize  =    100
  Thickness   =     1.2
  Sigma      =      1.491E3
  Order       =     0
  Mask        =     1
  Filmtype    =     R
  PEC         =     true
$End
* ------- I0 -------
$Layer
  Number      =     2
  Name        =     I0
  Thickness   =     31.5
  Order       =     1
  Mask        =    -1
  EpsilonR    =     3.78
  Filmtype    =     I
$End
* ------- M1 -------
$Layer
  Number      =     3
  Name        =     M1
  SegmentSize  =    100
  Thickness   =     1.2
  Sigma      =      1.491E3
  Order       =     2
  Mask        =     1
  Filmtype    =     R
$End
* ------- I1 -------
$Layer
  Number      =     4
  Name        =     I1
  Thickness   =     31.5
  Order       =     3
  Mask        =    -1
  EpsilonR    =     2.31
  Filmtype    =     I
$End
* ------- M2 -------
$Layer
  Number      =     5
  Name        =     M2
  SegmentSize   =   100
  Thickness   =     1.2
  Sigma      =      1.491E3
  Order       =     4
  Mask        =     1
  Filmtype    =     R
$End
```

The filter geometry is described in parameterized IXI File 15.4.2.

IXI File 15.4.2 — microstripfilter_multilayer.ixi.

```
$PARAMSTRING
  -l process_multilayer.ldf
  -S LIN 0.5e9 3.0e9 200 50 50
$END
$VARIABLE
  Name L1
  Value 435
$END
$VARIABLE
  Name L2
  Value 870
$END
$VARIABLE
  Name w
  Value 90
$END
$VARIABLE
  Name Sout
  Value 50
$END
$VARIABLE
  Name Sin
  Value 35
$END
$VARIABLE
  Name PadL
  Value 435
$END
$VARIABLE
  Name PadW
  Value 230
$END
$VARIABLE
  Name PadOff
  Value 45
$END
$VARIABLE
  Name glength
  Value %L1*2 + %L2*3
$END
$VARIABLE
  Name gwidth
  Value 1000
$END
$STRUCT
  Name microstripfilter
  $PATH
    Layer M0
    Width %gwidth
    XY
    ( -%glength/2, 0  )
    (  %glength/2, 0  )
  $END
  $PATH
    Layer M1
   Width %PadW
    XY
    ( -%L2/2-%PadL,-%PadOff+%PadW/2 )
    ( -%L2/2      ,-%PadOff+%PadW/2 )
  $END
  $PATH
    Layer M1
Width %PadW
    XY
    ( %L2/2      ,%PadOff-%PadW/2 )
    ( %L2/2+%PadL,%PadOff-%PadW/2 )
  $END
  $PATH
    Layer M2
   Width %w
    XY
    ( -%L2/2-%L2-%L1,%Sin/2+%w+%Sout+%w/2 )
    ( -%L2/2         %Sin/2+%w+%Sout+%w/2 )
  $END
  $PATH
    Layer M2
Width %w
    XY
    ( -%L2/2-%L2 , %Sin/2+%w/2  )
    (  %L2/2     , %Sin/2+%w/2  )
  $END
  $PATH
    Layer M2
    Width %w
    XY
    ( -%L2/2     , -%Sin/2-%w/2  )
    (  %L2/2+%L2 , -%Sin/2-%w/2  )
  $END
  $PATH
    Layer M2
    Width %w
    XY
    ( %L2/2          ,-%Sin/2-%w-%Sout-%w/2 )
    ( %L2/2+%L2+%L1,-%Sin/2-%w-%Sout-%w/2 )
  $END
  $PATH
    Layer TERM
    Text  "Z1 M2 M0"
    XY
    ( -%L2/2-%L2-%L1,%Sin/2+%w+%Sout     )
    ( -%L2/2-%L2-%L1,%Sin/2+%w+%Sout+%w )
  $END
  $PATH
    Layer TERM
    Text  "Z2 M2 M0"
    XY
    ( %L2/2+%L2+%L1,-%Sin/2-%w-%Sout-%w )
    ( %L2/2+%L2+%L1,-%Sin/2-%w-%Sout     )
  $END
  $TEXT
    text "gbound1 M0"
    layer text
    xy
    ( 0, %gwidth/2*0.95 )
  $END
  $TEXT
    text "gbound1 M0"
    layer text
    xy
    (  0, -%gwidth/2*0.95 )
  $END
$END
```

For this filter, source and load impedance is 50 Ω. The parameter string is defined in the IXI file, so that *InductEx* is simply executed with:

```
inductex microstripfilter_multilayer.ixi
```

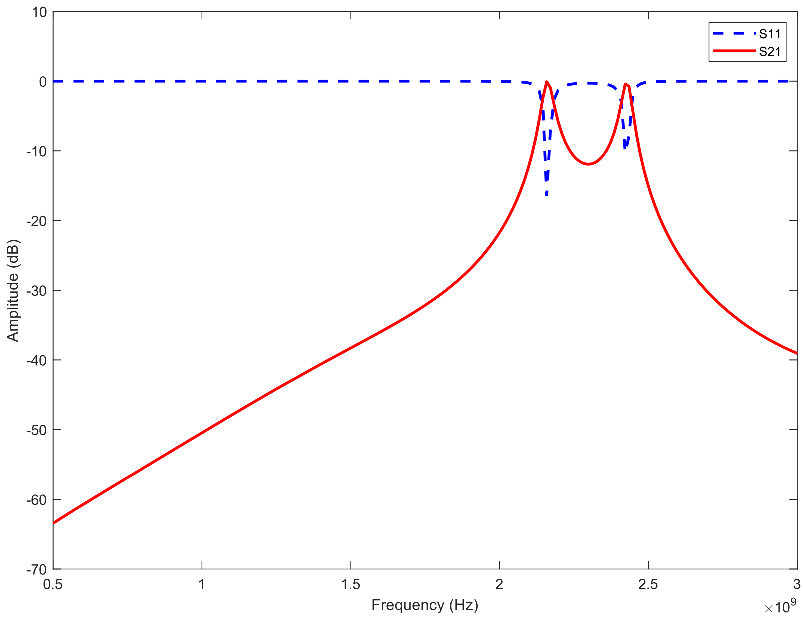


Figure 15.6: S-parameter plots for microstrip filter with multiple dielectrics.

The calculated results for S_{11} and S_{21} are shown Fig. 15.6.

15.5 S-parameters of coplanar interdigital filter

Example works only with Super licence tier

This example, available in `examples/sparams/sc_coplanar_interdigitalfilter/`, demonstrates:

- S-parameter analysis of a coplanar stucture.
- Creation of air bridges with vias and conductors.
- Slow convergence.

For this example, a three-pole Chebyshev bandpass filter implemented as a superconductor coplanar interdigital filter is analysed. A rendering of the filter model is shown in Fig. 15.7. The design is presented in [22], and has a centre frequency around 10 GHz.

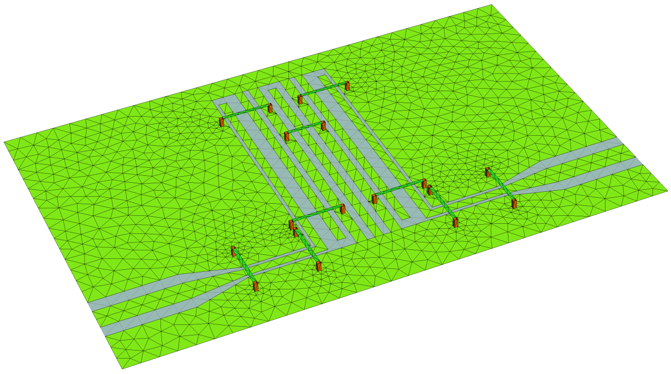


Figure 15.7: Rendering of a coplanar interdigital filter.

The superconductor layer is modelled as niobium with thickness 200 nm and penetration depth 90 nm. *InductEx* does not support anisotropic permittivity, so that the sapphire dielectric below the filter is modelled with $\varepsilon_r = 10.31$. Air bridges are defined as vias and metal connections in layers above the coplanar layer. In this model, the air bridge material is selected as niobium. The ground plane edges at the north and south are electrically connected through the definition of a ground boundary (gbound1) inside the ground plane, close to each of the edges.

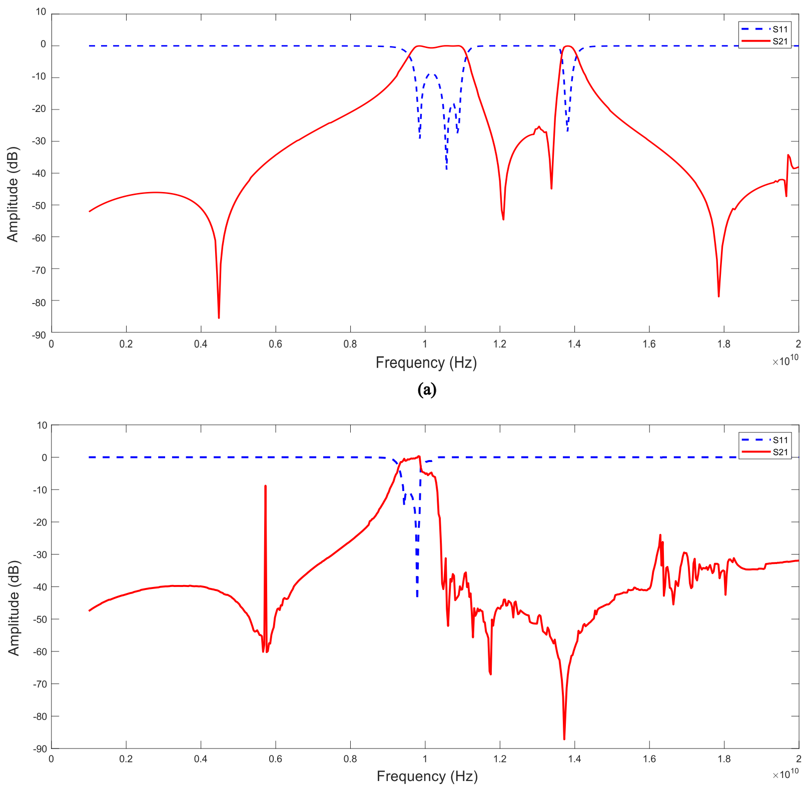
The filter geometry is described in a parameterized IXI file, of which only the parameter string block is shown in 15.5.1.

■ IXI File 15.5.1 — interdigitalfilter.ixi.

```
$PARAMSTRING
  -l interdigitalfilter.ldf
  -S LIN 1e9 20e9 400 50 50
  -c 2000
$END
...
```

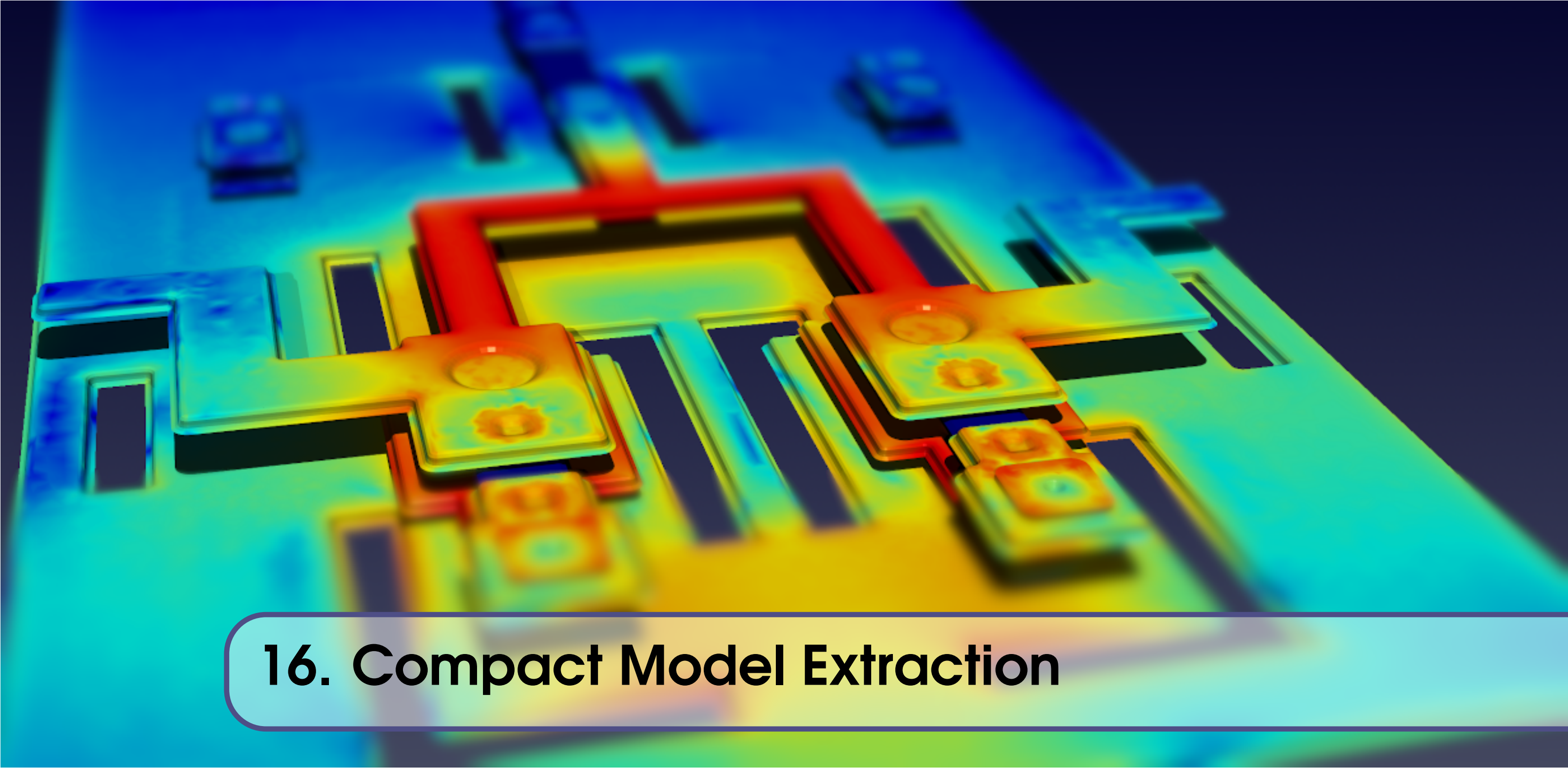
The inclusion of air bridges affects the convergence rate of *TetraHenry* , so that it needs more than the default of 400 iterations to reach convergence at every frequency step. To prevent loss of accuracy, we set the iteration limit to 2000 with the `-c 2000` command line parameter.

The calculated results for S_{11} and S_{21} are shown Fig. 15.8 for two analyses: one without any air bridges, and the other with superconductor air bridges included.



(b)

Figure 15.8: S-parameter plots for coplanar interdigital filter with (a) no air bridges and (b) air bridges.



16. Compact Model Extraction

16.1 Flux trapping analysis

> **Example works only with Super licence tier**
>
> This example, available in `examples/fluxtrapping/squidmoats/`, demonstrates:
> - Flux trapping analysis.
> - Critical field calculation for flux trapping.
> - Back-annotation of extracted results.

The layout of a basic two-junction SQUID with direct modulation inputs and a centre bias over a ground plane is shown in Fig. 16.1. Two moats are defined in the ground plane. For extraction purposes, the ground plane is defined as a positive mask layer and it is not cropped to near the layout structures. This example is very similar to those presented in [8].

For flux trapping analysis, each moat that requires analysis is labelled with "F*name*". Here, we chose `F1` and `F2`. The "F" label specifies the position of a fluxon path in the *xy* plane of a thin-film layout. The path is extended 10 Units above and below the zero *z* plane, and then closed at infinity in the *x* direction. More complicated fluxon paths can be defined in the circuit netlist.

The circuit netlist is shown in Netlist 16.1.1. Note that a control block is included, and that *InductEx* is instructed to back-annotate the extracted results and the compact simulation model for flux in moats to the *JoSIM* netlist in `squidsim.cir`, which is listed in Netlist 16.1.2. Inductor L_{IB} is a dc bias inductor (in practice it is in series with a resistor) and must therefore be excluded from coupling extraction. This is set by adding COUPLING=OFF (without any spaces) at the end of the netlist card for L_{IB}.

The extracted value for inductor L_1 will be assigned to inductor L_1 in the *JoSIM* netlist, and the same for inductor L_2.

Netlist 16.1.1 — squidmoats.cir.

```
.control
BACK-ANNOTATE squidmoatsim.cir
```

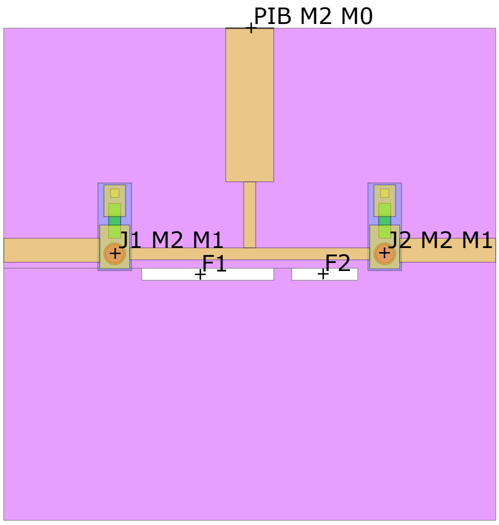


Figure 16.1: GDS layout view of a basic two-junction SQUID loop with a ground plane and two moats.

```
.endc
L1      1   2   5p  [L1]
L2      2   3   5p  [L2]
LIB     10  2   coupling=off
PIB     10  0
J1      1   0
J2      3   0
.end
```

■ **Netlist 16.1.2 — squidmoatsim.cir.**

```
* JoSIM SQUID simulation netlist
IS     0 10    pwl(0 0 10p 0 1000p 800E-6)
LS     10    3     4.4E-12
L1     1     3     4.55E-12
L2     3     2     4.55E-12
B1     1     0     jj1 area=0.25
B2     2     0     jj1 area=0.25
R1     1     0     2.8
R2     2     0     2.8
.model jj1 jj(rtype=1, vg=2.8m, cap=0.07p, r0=160, rn=16, icrit=1m)
.tran 0.05p 0.99n 0.0 0.05p
.print PHASE B1
.print DEVI IS
.print NODEV 3 0
.end
```

■ **Netlist 16.1.3 — squidsim_annotated.cir.**

```
* Back-annotated simulation file written by InductEx
* ---------------------------------------------------------
*  Trapped flux compact model; generated with InductEx
* --------------------------------------------------------
```

```
.param phase-f2 = (2*PI) * (0)
.param phase-f1 = (2*PI) * (0)

IS    0    10    pwl(0 0 10p 0 1000p 800E-6)
LS    10   3     4.4E-12
L1    1    3     4.552E-012
L2    3    2     4.547E-012
B1    1    0     jj1 area=0.25
B2    2    0     jj1 area=0.25
R1    1    0     2.8
R2    2    0     2.8

.model jj1 jj(rtype=1, vg=2.8m, cap=0.07p, r0=160, rn=16, icrit=1m)
.tran 0.05p 0.99n 0.0 0.05p
.print PHASE B1
.print DEVI IS
.print NODEV 3 0
* Fluxon phase sources from . 1 fluxon = 2*pi
Pf2  5000  0   pwl(0 0 1p 0 10p phase-f2)
Lf2  5000  0  4.27326E-012
Pf1  5001  0   pwl(0 0 1p 0 10p phase-f1)
Lf1  5001  0  6.81451E-012
K<f2><1>  Lf2  L1  -0.00295045
K<f2><2>  Lf2  L2  0.109427
K<f1><1>  Lf1  L1  0.121324
K<f1><2>  Lf1  L2  0.0291004
K<f1><f2>  Lf1  Lf2  -0.171452
.end
```

InductEx is executed from the command prompt with:

```
inductex squidmoats.gds -l process.ldf -i squidmoats.geo -th
-n squidmoats.cir
```

An excerpt from the output is shown below.

Command Prompt

```
Inductance [H]
Name        Design        Extracted     AbsDiff       PercDiff
L1          5E-12         4.55172E-12   -4.4828E-13   -8.9656%
L2          5E-12         4.5461E-12    -4.539E-13    -9.078%
LIB         --            4.46745E-12   +4.4674E-12   --%
LF2         --            4.27326E-12   +4.2733E-12   --%
LF1         --            6.81451E-12   +6.8145E-12   --%

Mutual Inductance [H]                                                   Coupling factor
Name        Design        Extracted     AbsDiff       PercDiff                k
M<F2><1>    ---           -1.3012E-14   -1.3012E-14   --%           -2.950E-3
M<F2><2>    ---           +4.8231E-13   +4.8231E-13   --%           +0.10943
M<F1><1>    ---           +6.757E-13    +6.757E-13    --%           +0.12132
M<F1><2>    ---           +1.6197E-13   +1.6197E-13   --%           +0.0291
M<F1><F2>   ---           -9.2521E-13   -9.2521E-13   --%           -0.17145
```

Note that *InductEx* added an inductor for each moat labelled for extraction (L_{F1} and L_{F2}), calculated the inductance of each, and calculated the mutual inductance between each of these moats and the circuit inductances L_1 and L_2.

The full simulation model is written to the file `squidmoatsim_annotated.cir` (so as not to overwrite the original file) listed in Netlist 16.1.3. With *JoSIM*, the number of fluxons in each moat can now be set as an integer value in the `.param` statements. Positive values denote fluxons with positve magnetic field intensity in the positive z direction.

The annotated simulation netlist can be simulated directly with *JoSIM*:

```
josim -o output.csv squidmoatsim_annotated.cir
```

A plot of the output, which measures the voltage over the SQUID and the phase over J_1, shows that the SQUID has a critical current of about 500 µA, as expected.

If a fluxon is added to the first moat, with `.param phase-f1 = (2*PI) * (1)`, then the SQUID critical current reduced to around 470 µA. With a fluxon in the second moat (and no fluxon in the first moat), the critical current is also around 470 µA. With a fluxon in each of the moats, the SQUID critical current reduces to around 430 µA.

Note: Simulation requires the use of phase sources and a simulator capable of the modified nodal phase method, such as *JoSIM*. The simulation must be run in phase mode, which is the default for *JoSIM*.

We can calculate the critical field for each moat with:

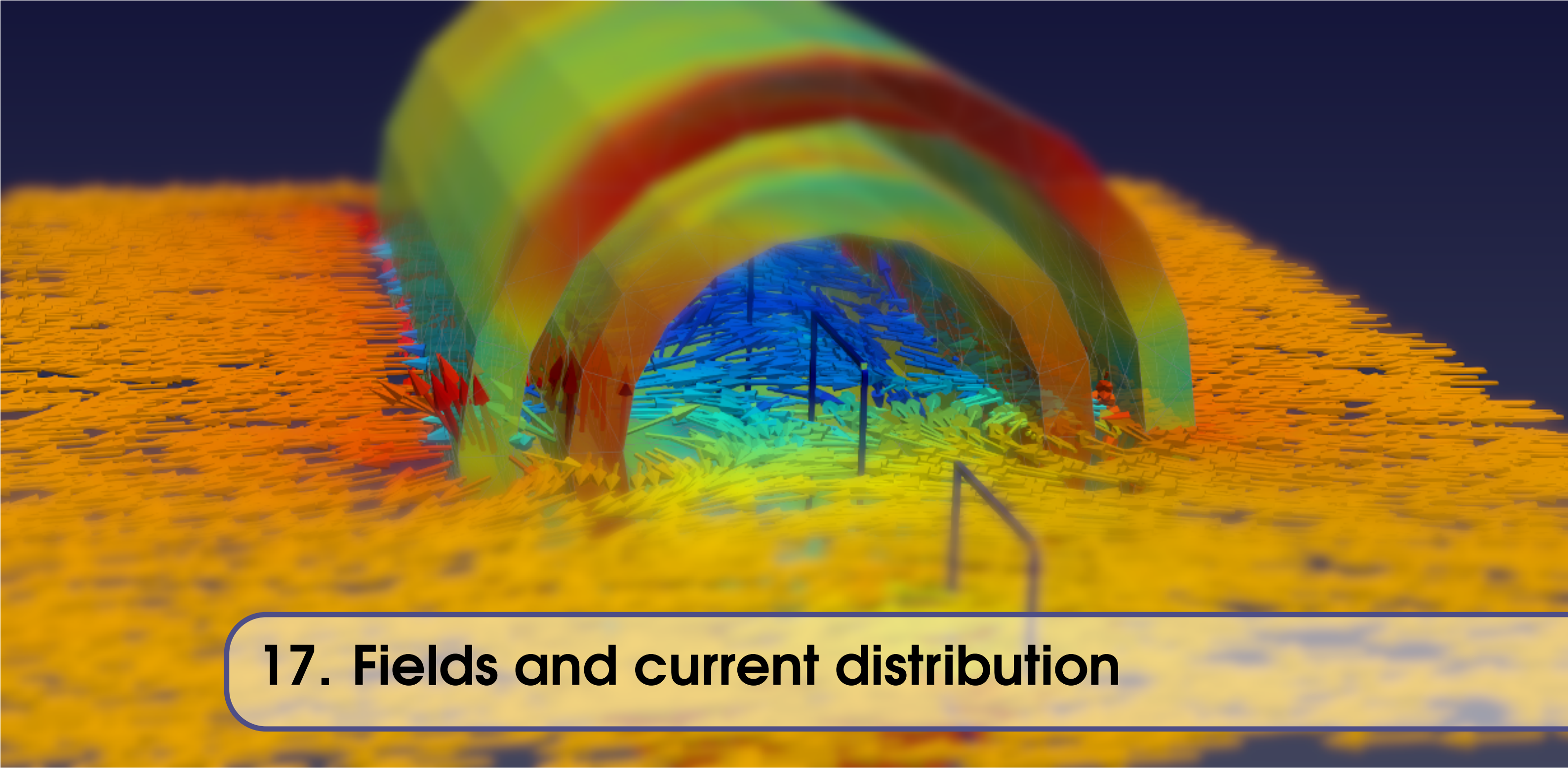
```
inductex squidmoats.gds -l process.ldf -i squidmoatscritfields.geo -B
```

The results are:

Command Prompt

```
Critical fields [T]: Bx          By          Bz
F2                   0.02668     0.000757    1.107E-05
F1                   0.02903     0.0004683   6.811E-06
```

The moats are in the ground plane, so that the critical fields in the z-direction are relevant. These are 6.81 µT for the moat marked F1 and 11.1 µT for the moat marked F2.



17. Fields and current distribution

17.1 Magnetic field caused by current in a rectangular loop

Example works with Professional and Super licence tiers

This example, available in `examples/fields/magneticfield_loop/`, demonstrates:

- Magnetic field intensity calculation.
- Magnetic flux density calculation.
- Cuboid volume for field plot.
- Field plots over multiple planes.
- Current sources.

Consider the calculation of magnetic field intensity around a rectangular loop when a current flows in the loop. An analytical solution is given in [23].

For this example, we use a wire loop with side lengths of 20 µm and 40 µm, and wire width and height of 0.1 µm.

■ **IXI File 17.1.1 — rectangularloop.ixi.**

```
$PARAMSTRING
  -l process.ldf -i loop.geo
  -th -M -n loop.cir
$END
$VARIABLE
  Name  a
  Value 20
$END
$VARIABLE
  Name  b
  Value 10
$END
$VARIABLE
  Name  w
  Value 1
$END
$STRUCT
  Name rectangularloop
  $PATH
    Layer M0
    Width %w
    XY
    ( -%a+0.6*%w, -%b )
    (  %a, -%b )
    (  %a,  %b )
    ( -%a,  %b )
    ( -%a, -%b+0.6*%w )
  $END
  $PATH
    Layer TERM
    Text  "P1+ M0"
    XY
    ( -%a+0.6*%w, -%b-%w/2 )
    ( -%a+0.6*%w, -%b+%w/2 )
  $END
  $PATH
    Layer TERM
    Text  "P1- M0"
    XY
    ( -%a-%w/2, -%b+0.6*%w )
    ( -%a+%w/2, -%b+0.6*%w )
  $END
$END
```

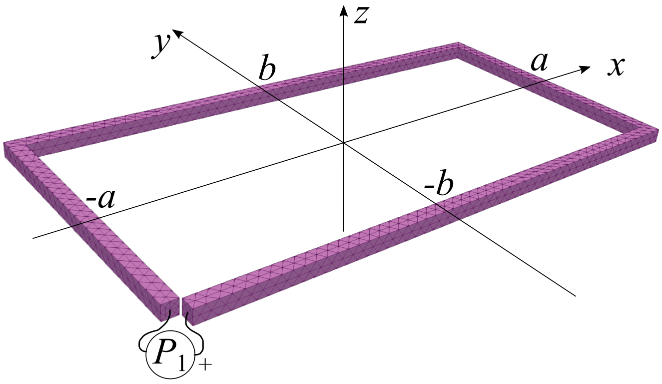


Figure 17.1: Geometry for a single rectangular loop with an excitation port connected.

If the loop is a superconductor, we can calculated the inductance in the simplest way by opening the loop and connecting a single excitation port P_1.

■ Netlist 17.1.2 — loop.cir.

```
* Title: Netlist for InductEx user manual - single inductor and port
L1      1   0
P1      1   0
.end
```

The layer definition file needs only a single layer and no ground plane.

■ Layer Definition File 17.1.3 — process.ldf.

```
$Parameters
  Units                = 1e-6
  SegmentSize          = 0.75
  ProcessHasGroundPlane = FALSE
  TermLayer            = 63
  TextLayer            = 64
  FieldCageScale    =  2.0
  FieldCageZScale   =  1
  FieldMeshSize     =  3.0
$End
* M0 - METAL LAYER
$Layer
  Number      =     1
  Name          =     M0
  Thickness     =     1
  Order         =     0
  Mask          =     1
  Lambda        =     0.1
  Filmtype      =     S
$End
```

The entire parameter string has been buried in the IXI file, so that *InductEx* is executed from the command prompt simply with:

```
inductex rectangularloop.ixi
```

For inductance extraction, port P_1 is excited with 1 V$\angle 0°$ at the default frequency (10×10^9 Hz). The result is an inductance L_1 of 7.84403×10^{-11} H, which implies a current of:

$$\frac{1}{2\pi(1 \times 10^{10})(7.84403 \times 10^{-11})i} = (0 - 0.2029\mathrm{i})\ \mathrm{A}.$$

The current measured through the port P_1 is opposite in polarity to the inductor current, because the port current is measured from the positive to the negative terminals of the port. The measured port current appears in the output file `./output/i.txt` as "`p1 p1 0.0000000000e+00 2.0289940572e-01j`".

The magnetic field intensity is written to `H_field.vtk`. A 3D plot of **H** is shown in Fig. 17.2. The field plot volume has x and y dimensions of 82 µm, twice that of the largest dimension ($2a + w$ = 41 µm). The z dimension is also 82 µm because to `FieldCageZScale` = 1.

The current is purely imaginary with a negative sign, so that a plot of the imaginary current shows the current direction to be clockwise through the rectangular loop (flowing into the positive terminal of port P_1). The magnetic field intensity is therefore also imaginary, directed from top to

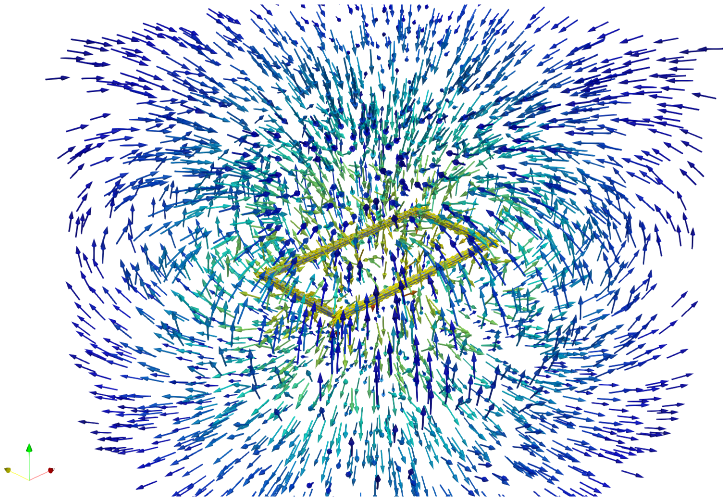


Figure 17.2: Magnetic field intensity around a superconductor loop excited with a 1 V source.

bottom through the loop area in the z axis. Inspection of the imaginary components of **H** at any point in the field plot shows it to match the result calculated for a loop current of -0.2029 A [23].

The example can be altered (`examples/fields/fieldplanes/`, to:

- calculate magnetic flux density in Tesla,
- plot the flux density over 2D planes instead of a 3D volume,
- use a current source rather than a voltage source,
- and use a different excitation frequency.

The command line switch `-MB` selects magnetic flux density, while 2D field planes are defined in the IXI file as listed in 17.1.4. A field plane is always defined as a rectangle of which only two opposite corners in the xy plane are given. The first plane (plane 0) is translated by -2 Units in the z direction. The second plane (plane 1) is rotated 90 degrees around the x axis.

In the LDF file 17.1.5, the layer is changed to a resistive layer with conductivity of $\sigma = 1 \times 10^3/(\Omega\,\mu\mathrm{m})$. The port in the circuit netlist file 17.1.6 is defined as a current source with amplitude -1 A. Current source convention is the same as that of SPICE, so that current flows from the positive to negative node in a current source. A negative source current thus translates into a positive current flowing into the circuit (and thus a positive real current flowing counter-clockwise around the rectangular loop in this example). The source excitation frequency is set to 1 kHz.

IXI File 17.1.4 — rectangularloop2.ixi.

```
$PARAMSTRING
  -l process2.ldf -i loop.geo
  -th -MB -n loop2.cir
$END
$VARIABLE
  Name a
  Value 20
$END
$VARIABLE
  Name b
  Value 10
$END
$VARIABLE
  Name w
  Value 1
$END
$FPLANE
  Rotate
  ( 0, 0, 0 )  // degrees
  Translate
  ( -20e-6, 0, -2E-6 )  // (m)
// Coordinates (only 2) in m
// specify 2 opposing corners
// in XY-plane. Z-coord ignored
  Coords
  (-3/2*%a*1E-6, -3/2*%b*1E-6)
  ( 3/2*%a*1E-6,  3/2*%b*1E-6)
$END
$FPLANE
  Rotate
  ( 90, 0, 0 )
  Translate
  ( 0, 0, 0 )
  Coords
  (-5/2*%a*1E-6,-5/2*%b*1E-6,0)
  ( 5/2*%a*1E-6, 5/2*%b*1E-6,0)
$END
$STRUCT
  Name rectangularloop
  $PATH
    Layer M0
    Width %w
    XY
    (  -%a+0.6*%w, -%b  )
    ( %a, -%b  )
    ( %a,  %b )
    ( -%a, %b )
    ( -%a, -%b+0.6*%w  )
  $END
  $PATH
    Layer TERM
    Text  "P1+ M0"
    XY
    ( -%a+0.6*%w, -%b-%w/2 )
    ( -%a+0.6*%w, -%b+%w/2 )
  $END
  $PATH
    Layer TERM
    Text  "P1- M0"
    XY
    ( -%a-%w/2, -%b+0.6*%w )
    ( -%a+%w/2, -%b+0.6*%w )
  $END
$END
```

Layer Definition File 17.1.5 — process2.ldf.

```
$Parameters
  Units              =  1e-6
  SegmentSize        =  0.75
  ProcessHasGroundPlane = FALSE
  TermLayer          =  63
  TextLayer          =  64
  FieldMeshSize      =  3.0
  Frequency          =  1E3
$End
$Layer
  Number      =     1
  Name        =     M0
  Thickness   =     1
  Order       =     0
  Mask        =     1
  Sigma       =     1E3
  Filmtype    =     R
$End
```

Netlist 17.1.6 — loop2.cir.

```
* Title: Netlist for InductEx user manual - port is current source
L1     1   0
P1     1   0   source=i  amplitude=-1
.end
```

InductEx is executed from the command prompt with:

```
inductex rectangularloop2.ixi
```

The current in the output file `ixth.cur`, listed as the output of the port into the circuit, is 1 A (real). The magnetic flux density is written to `/output/B_loop.vtk`. A plot of the real part of **B** in the planes and the plane positions referenced to the current loop is shown in Fig. 17.3. The flux density at a given coordinate matches that calculated for a loop current of 1 A from [23].

No inductance is extracted in this case, because a current source is used.

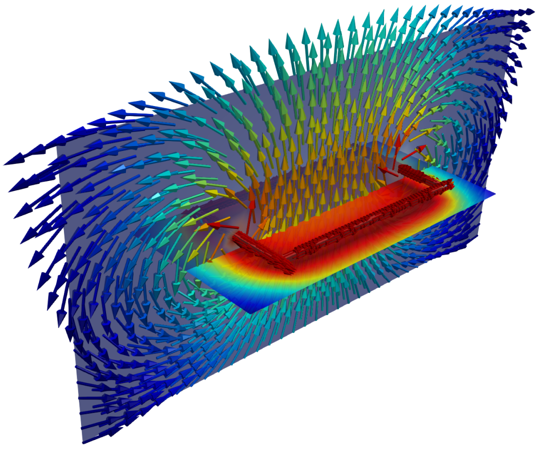


Figure 17.3: Magnetic flux density in planes near a superconductor loop excited with a 1 A source.

17.2 Coupling of external field to coils with mu-metal shields

Example works only with Super licence tier

This example, available in `examples/fields/mumetalshield/`, demonstrates:

- External magnetic fields.
- Import of external geometries.
- Parameterisation of external geometries.
- Magnetic permeability.
- Assignment of material properties to imported geometries.
- Specification of frequency.

This example shows the calculation of coupling of an external field to a set of square loops, some of which are shielded by a multi-layer high permeability shield. We use a wire loop with side lengths of 40 µm and wire width and height of 1 µm for the square loops, and two capped cylinders for the shield. The global parameter `Frequency` is set to 1 Hz.

Layer Definition File 17.2.1 — processloops.ldf.

```
$Parameters
  Units                  =  1e-6
  Frequency              =  1
  SegmentSize            =  1
  ProcessHasGroundPlane = FALSE
  TermLayer              =  63
  TextLayer          =  64
  FieldMeshSize      =  2.0
$End
$Layer
  Number       =     1
  Name         =     M0
  Thickness     =     1
  Order         =     0
  Mask          =     1
  Lambda        =     0.1
  Filmtype      =     S
$End
```

Gmsh GEO File 17.2.2 — cylinderinner.geo.

```
// Variables: name must start with "_" for InductEx
_lc   = 0.02e-3; // Characteristic length
_diam = 0.1e-3;  // Outer diameter (m)
_len  = 0.2e-3;  // Cylinder length (m)
_t = 0.01e-3;    // Plate thickness (m)
_step = 2*Pi/20;
Point(1) = {_diam/2*Cos (_step*0), _diam/2*Sin(_step*0), 0, _lc};
Point(2) = {_diam/2*Cos(_step*1), _diam/2*Sin(_step*1), 0, _lc};
...
Point(20) = {_diam/2*Cos(_step*19), _diam/2*Sin(_step*19), 0, _lc};
Point(21) = {_diam/2*Cos(_step*0), _diam/2*Sin(_step*0), _len, _lc};
...
Point(40) = {_diam/2*Cos(_step*19), _diam/2*Sin(_step*19), _len, _lc};
Point(41) = {(_diam/2-_t)*Cos(_step*0), (_diam/2-_t)*Sin(_step*0), _len, _lc};
...
Point(60) = {(_diam/2-_t)*Cos(_step*19), (_diam/2-_t)*Sin(_step*19), _len, _lc};
Point(61) = {(_diam/2-_t)*Cos(_step*0), (_diam/2-_t)*Sin(_step*0), _t, _lc};
...
Point(80) = {(_diam/2-_t)*Cos(_step*19), (_diam/2-_t)*Sin(_step*19), _t, _lc};
Line(1) = {1, 2};
...
Line(120) = {80, 60};
Line Loop(1) = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20};
...
Line Loop(44) = {-120, 80, 101, -60};
Plane Surface(1) = {1};
Plane Surface(2) = {2, 3};
...
Plane Surface(43) = {44};
Surface Loop(1) = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ... 41, 42, 43};
Volume(1) = {1};
Physical Volume("Innershield") = {1};
```

The capped cylinders are defined as parameterised objects in Gmsh `.geo` files.

Note that the cylinder is described with parameters which are defined as variables in `cylinderinner.geo` in GEO file 17.2.2. *InductEx* will only recognise variables that start with the "_" character. Any number of volumes can be included in the geometry file, although attributes for each volume have to be defined with a separate "Physical Volume" line. The physical volume line in this example sets the conductivity (sigma) and penetration depth (lambda) as zero with `-s 0.0` and `-l 0.0` respectively. The relative permeability is set to 50×10^3 with `-u 50e3`.

The inner cylinder model, meshed with Gmsh, is shown in Fig. 17.4. It is open to the positive z direction.

A the outer cylinder model uses the exact same structure, but different variable values to achieve different dimensions. The variable declaration of `cylinderouter.geo` are shown in GEO file excerpt 17.2.3.

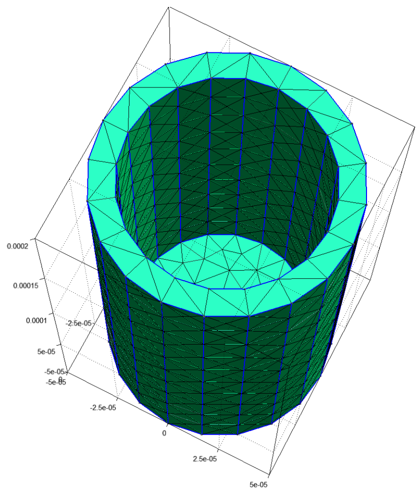


Figure 17.4: Capped cylinder as meshed with Gmsh from file cylinderinner.geo.

■ Gmsh GEO File 17.2.3 — cylinderouter.geo.

```
_lc = 0.02e-3;   // Characteristic length
_diam = 0.14e-3; // Outer diameter (m)
_len = 0.22e-3;  // Cylinder length (m)
_t = 0.01e-3;    // Plate thickness (m)
_step = 2*Pi/20;
Point(1) = {_diam/2*Cos (_step*0), _diam/2*Sin(_step*0), 0, _lc};
...
Volume(1) = {1};
Physical Volume("Outershield") = {1};
```

Three loops are defined, each separated by a distance d from their respective centres, shown in IXI file 17.2.4. The parameter string includes `-b` to request an external field analysis, and `-MB` to request a magnetic flux density plot. A single field plane is defined and rotated to slice the shield cylinders in the yz plane.

A double-walled cylinder is built from `cylinderinner.geo` and `cylinderouter.geo`, each imported, rotated and shifted as required.

■ IXI File 17.2.4 — loops.ixi.

```
$PARAMSTRING
  -l processloops.ldf
  -i loops.geo
  -th -MB -b
  -n loops.cir
  -zoff
$END
$VARIABLE
  Name a
  Value 20
$END
$VARIABLE
  Name w
  Value 1
$END
$VARIABLE
  Name d
  Value 100
$END
$FPLANE
  Rotate
  ( 90, 0, 0 )
  Translate
  ( 0, 0, 0 )
  Coords
  ( -150E-6, -150E-6, 0 )
  (  250E-6,  150E-6, 0 )
$END
$MATERIAL
  Name     "Mumetal"
  Sigma    0.0
  Lambda   0.0
  MuR      50E3
$END
$IMPORT
  name "cylinderinner.geo"
  meshsize 10e-6
  rotate
  ( 0, -90, 0 )
  translate
  ( 200E-6, 0, 0 )
  scale 1
  unit  1
  $PHYSICAL
    name  "Innershield"
material "Mumetal"
  $END
$END
$IMPORT
  name "cylinderouter.geo"
  meshsize 15e-6
  rotate
  ( 0, -90, 0 )
  translate
  ( 220E-6, 0, 0 )
  scale 1
  unit  1
  $PHYSICAL
    name "Outershield"
material "Mumetal"
  $END
$END
$STRUCT
  Name squareloop
  $PATH
    Layer M0
Width %w
    XY
    ( -%a+0.6*%w, -%a  )
    ( %a, -%a  )
    ( %a,  %a )
    ( -%a, %a )
    ( -%a, -%a+0.6*%w  )
  $END
$END
$STRUCT loops
  $SREF
    Name squareloop
XY
( -%d, 0 )
  $END
  $SREF
    Name squareloop
XY
( 0, 0 )
  $END
  $SREF
    Name squareloop
XY
( %d, 0 )
  $END
  $PATH
    Layer TERM
    Text  "P1+ M0"
    XY
    (-%d-%a+0.6*%w, -%a-%w/2)
    (-%d-%a+0.6*%w, -%a+%w/2)
  $END
  $PATH
    Layer TERM
    Text  "P1- M0"
    XY
    (-%d-%a-%w/2, -%a+0.6*%w)
    (-%d-%a+%w/2, -%a+0.6*%w)
  $END
  $PATH
    Layer TERM
    Text  "P2+ M0"
    XY
    ( -%a+0.6*%w, -%a-%w/2 )
    ( -%a+0.6*%w, -%a+%w/2 )
  $END
  $PATH
    Layer TERM
    Text  "P2- M0"
    XY
    ( -%a-%w/2, -%a+0.6*%w )
    ( -%a+%w/2, -%a+0.6*%w )
  $END
  $PATH
    Layer TERM
    Text  "P3+ M0"
    XY
    (%d-%a+0.6*%w, -%a-%w/2)
    (%d-%a+0.6*%w, -%a+%w/2)
  $END
  $PATH
    Layer TERM
    Text  "P3- M0"
    XY
    (%d-%a-%w/2, -%a+0.6*%w)
    (%d-%a+%w/2, -%a+0.6*%w)
  $END
$END
```

The netlist for the three loops contains mutual coupling.

■ Netlist 17.2.5 — loop.cir.

```
* Title: Netlist for InductEx user manual
L1      1    0
L2      2    0
L3      3    0
M12     L1   L2   1e-6
M23     L2   L3   1e-6
M13     L1   L3   1e-6
P1      1    0
P2      2    0
P3      3    0
.end
```

The entire parameter string has been buried in the IXI file, so that *InductEx* is executed from the command prompt simply with:

```
inductex loops.ixi
```

The calculation output is shown below. *InductEx* calculates the loop inductances and mutual coupling by exciting the voltage ports over each loop, and then calculates coupling from the external field by applying 1 T in each axial direction and determining the current induced in every loop. For the first loop, the self-inductance of the first (unshielded) loop is about 117.4 pH. The x and y-directed magnetic fields have very low coupling to the coil, but the z-directed field is perpendicular to the loop surface and thus couples strongly. In the presence of the nearby shield and the mutual coupling from the other loops, 10.93 A is excited in the first loop by the 1 T z-directed field (close to the theoretical value for the total flux through the loop area in the absence of kinetic inductance). *InductEx* then calculates the mutual inductance required to produce this current in the first loop, and finally calculates the coupling factor from a 1 H inductance that models the external field if it is excited by 1 A to represent 1 T. This coupling factor, listed in the results as KFZ0, can be used in SPICE simulations to model external field coupling to the loop. A discussion is presented in [24].

It is also clear from the results that the field coupling to the second loop, located at the open end of the shield, is significantly lower than that to the first loop, while the coupling to the third (shielded) loop is nearly two orders of magnitude below that to first loop.

Command Prompt

```
Inductance [H]
Name      Design      Extracted     AbsDiff      PercDiff
L1        --          1.17299E-10   +1.173E-10   --%
L2        --          1.23387E-10   +1.2339E-10  --%
L3        --          1.29066E-10   +1.2907E-10  --%

Mutual Inductance [H]                                        Coupling factor
Name      Design      Extracted     AbsDiff      PercDiff           k
M12       +1E-06      -7.0832E-14   -1E-06       -100%       -5.888E-4
M23       +1E-06      +7.1823E-13   -1E-06       -100%       +5.691E-3
M13       +1E-06      +4.7347E-13   -1E-06       -100%       +3.848E-3


External magnetic field coupling (x-directed):
                                   Mutual Inductance [H]     Coupling factor
KFX0     LFX   L1                       -3.4176E-12             -3.156E-07
KFX1     LFX   L2                       -6.7855E-12             -6.109E-07
KFX2     LFX   L3                       -1.0806E-11             -9.511E-07

External magnetic field coupling (y-directed):
                                   Mutual Inductance [H]     Coupling factor
KFY0     LFY   L1                       -4.2622E-12             -3.935E-07
KFY1     LFY   L2                       -9.4370E-12             -8.496E-07
KFY2     LFY   L3                        6.4016E-12              5.635E-07

External magnetic field coupling (z-directed):
                                   Mutual Inductance [H]     Coupling factor
KFZ0     LFZ   L1                        1.2832E-09              1.185E-04
KFZ1     LFZ   L2                        2.0715E-10              1.865E-05
KFZ2     LFZ   L3                       -1.6647E-11             -1.465E-06
```

The magnitude of the magnetic flux density as calculated by *InductEx* is shown in Fig. 17.5 and Fig. 17.6. The plots were generated with ParaView from the flux density over the plane as printed to (`B_loops.vtk`).

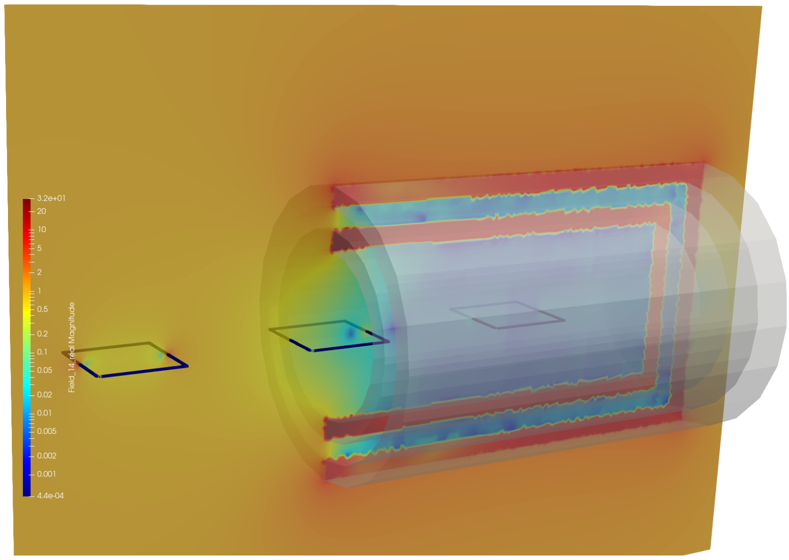


Figure 17.5: Magnetic flux density magnitude around shield and loops in a 1 Tesla z-directed ambient flux density.

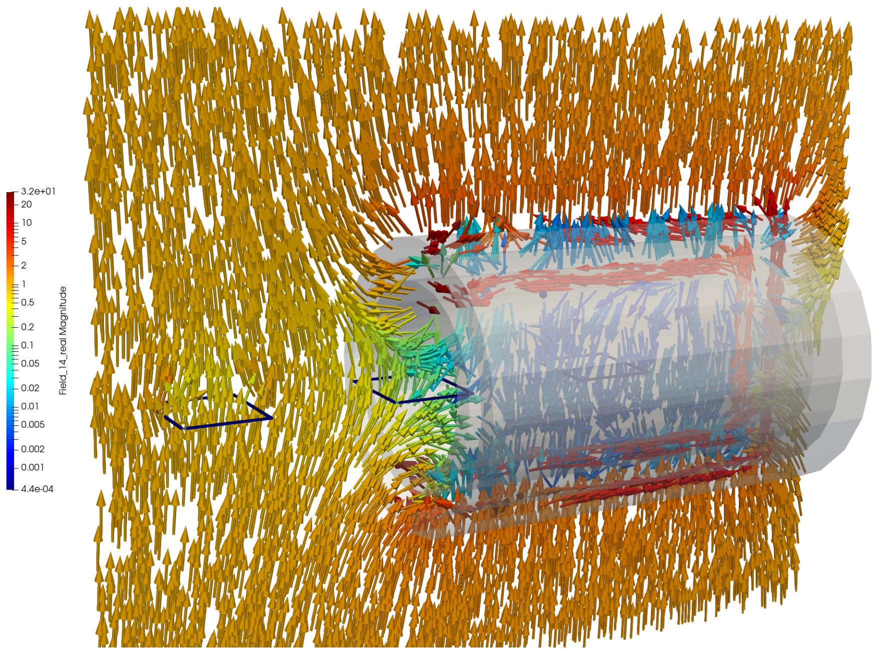
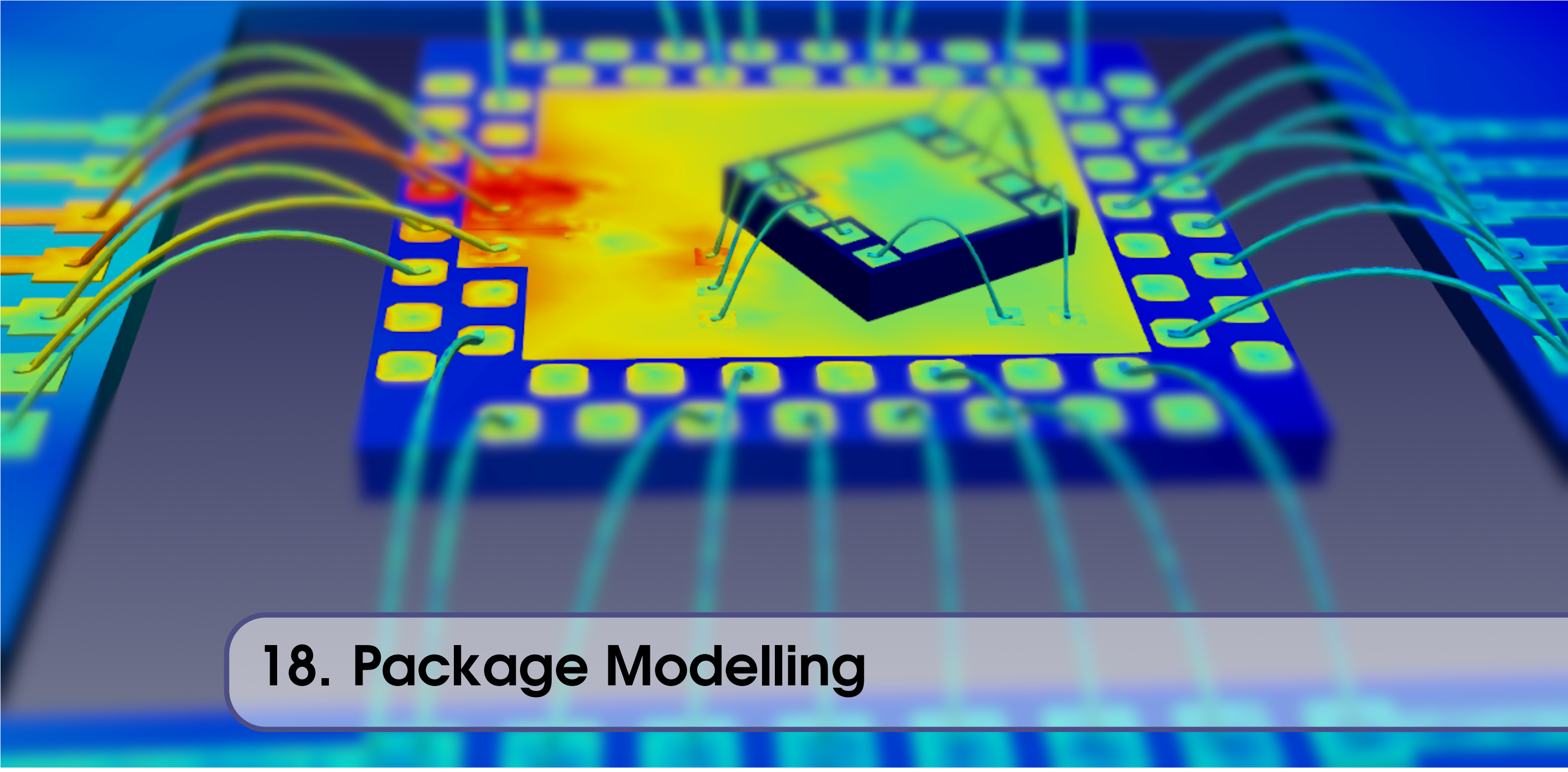


Figure 17.6: Magnetic flux density vectors around shield and loops in a 1 Tesla z-directed ambient flux density represented by glyphs.



18. Package Modelling

18.1 Chip-to-chip wire bonds

Example works only with Super licence tier

This example, available in `examples/package/wedge/` and `examples/package/ballstitch/`, demonstrates:

- wire bond definition.
- multiple chip layout import.
- wedge-type wire bonds.
- ball-and-stitch wire bonds.

For this example, we find the total inductance between the ports on two separate chips that have signal and ground pads connected through wire bonds. The two chip dies use the same process, defined in Layer Definition File 18.1.1, although it is possible to assign a different process to every die.

The first chip is defined in IXI File 18.1.2. Bond locations are identified by text labels that start with "#". Chip 2 has a similar IXI file, but with the signal and ground pins swapped.

■ **Layer Definition File 18.1.1 — process1.ldf.**

```
$Parameters
  Units            =  1e-6
  SegmentSize      =  5.0
  AbsMin           =  0.025
  GPLayer          =  1
  TermLayer        =  19
  TextLayer        =  18
$End
* ----------- M0 ------------
$Layer
  Number      =      17
  Name        =      M0
  Thickness   =      0.2
  Lambda      =      0.09
  Order       =      0
  Mask        =      1
  Filmtype    =      S
$End
* ----------- IO ------------
$Layer
  Number      =      2
  Name        =      IO
  Thickness   =      0.15
  Order       =      1
  Mask        =     -1
  Filmtype    =      I
$End
* ----------- M1 ------------
$Layer
  Number      =      3
  Name        =      M1
  Thickness   =      0.25
  Lambda      =      0.09
  Order       =      2
  Mask        =      1
  Filmtype    =      S
$End
```

■ IXI File 18.1.2 — chip1.ixi.

```
$STRUCT
  Name "Pad"
  $POLY
    Layer M1
    XY
    ( -%pw/2 ,  %pw/2 )
    ( -%pw/2 , -%pw/2 )
    (  %pw/2 , -%pw/2 )
    (  %PW/2 ,  %PW/2 )
    ( -%PW/2 ,  %PW/2 )
  $END
$END
$STRUCT
  Name "chipmain"
  $POLY
    Layer M0
    XY
    ( -%cg/2 ,  %cg/2 )
    ( -%cg/2 , -%cg/2 )
    (  %cg/2 , -%cg/2 )
    (  %cg/2 ,  %cg/2 )
    ( -%cg/2 ,  %cg/2 )
  $END
  $SREF
    Name Pad
      Reflect False
      Angle   0
      XY
      (-%cg/2 + %PtE + %pw/2, -%PadSep/2 - %pw/2)
  $END
  $SREF
    Name Pad
      Reflect False
      Angle   0
      XY
      (-%cg/2+%PtE+%pw/2, %PadSep/2+%pw/2)
  $END
  $SREF
    Name Pad
      Reflect False
      Angle   0
      XY
      (%cg/2-%PtE-%pw/2, -%PadSep/2-%pw/2)
  $END
  $SREF
    Name   Pad
      Reflect False
      Angle   0
      XY
      (%cg/2-%PtE-%pw/2, %PadSep/2+%pw/2)
  $END
  $SREF
    Name   Pad
      Reflect False
      Angle   0
      XY
      ( -%PadSep/2-%pw/2, -%cg/2+%PtE+%pw/2)
  $END
  $SREF
    Name   Pad
      Reflect False
      Angle   0
      XY
      ( %PadSep/2+%pw/2, -%cg/2+%PtE+%pw/2 )
  $END
  $SREF
    Name   Pad
      Reflect False
      Angle   0
      XY
      ( -%PadSep/2-%pw/2, %cg/2-%PtE-%pw/2 )
  $END
  $SREF
    Name   Pad
      Reflect False
      Angle   0
      XY
      ( %PadSep/2+%pw/2, %cg/2-%PtE-%pw/2 )
  $END
  $POLY
    Layer I0 // Via from pad to ground
    XY
(%cg/2-%PtE-%pw/2-%pw/6,%PadSep/2+%pw/2+%pw/6)
(%cg/2-%PtE-%pw/2-%pw/6,%PadSep/2+%pw/2-%pw/6)
(%cg/2-%PtE-%pw/2+%pw/6,%PadSep/2+%pw/2-%pw/6)
(%cg/2-%PtE-%pw/2+%pw/6,%PadSep/2+%pw/2+%pw/6)
(%cg/2-%PtE-%pw/2-%pw/6,%PadSep/2+%pw/2+%pw/6)
  $END
  $PATH
    Layer M1
Width 1.0
    XY
    ( 0.0             , -%PadSep/2-%pw/2 )
    ( %cg/2-%PtE-%pw/2, -%PadSep/2-%pw/2 )
  $END
  $PATH
    Layer term
    Text  "P1 m1 m0"
    XY
    ( 0.0 , -%PadSep/2 - %pw/2 + 0.5 )
    ( 0.0 , -%PadSep/2 - %pw/2 - 0.5 )
  $END
  $TEXT
    Layer TEXT
    Text "#Pad1" // "#" marks a bond location
    xy
    ( %cg/2-%PtE-%pw/2, -%PadSep/2-%pw/2 )
  $END
  $TEXT
    Layer TEXT
    Text "#Pad_Two"
    xy
    ( %cg/2-%PtE-%pw/2, %PadSep/2+%pw/2 )
  $END
$END
```

The two chips are imported and the wire bonds are defined in IXI File 18.1.4. Chip 2 is rotated by 200° around the z-axis to show wire bonds tracking the bond locations. It is also translated upwards so that the die surfaces are not on the same level.

■ **Netlist 18.1.3 — netlist.cir.**

```
* Title: Netlist for InductEx user manual
Linterchip  1  2  0.55n
P1          1  0  // Port on chip1
P2          2  0  // Port on chip2
.end
```

■ **IXI File 18.1.4 — wedge-type.ixi.**

```
$PARAMSTRING
  -l process1.ldf
  -i wedge-type.geo
  -th
  -n netlist.cir
$END
$DIE
  name    "chip1"
  layout  "chip1.ixi"
  process "process1.ldf"
  rotate
  ( 0, 0, 0 )
  translate
  ( 0, 0, 0 )
$END
$DIE
  name    "Chip_Two"
  layout  "chip2.ixi"
  process "process1.ldf"
  rotate
  ( 0, 0, 200 )
  translate
  ( 600e-6, 0, 20e-6 )
$END
$WEDGE
  Name Wedge1
  FootLength 42e-6
  FootHeight 3e-6
  FootWidth   15e-6
  FootChamfer 12e-6
  ToeLength 3e-6
  ToeHeight 3e-6
  Throat 6e-6
  MeshSize  10e-6
$END
$WIREBONDSETUP
  Name      WB1
  StartType "wedge"
  EndType   "wedge"
  StartBond "Wedge1"
  EndBond   "wedge1"
  Material  "Gold"
  LengthSections 12
  MeshSize    10e-6
  VertexHeight 100e-6
  Diameter     10e-6
$END
$MATERIAL
  Name     "Gold"
  Sigma    4.1E9
  Lambda   0.0
$END
$WIREBOND
  BondSetup  WB1
  StartPin  "chip1.Pad1"
  EndPin  "chip_Two.Pad_Two"
$END
$WIREBOND
  BondSetup  WB1
  StartPin  "chip1.Pad_Two"
  EndPin  "chip_Two.Pad1"
$END
```

The netlist is shown in Netlist 18.1.3. *InductEx* is executed with:

```
inductex wedge-type.ixi
```

The resulting output is shown below:

Command Prompt

```
Inductance [H]
Name         Design        Extracted      AbsDiff       PercDiff
LINTERCHIP   5E-10         4.94348E-10    -5.652E-12    -1.1304%
```

The calculation model is shown in Fig. 18.1.

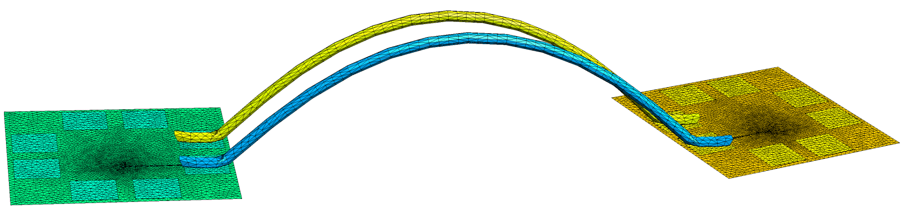


Figure 18.1: Wedge-type wire bonds between two dies.

Ball-and-stitch wire bonds are demonstrated with the same chip layouts and process file, but the alterations to the project setup file as shown in IXI File 18.1.5

IXI File 18.1.5 — ball-stitch.ixi.

```
$PARAMSTRING
  -l process1.ldf
  -i ball-stitch.geo
  -th
  -n netlist.cir
$END
$DIE
  name    "chip1"
  layout  "chip1.ixi"
  process "process1.ldf"
  rotate
  ( 0, 0, 0 )
  translate
  ( 0, 0, 0 )
$END
$DIE
  name    "Chip_Two"
  layout  "chip2.ixi"
  process "process1.ldf"
  rotate
  ( 0, 0, 180 )
  translate
  ( 400e-6, 0, -100e-6 )
$END
$WEDGE
  Name Stitch1
  FootLength 42e-6
  FootHeight 3e-6
  FootWidth   15e-6
  FootChamfer 12e-6
  ToeLength 3e-6
  ToeHeight 3e-6
  Throat 6e-6
  MeshSize  10e-6
$END
$BALL
  Name Ball1
  PadDiameter 20e-6
  BallHeight 25e-6
  BallDiameter 35e-6
  BallSegments 7
  MeshSize  5e-6
$END
$WIREBONDSETUP
  Name      WB1
  StartType "ball"
  EndType   "wedge"
  StartBond "ball1"
  EndBond   "stitch1"
  Material  "Gold"
  LengthSections 5
  MeshSize       10e-6
  VertexHeight   100e-6
  Diameter       10e-6
  TopLength      50e-6
  BendRadius     40e-6
$END
$MATERIAL
  Name     "Gold"
  Sigma    4.1E9
  Lambda   0.0
$END
$WIREBOND
  BondSetup  WB1
  StartPin   "chip1.Pad1"
  EndPin   "chip_Two.Pad_Two"
$END
$WIREBOND
  BondSetup  WB1
  StartPin   "chip1.Pad_Two"
  EndPin   "chip_Two.Pad1"
$END
```

The calculation model is shown in Fig. 18.2.

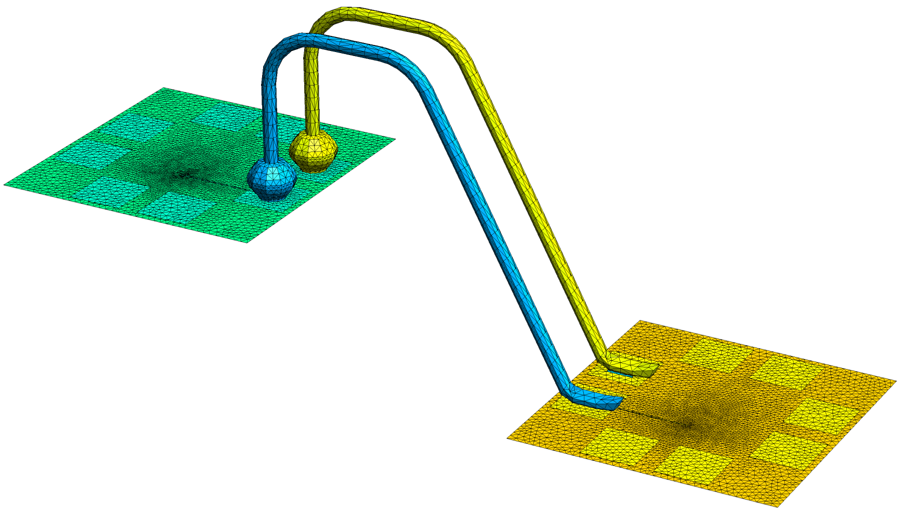


Figure 18.2: Ball-and-stitch wire bonds between two dies.

18.2 Chip-to-chip solder bumps

Example works only with Super licence tier

This example, available in `examples/package/bump/`, demonstrates:

- flip-chip setup.
- solder bump definition.

For this example, we find the total inductance between the ports on two separate chips, flipped together, that have signal and ground pads connected through solder bumps. The two chip dies use the same process (defined in Layer Definition File 18.1.1) used in Section 19.1. The same chip IXI files are used as in Section 19.1.

The two chips are imported and the solder bump bonds are defined in IXI File 18.2.1. Chip 2 is rotated by 180° around the y-axis to flip it, and by another 180° around the z-axis. It is translated 30 µm upwards in the z-direction to create space for the bump bonds. Two bump setups are configured; one to be convex and the other to be concave.

IXI File 18.2.1 — bump-bond.ixi.

```
$PARAMSTRING
  -l process1.ldf
  -i bump-bond.geo
  -th
  -n netlist.cir
$END
$DIE
  name    "chip1"
  layout  "chip1.ixi"
  process "process1.ldf"
  rotate
  ( 0, 0, 0 )
  translate
  ( 0, 0, 0 )
$END

$DIE
  name    "Chip_Two"
  layout  "chip2.ixi"
  process "process1.ldf"
  rotate
  ( 0, 180, 180 )
  translate
  ( 0, 0, 30e-6 )
$END

$BUMPSETUP
  Name       BumpConvex
  Material   "Gold"
  LengthSections         5
  CircumferenceSections  20
  MeshSize               15e-6
  PadDiameter            30e-6
  Diameter               38e-6
$END
$BUMPSETUP
  Name       BumpConcave
  Material   "Gold"
  LengthSections         7
  CircumferenceSections  20
  MeshSize               15e-6
  PadDiameter            30e-6
  Diameter               25e-6
$END
$MATERIAL
  Name      "Gold"
  Sigma     4.1E9
  Lambda    0.0
$END
$BUMP
  BondSetup  BumpConcave
  StartPin   "chip1.Pad1"
  EndPin     "chip_Two.Pad_Two"
$END
$BUMP
  BondSetup  BumpConvex
  StartPin   "chip1.Pad_Two"
  EndPin     "chip_Two.Pad1"
$END
$BUMP
  BondSetup  BumpConvex
  StartPin   "chip1.Pad3"
  EndPin     "chip_Two.Pad8"
$END
$BUMP
  BondSetup  BumpConvex
  StartPin   "chip1.Pad4"
  EndPin     "chip_Two.Pad7"
$END
$BUMP
  BondSetup  BumpConvex
  StartPin   "chip1.Pad5"
  EndPin     "chip_Two.Pad6"
$END
$BUMP
  BondSetup  BumpConvex
  StartPin   "chip1.Pad6"
  EndPin     "chip_Two.Pad5"
$END
$BUMP
  BondSetup  BumpConvex
  StartPin   "chip1.Pad7"
  EndPin     "chip_Two.Pad4"
$END
$BUMP
  BondSetup  BumpConvex
  StartPin   "chip1.Pad8"
  EndPin     "chip_Two.Pad3"
$END
```

The calculation model built by *InductEx* is shown in Fig. 18.3.

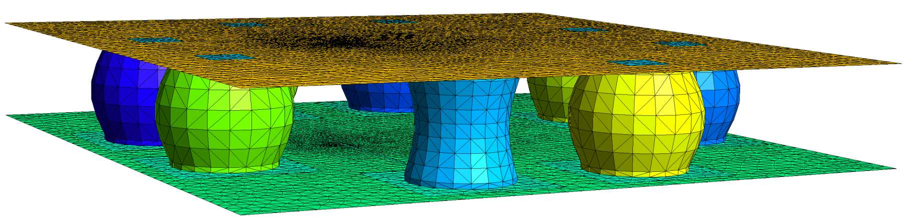
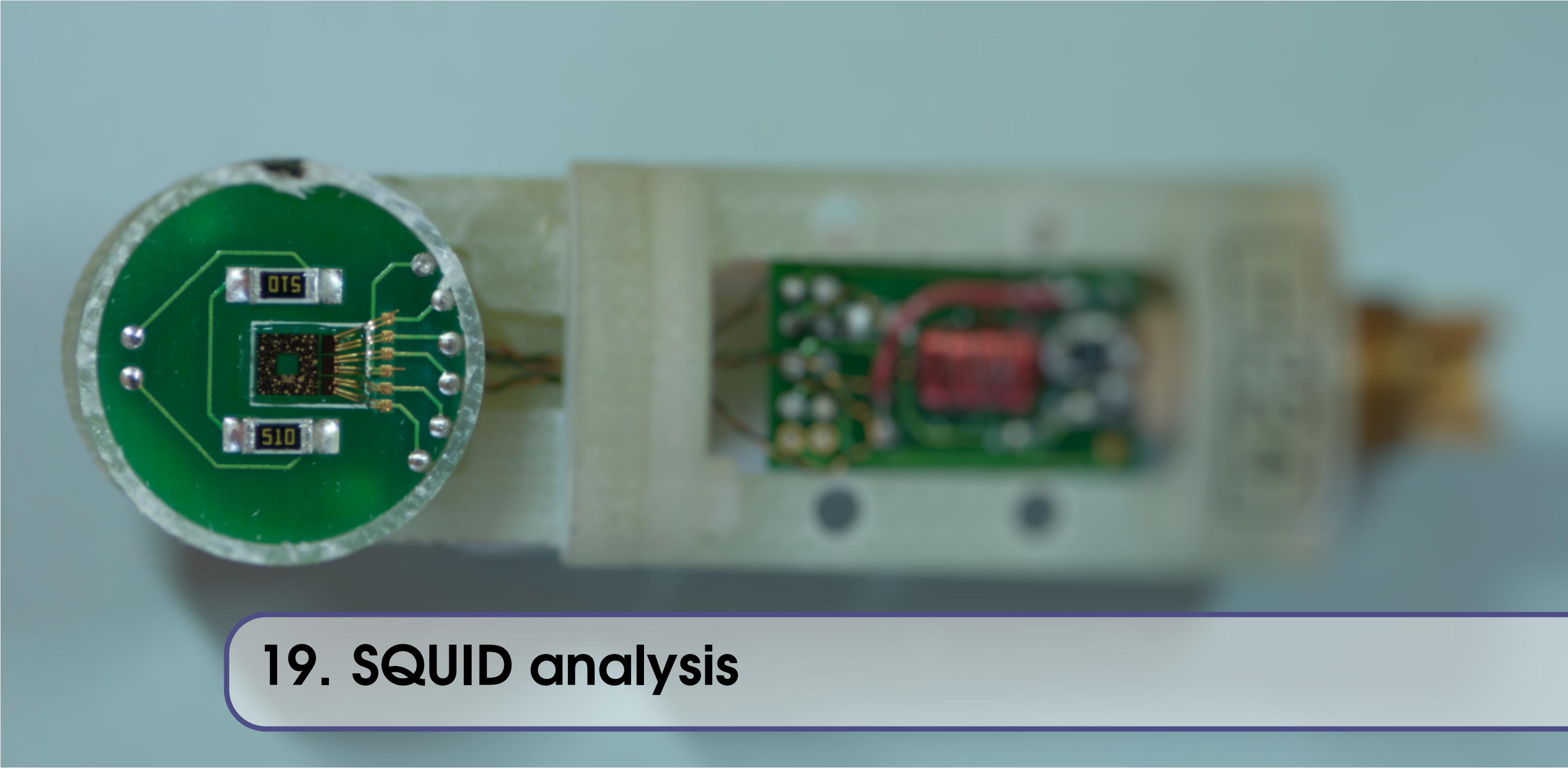


Figure 18.3: Solder bump bonds between flipped chips.



19. SQUID analysis

19.1 Single-turn direct-coupled SQUID

Example works only with Super licence tier

This example, available in `examples/squid/singleturn_directcoupled/`, demonstrates:

- mesh masking.
- back-annotation.
- JoSIM simulation.
- effective area calculation.
- frequency.

The M2700 SQUID from Star Cryoelectronics is modelled through geometry entry in an IXI file. For more information on the SQUID, see [25] and [26].

We are interested in finding two characteristic values: the field sensitivity in nT/Φ_0 and the feedback coil coupling in μA/Φ_0.

The SQUID is described as an integrated circuit in the IXI file, with an associated process. Process parameters are estimated to arrive at the layer definition file `process.ldf` as shown in Layer Definition File 19.1.1.

■ Layer Definition File 19.1.1 — m2700s.ldf.

```
$Parameters
  Units                  =  1e-6
  Frequency              =  1
  SegmentSize            =  50
  AbsMin                 =  0.1
  ProcessHasGroundPlane = FALSE
  TermLayer          =  19
  TextLayer          =  18
$End
$Layer
  Number      =      1
  Name        =      YBCO
  Thickness   =      0.2
  Lambda      =      0.24
  Order       =      0
  Mask        =      1
  Filmtype    =      S
$End
```

.

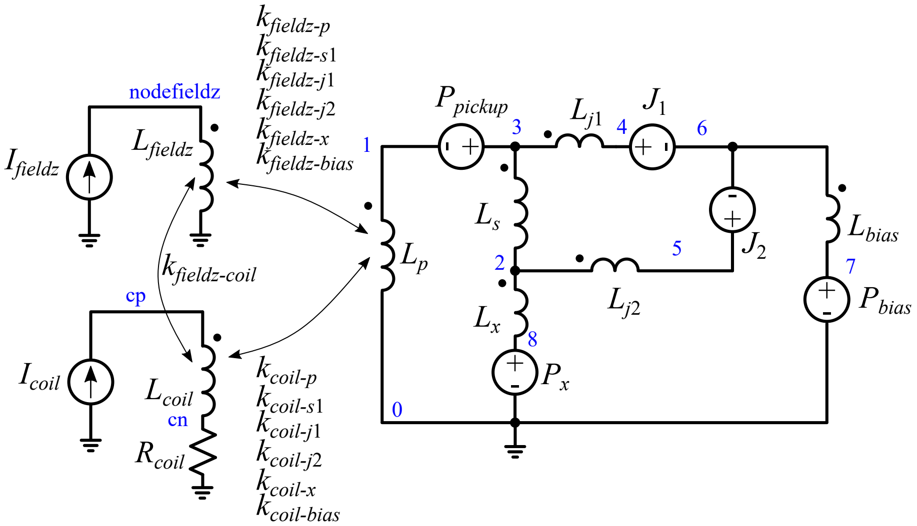


Figure 19.1: Full circuit model of the M2700 direct-coupled SQUID with coupling from an external coil and an external magnetic field.

■ **Netlist 19.1.2 — m2700.cir.**

```
* Title: Netlist for InductEx user manual - M2700
.control
  BACK-ANNOTATE m2700full_sim.cir
  set Rcond     = 1e-14
  set AbsTol    = 1e-6
  set Tol       = 1e-6
  set Frequency = 1
.endc
Lp        1   0    10n  [lp]
Ls1       3   2    80p  [ls1]
LJ1       3   4    1p   [lj1]
LJ2       2   5    1p   [lj2]
Lx        2   8    1p   [lx]
Lbias     6   7    10p  [lbias]
Lcoil     cp  cn   36u  [lcoil]
Rcoil     cn  0    1    [rcoil]
kcoil_p     Lcoil  Lp 0.1    [kcoil_p]
kcoil_s1    Lcoil  Ls1 0.1   [kcoil_s1]
kcoil_j1    Lcoil  Lj1 0.1   [kcoil_j1]
kcoil_j2    Lcoil  Lj2 0.1   [kcoil_j2]
kcoil_x     Lcoil  Lx 0.1    [kcoil_x]
kcoil_bias  LCoil  Lbias 0.1 [kcoil_bias]
J1        4   6
J2        5   6
Pbias     7   0
Ppickup   3   1
Px        8   0
p1        cp  0
.end
```

The circuit model of the SQUID and the feedback coil, including the z-directed external magnetic field component to be extracted by InductEx, is shown in Fig. 19.1. The netlist without the magnetic field component is shown in Netlist 19.1.2. The coil is copper and has resistivity. A resistor is thus included in the coil circuit, and coupling from the coil inductor to every inductor in the SQUID circuit is added. All inductors are marked for back-annotation to the inductors named in square brackets in the JoSIM simulation file `m2700_sim.cir`.

Simulation frequency is set to 1 Hz because the SQUID is used for low-frequency measurements. This is done in the LDF file, but is better done in the control block of the circuit netlist, as is also shown here. The control block includes three additional parameters that are set due to the very large difference in inductances between the coil and the SQUID structure:

- **Rcond** sets the threshold below which singular value decomposition values are considered to be zero. The default is 1×10^{-14}, but can be lowered if the ratio of coil inductance to SQUID inductances is excessive.
- **AbsTol** Sets the absolute tolerance of the GMRES solution step in TetraHenry. The default is 1×10^{-4}, but smaller values yield more precise results for weak mutual inductance.
- **Tol** Sets the minimum relative tolerance of the GMRES solution step in TetraHenry. The default is 1×10^{-4}, but smaller values yield more precise results for weak mutual inductance.

For this analysis example, the feedback coil outer dimension is set to 3.96 mm, the length to 2.0 mm, and the depth of the coil below the SQUID die plane as 2.46 mm. The wire diameter is 76 µm. the coil has 175 turns, which is modelled here with 10 up-down wrapping layers along the length of the coil.

A Gmsh model has been built separately and included as `coil175_10wraps_2mm.geo` with this example. The coil is imported into the layout, and assigned the material parameters of copper, with assumed bulk conductivity of 58.7 S/m, as shown in the redacted excerpt in IXI File 19.1.3

Due to the size of the SQUID pickup loop, the global segment size is set to 50 µm in the LDF file. The SQUID is too small for such large segments, so that a mask is defined to enforce a mesh size of 2 µm inside a defined polygon.

IXI File 19.1.3 — m2700.ixi.

```
PARAMSTRING
  -l m2700.ldf -i m2700.geo -th -n m2700.cir
  -b  // External magnetic field analysis
$END
...
$VARIABLE
  Name  LOOPSIDE  // Pickup loop side length
  Value 2700
$END
...
$STRUCT
  Name Main
///
...
  // Definition of SQUID layout
...
///
$END
$MATERIAL
  Name     "Copper"
  Sigma    58.7E6
  Lambda   0.0
$END
```

```
$IMPORT
  name "coil175_10wraps_2mm.geo"
  rotate
  ( 0, -90, 0 )
  translate
  ( -(%LOOPSIDE/2)*1e-6, 0, -4.460e-3 )
  scale 1
  unit  1
  terminal "P1+" T1
  terminal P1-  T2
  $PHYSICAL
    name     "coil"
    material "Copper"
    meshsize 1e-3
  $END
$END
$MASK
  Name CLMask1
  MeshSize 2
  XY
   ...
$END
```

Execute *InductEx* with:

```
inductex m2700.ixi
```

Figure 19.2: InductEx model of the M2700 SQUID with feedback coil in its approximate position.

The extraction results are back-annotated to a simulation netlist for JoSIM or a similarly capable superconductor circuit simulator. The back-annotated file (slightly redacted) is shown in Netlist 19.1.4. The current source `IFieldz` has been edited to create a 100 nA sweep from 5 ps to 1 ns.

The circuit is then simulated in JoSIM to obtain the voltage and current plots vs time shown in Fig. 19.3.

Netlist 19.1.4 — m2700_sim_annotated.cir.

```
* Back-annotated simulation file
B1      4   6    jj1  area=1
B2      5   6    jj1  area=1
LJ1     3   4    1.355E-011
LJ2     2   5    6.187E-012
Lx      2   0    1.833E-011
Lbias   6   7    7.129E-011
Rbias   7   17   1
IBias   0   17   pwl(0 0 5p 25u)

LS1     3   2    6.418E-011
LP      3   0    1.939E-009

Lcoil   cp  cn   5.161E-005
Rcoil   cn  0    7.273
Icoil   0   cp   pwl(0 0 1.5n 0 2.5n 50u)

kcoil_p    Lcoil Lp    0.02185
kcoil_s1   Lcoil Ls1   -2.006E-007
kcoil_j1   Lcoil Lj1   -0.02527
kcoil_j2   Lcoil Lj2   -0.03703
kcoil_x    Lcoil Lx    -0.1276
kcoil_bias Lcoil Lbias -0.05326

.model jj1 jj(rtype=1, vg=100uV,
+ cap=20fF, rn=14ohm, icrit=0.01mA)

.tran 0.25p 2500p 0 0.25p

.plot v(6)
.plot i(lcoil)
.plot i(lfieldz)
* -----------------------------
*  External field compact model
* -----------------------------
* Field current sources. Ampere
* to Tesla ratio is 1:1
* External magnetic field coupling (x-directed):
IFieldx 0 NodeFieldx pwl(0 0 5p 0)
LFieldx NodeFieldx 0   1
KFX0      LFIELDX   LP       9.63024520860869E-9
KFX1      LFIELDX   LS1     -4.77142424631055E-9
KFX2      LFIELDX   LJ1     -1.04822299427506E-7
KFX3      LFIELDX   LJ2     -1.08996998665198E-7
KFX4      LFIELDX   LX       5.43941375054776E-8
KFX5      LFIELDX   LBIAS   -7.78020974918814E-8
KFX6      LFIELDX   LCOIL    6.74390931298265E-8
* External magnetic field coupling (y-directed):
IFieldy 0 NodeFieldy pwl(0 0 5p 0)
LFieldy NodeFieldy 0   1
KFY0      LFIELDY   LP       3.43838856895961E-10
KFY1      LFIELDY   LS1     -4.92778375894472E-9
KFY2      LFIELDY   LJ1      6.61258500003104E-9
KFY3      LFIELDY   LJ2     -6.99677584941501E-9
KFY4      LFIELDY   LX      -5.15517794955348E-9
KFY5      LFIELDY   LBIAS    8.21165800277486E-10
KFY6      LFIELDY   LCOIL    0.000318408366975167
* External magnetic field coupling (z-directed):
IFieldz 0 NodeFieldz pwl(0 0 5p 0 1n 100n)
LFieldz NodeFieldz 0   1
KFZ0      LFIELDZ   LP       0.0276518737224822
KFZ1      LFIELDZ   LS1     -0.0757384957482001
KFZ2      LFIELDZ   LJ1     -0.165947153449997
KFZ3      LFIELDZ   LJ2     -0.000493849621129741
KFZ4      LFIELDZ   LX      -0.141420506822723
KFZ5      LFIELDZ   LBIAS   -0.0724813879493786
KFZ6      LFIELDZ   LCOIL    0.170646405626656
.END
.END
*========================
```

The InductEx extractions obtain a feedback coil inductance of 51.6 μH and a resistance of 7.27 Ω at 1 Hz. This is then taken as equivalent to the dc values.

From the simulation results in Fig. 19.3, the feedback coil coupling is obtained as 5.9 $\mu A/\Phi_0$, while the z-directed magnetic field sensitivity is read off as 25.9 nT/Φ_0.

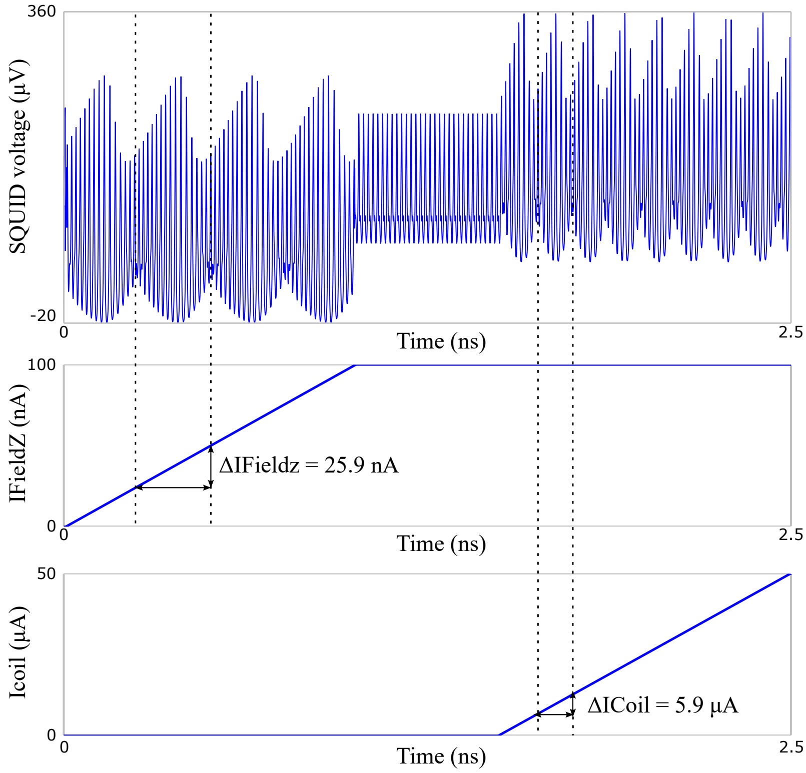


Figure 19.3: JoSIM simulation results on extracted and back-annotated SQUID circuit with feedback coil and external field excitation currents.

Bibliography

[1] C. J. Fourie, O. Wetzstein, T. Ortlepp, and J. Kunert, "Three-dimensional multi-terminal superconductive integrated circuit inductance extraction," *Superconductor Science and Technology*, vol. 24, no. 12, p. 125015, Nov 2011.

[2] C. J. Fourie and W. J. Perold, "Simulated inductance variations in rsfq circuit structures," *IEEE Transactions on Applied Superconductivity*, vol. 15, no. 2, pp. 300–303, June 2005.

[3] C. J. Fourie, O. Wetzstein, J. Kunert, H. Toepfer, and H.-G. Meyer, "Experimentally verified inductance extraction and parameter study for superconductive integrated circuit wires crossing ground plane holes," *Superconductor Science and Technology*, vol. 26, no. 1, p. 015016, Nov.

[4] C. J. Fourie, O. Wetzstein, J. Kunert, and H. Meyer, "Sfq circuits with ground plane hole-assisted inductive coupling designed with inductex," *IEEE Transactions on Applied Superconductivity*, vol. 23, no. 3, p. 1300705, June 2013.

[5] C. J. Fourie, "Full-gate verification of superconducting integrated circuit layouts with inductex," *IEEE Transactions on Applied Superconductivity*, vol. 25, no. 1, pp. 1–9, Feb 2015.

[6] K. Jackman and C. J. Fourie, "Flux trapping analysis in superconducting circuits," *IEEE Transactions on Applied Superconductivity*, vol. 27, no. 4, pp. 1–5, June 2017.

[7] K. Jackman and C. J. Fourie, "Flux trapping experiments to verify simulation models," *Superconductor Science and Technology*, vol. 33, no. 10, p. 105001, aug 2020. [Online]. Available: https://doi.org/10.1088/1361-6668/aba79b

[8] C. J. Fourie and K. Jackman, "Experimental verification of moat design and flux trapping analysis," *IEEE Transactions on Applied Superconductivity*, vol. 31, no. 5, pp. 1–7, 2021.

[9] K. Jackman and C. J. Fourie, "Tetrahedral modeling method for inductance extraction of complex 3-d superconducting structures," *IEEE Transactions on Applied Superconductivity*, vol. 26, no. 3, pp. 1–5, April 2016.

[10] C. Geuzaine and J.-F. Remacle, “Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities,” *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.

[11] S. Nagasawa, T. Satoh, K. Hinode, Y. Kitagawa, M. Hidaka, H. Akaike, A. Fujimaki, K. Takagi, N. Takagi, and N. Yoshikawa, “New nb multi-layer fabrication process for large-scale sfq circuits,” *Physica C: Superconductivity*, vol. 469, no. 15, pp. 1578 – 1584, 2009, proceedings of the 21st International Symposium on Superconductivity (ISS 2008).

[12] W. H. Chang, “The inductance of a superconducting strip transmission line,” *Journal of Applied Physics*, vol. 50, pp. 8129–8134, Dec. 1979.

[13] J. Jaycox and M. Ketchen, “Planar coupling scheme for ultra low noise dc squids,” *IEEE Transactions on Magnetics*, vol. 17, no. 1, pp. 400–403, January 1981.

[14] S. Kempf, A. Ferring, A. Fleischmann, and C. Enss, “Direct-current superconducting quantum interference devices for the readout of metallic magnetic calorimeters,” *Superconductor Science and Technology*, vol. 28, no. 4, p. 045008, feb 2015. [Online]. Available: https://doi.org/10.1088/0953-2048/28/4/045008

[15] J. A. Delport, K. Jackman, P. l. Roux, and C. J. Fourie, “Josim—superconductor spice simulator,” *IEEE Transactions on Applied Superconductivity*, vol. 29, no. 5, pp. 1–5, Aug 2019.

[16] S. V. Polonsky, V. K. Semenov, and P. N. Shevchenko, “PSCAN: personal superconductor circuit analyser,” *Superconductor Science and Technology*, vol. 4, no. 11, pp. 667–670, nov 1991.

[17] N. Takeuchi, Y. Yamanashi, and N. Yoshikawa, “Adiabatic quantum-flux-parametron cell library adopting minimalist design,” *Journal of Applied Physics*, vol. 117, no. 17, p. 173912, 2015.

[18] S. K. Tolpygo, V. Bolkhovsky, T. J. Weir, A. Wynn, D. E. Oates, L. M. Johnson, and M. A. Gouker, “Advanced fabrication processes for superconducting very large-scale integrated circuits,” *IEEE Transactions on Applied Superconductivity*, vol. 26, no. 3, pp. 1–10, April 2016.

[19] C. J. Fourie, C. L. Ayala, L. Schindler, T. Tanaka, and N. Yoshikawa, “Design and characterization of track routing architecture for rsfq and aqfp circuits in a multilayer process,” *IEEE Transactions on Applied Superconductivity*, vol. 30, no. 6, pp. 1–9, 2020.

[20] J.-S. Hong and M. J. Lancaster, *Microstrip filters for RF/Microwave applications*. John Wiley and Sons, Inc., 2001.

[21] J. Jayabalan, B.-L. Ooi, M.-S. Leong, and M. Iyer, “Novel circuit model for three-dimensional geometries with multilayer dielectrics,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 4, pp. 1331–1339, 2006.

[22] H. R. Mohebbi, O. W. B. Benningshof, I. A. J. Taminiau, G. Miao, and D. G. Cory, “Superconducting coplanar interdigital filter with robust packaging,” *IEEE Transactions on Applied Superconductivity*, vol. 25, no. 3, pp. 1–4, 2015.

[23] M. Misakian, “Equations for the magnetic field produced by one or more rectangular loops of wire in the same plane,” *Journal of Research of the National Institute of Standards and Technology*, vol. 105, no. 4, pp. 557–564, 2000.

[24] C. J. Fourie and K. Jackman, “Software tools for flux trapping and magnetic field analysis in superconducting circuits,” *IEEE Transactions on Applied Superconductivity*, vol. 29, no. 5, pp. 1–4, Aug 2019.

[25] L. P. Lee, J. Longo, V. Vinetskiy, and R. Cantor, “Low-noise $YBa_2Cu_3O_{7-delta}$ direct-current superconducting quantum interference device magnetometer with direct signal injection,” *Applied Physics Letters*, vol. 66, no. 12, pp. 1539–1541, Mar. 1995.

[26] R. Cantor, L. Lee, M. Teepe, V. Vinetskiy, and J. Longo, “Low-noise, single-layer $YBa_2Cu_3O_{7-x}$ DC SQUID magnetometers at 77 K,” *IEEE Transactions on Applied Superconductivity*, vol. 5, no. 2, pp. 2927–2930, 1995.

Index