3D-MLSI

The program for extraction of 3D inductances of multilayer superconductor circuits

M. M. Khapaev,
Moscow State University,
Dept. of Computational Mathematics and Cybernetics,
Chair of Numerical Methods,
vmhap@cs.msu.su
http://www.cmc.msu.ru/vm/sotr/vmhap

This manual has been written by M. Khapaev and E. Goldobin (gold@uni-tuebingen.de)

April 24, 2012

Contents

1 Introduction **2**

2 The purpose of 3D-MLSI **2**
2.1 Application area . . . 2
2.2 Physical definitions . . . 2
2.3 Numerical technique . . . 3
2.4 Restrictions . . . 4
2.5 Further plans . . . 4

3 3D-MLSI components and files **4**
3.1 **3D-MLSI** tasks flow . . . 4
3.2 **UPM** . . . 4
3.3 **MLW** . . . 5
3.4 Predefined equivalent circuit inductances assigning using **MAPLE** 5
3.5 **D2D** . . . 6

4 System requirements and installation **6**

5 Native input data file commands **6**
5.1 Units . . . 6
5.2 Input file commands list . . . 6
5.3 Command *ell* . . . 9
5.4 Drawing geometry using polylines . . . 10
5.5 Command *tp* . . . 11
5.6 Terminals with non constant current density . . . 11

6 Automatic generation of equivalent circuits and terminal paths **12**

7 *DXC* input **12**
7.1 *DXC* format . . . 12
7.2 **D2D** converter . . . 13
7.3 **D2D** *DXC* file restrictions . . . 13
7.4 *def.thn* file . . . 14
7.5 *TRM* file . . . 14

8 Output **14**
8.1 Export of the magnetic field . . . 15
8.2 **CalcH**: calculating magnetic field and related quantities . . . 15

9 Student version **17**

1 Introduction

Inductance estimation is an essential step in the design of the superconducting devices and circuits. An up-to-date and detailed review of available tools for this problem is found in Survey of Inductance Estimation Tools [1]. Except for a well developed and most frequently used two dimensional program **LMETER** [3] the majority of the tools are developed for structures with three dimensional (3D) distribution of the current and magnetic field. Up to now, applications of these 3D tools in circuit design have been very challenging due to the difficulties related to the input data representation and time consuming computations.

The idea behind the program is to allow the user to define as general structures as he needs perhaps with certain compromise with automation of input data preparation.

In addition to the extraction of inductances from the layout the **3D-MLSI** allows to calculate the current distribution in thin superconducting films. The excitation currents can be either injected via special terminals or induced by the magnetic flux trapped in the holes of the films or induced by external magnetic field.

2 The purpose of 3D-MLSI

2.1 Application area

The **3D-MLSI** can simulate both high- and low-T_c superconducting structures composed using parallel thin superconducting films. It is possible to :

- simulate self and mutual inductances for currents circulating around the holes and for variety of terminal-to-terminal or terminal chain currents;
- simulate and visualize the current and magnetic field distribution in (just above) the films.

2.2 Physical definitions

The package is intended for studying the currents in planar conducting layers separated by layers of dielectric. Let t_m be the thickness of conducting layers and d_k be the thickness of dielectric layers, k, m - the numbers of the layers. Conducting layers can contain few metal patches of arbitrary shape. Patches are needed for re-definition London penetration depth and heights of a piece of conductor.

Let the number of conductors in all layers be N_c and the total number of holes in all conductors will be N_h. Each conductor can have current terminals and holes.

In the package it is assumed that

$$t_m \ll l, \quad \lambda_m \sim t_m, \tag{1}$$

where l is the typical lateral size of the film (the size of it's most important part for example) in the x–y plane, λ_m is the London penetration depth.

Each conductor has the projection on the (x, y) plane. The boundary of this projection is the subject of input. The position of the conductor on z axe

is $[h_m^0, h_m^1]$, $m = 1, \ldots, N_c$. We assume that all current terminals are on the external boundary of the conductors. This limitation will be improved in future versions of the program.

For inductances calculation the magnetic field is excited by currents circulating around holes and currents through chains of terminals on the conductors. Let N_t be the number of these terminal chains and N_h the total number of holes in all conductors. Thus the total number $N = N_t + N_h$ of excitation currents is the dimension of inductance matrix.

2.3 Numerical technique

The details can be found in [4, 5]. Shortly the technique is the following:

- Start from steady London and Maxwell equations;
- Assume the planarity of superconductor sheets;
- To simplify equations using planarity: current is 2D sheet but magnetic field full 3D;
- To present current using potential representation called stream function;
- To rewrite the problem in terms of stream function;
- To make some tricks in order to approximate small but finite thickness of superconductor films;
- To apply for numerical solution Finite Element Method;
- Varying boundary conditions and right parts of equations obtain the necessary solutions;
- To calculate if necessary inductances via calculating full energy;
- Make postprocessing of inductance matrix in order to avoid holes containing zero fluxoids;
- To present sheet current in form of colored maps and streamlines;
- If necessary assign inductances in predefined equivalent circuit using calculated inductances. Use for this purpose the program for analytical calculations **MAPLE** [8].

The main advantage of equations of stream function technique is very clear problem definition ideally suitable for Finite Element Method solution. The stream function equations by sense and in part by solution are practically similar to well-known Poisson equations and boundary problems for this equation.

Let us pay attention on the fact that excitation currents including terminal and hole circulating currents are taken into account as simple boundary conditions.

2.4 Restrictions

1. The terminals can be placed only at the external boundary of the conductors;

These restrictions are only technical and can be improved in the future versions of the **3D-MLSI**.

2.5 Further plans

- *CIF* and *DXC* input;
- Improving the restrictions pointed above;
- Certain improving in numerical technique;
- Output similar to well-known program **LMETER** (for compatibility);
- New circuits simulators oriented output;
- Predefined equivalent circuit extraction from **MAPLE** to executable;
- New automatic equivalent circuit generation.

3 3D-MLSI components and files

3.1 3D-MLSI tasks flow

The package consists of a pre-processor part **UPM**, a numerical core **MLW**, a tool for extraction of predefined equivalent circuit inductances and optional preprocessor part **D2D**.

Under WINDOWS all components of **3D-MLSI** are driven by the graphical shell **WPM**.

All programs except eqivalent circuit parameters extraction (exists as **MAPLE** script) create log files for events and errors tracing.

3.2 UPM

The program **UPM** is the preprocessor. **UPM** needs an input file *name.dat* specified as the first argument in the command line. Optionally, the file containing the terminal paths for currents can be specified as the second argument of the command line. If the second argument is omitted, **UPM** tries to read the terminal path from the file *name.tp*. As a last resort when the task is to calculate the inductances, the terminal paths are generated automatically. See section 6 for details.

As a result of successful execution **UPM** creates two files: *name.upm* which contains various data and *name.trg* which contains triangular mesh.

3.3 MLW

MLW is a numerical core. As input, **MLW** needs two files: *name.upm* and *name.trg*. As output, when executed successfully **MLW** creates the files *name.psi* (solution) and *name.out* (inductances or other specific data). For problem 4 the program creates optional file *name.l* with the data obtained from locking principle.

If the inductances are used for the circuit simulation and circuit simulator allows matrix component equations, the result can be presented in the form suitable for circuit simulation program. It means:

- Equivalent circuit nodes are associated with terminals and have the numbers of these terminals;
- In the output are present pairs of node numbers and terminal-to-terminal self inductances for these nodes (as *inductor* in *SPICE* [7]);
- For each pair of terminal-to-terminal defined by four node numbers mutual inductance (in contrast with coupling coefficient in *SPICE*) is printed.

3.4 Predefined equivalent circuit inductances assigning using MAPLE

Calculated inductances can be used for approximate assigning of the values in a predefined equivalent circuit.

Extraction of the equivalent circuit inductances is based on the following algorithm:

1. Consider the $v \times l$ connectivity matrix A of the equivalent circuit of the layout with v vertices and l branches.
2. From matrix A, evaluate $\sigma \times l$ matrix P of physically relevant fundamental current loops in the circuit. For planar equivalent circuits $\sigma = l - v + 1$.
3. Consider the symbolic $l \times v$ sparse matrix Q. The portrait of Q coincide with portrait of P^T. The non-zero values of Q are unknown values of netlist inductances.
4. Consider the $\sigma \times \sigma$ matrix I of excitation currents.
5. For fundamental loops, calculate matrix of "physical" inductances L.
6. Taking into account the equality of fluxoids, $P \cdot Q \cdot I = L \cdot I$ and as result we obtain matrix equation
$$P \cdot Q = L \tag{2}$$
with σ^2 equations and l unknowns.
7. Solve (2) using the least squares method [8, 9] for netlist inductances.

The equivalent circuit and the fundamental loops have to be chosen properly. Fundamental loops have to contain "physically" adequate flux trapped in them.

To apply this technique one needs to use the system of analytical calculations **MAPLE**. The **MAPLE** program needs the file *name.out* produced by **MLW** and netlist connectivity matrix. As a result netlist inductances are assigned.

This approach needs analysis of circuit work as well as equivalent circuit topology parsing. Topology analysis is the necessary step for other known programs (**LM2CIR**, [10]) and needs some manual work.

3.5 D2D

The program **D2D** is an optional converter for some high-T_c applications. It converts *DXC* layout data to native **3D-MLSI** input format. This program is limited to specific configurations and will be substituted in future by powerful and general preprocessor.

For details see the section 7 "*DXC* input".

4 System requirements and installation

The inductance calculation part is platform independent. It can be compiled for MS WINDOWS or for different versions of UNIX.

Under WINDOWS all parts of the package except eqivalent circuit parameters extraction are driven by the graphical shell **WPM**. The **WPM** is also intended for visualization of the results such as current distribution.

The installation is simple. Copy all executable files and *dxcview.pal* in a directory of your choice and set the shortcut to **WPM**.

Batch use of **UPM** and **MLW** needs setting the *PATH* variable.

Equivalent circuit parameters assigning script needs **MAPLE**.

5 Native input data file commands

5.1 Units

If the length, width and thickness of the layout elements, as well as such constants like λ (London penetration depth) is given in μm, the calculated inductance has dimensions of pH. The flux trapped in the holes should be integer and specifies the number of flux quanta Φ_0 trapped in the hole. The terminal currents should be specified in mA. The magnetic field is given in "natural" units of mA/μm (rather than A/m).

In the case when the typical length is different then μm it is necessary to present all coordinates in dimensionless units taking a typical length as a unit of length, so that the input data does not contain extremely large or small values. To obtain the right result, it is necessary to multiply the calculated inductances by the unit of length. By other words, the inductance scales proportionally to the physical (linear) dimension of the structure.

If the coordinates are presented in certain CAD units, it is possible to convert them into necessary units automatically during file processing.

5.2 Input file commands list

The input file (*DAT* file) is a text file which you can create using a text editor. Each line is either a command or a comment. The commands have the form

```
command_name arg arg .... arg [ arg arg
```

where *arg* is a number or a string or a character. The following rules apply:

- The commands are case sensitive.
- The delimiters between the arguments are space ' ', equal sign '=', comma ',' or tabulation.
- The order of commands (except special cases) is not important. But the last of similar commands (except *ell*) overrides preceding.
- Unknown or erroneous commands are ignored without warning.
- Many commands have a default value.
- The empty strings in the input file are ignored. The double space at the beginning of a line denotes a comment.

In this manual the arguments after (or in) square bracket '[' are optional. The commands are:

cc *text string* Descriptive name of the project (will be passed through all task flow);

rem, REM ; Comment (ignored);

nc=*value* Number of conductors;

pb= *id* Problem to solve:

0 Calculate the inductance matrix assuming no trapped flux in the holes. Default.

1 Calculate the full inductance matrix for the holes and the terminal paths.

2 Calculate the current distribution for a given flux $\Phi = n\Phi_0$ trapped in the holes, for given external magnetic field $\mathbf{H}_n$, and for given currents along the terminal paths. The value of n is given using `hc i n`. The value of $\mathbf{H}_n$ is given using `bn=`$\mathbf{H}_n$. **WPM** can be used to visualize the solution.

3 Calculate the current distribution for a given total currents around the holes (???), given external magnetic field $\mathbf{H}_n$, and given currents along the terminal paths. **WPM** can be used to visualize the solution.
User sets field, currents and fluxoids. The fluxes in the holes are arbitrary. ???//Ed

4 Calculate boundary distributed inductances using locking principle.

The difference between the problem 2 and 3 is the way how **3D-MLSI** interprets the arguments of the command **hc** (see below).

bn=*value* External magnetic field $\mathbf{H}_n$ (or its z - component) normal to the surface of all superconducting films. The default value is 0. Makes sense only for **pb**=2 and 3. $\mathbf{H}_n$ is given in units of mA/μm.

ah=*value* Global finite elements resolution step. The default value is 1.0. The position of this command in the file is important! If another **ah** command is present later in the input file, it will overwrite the previous value.

tol=*value* Tolerance for input points (see command **ell**). In **UPM** (but not **WPM**) the points which are closer than the tolerance are treated as coinciding. The default value is 2×10^{-7}.

lmbd=*value* Global for all nets London penetration depth λ (μm).

grds=*value* Scaling factor to convert CAD units to physical units. For example if one CAD unit is equal to 1 inch, use $grds = 25400$ to convert to μm. The default value is 1.0 and all data will be in μm.

avg=*on/off* Conductors finite thickness switch (default is on). If 'on' programs assume the conductors are of finite thickness. In this case commands **ms** are ignored. If 'off" conductors are presented by infinitely thin current sheets. The position (height) of sheets within conductor is then defined by command **ms**.

tb=*on/off* Is valid for **pb=2,3** only. If "on" magnetic field on terminals is calculated. Output for this command is in file *name.out*. The output format is a set of lines as

$$n \quad s \quad x \quad y \qquad H_n$$

Here n is terminal number, H_n is lateral, normal (internal normal) to terminal component of magnetic field an (x, y) is the point on terminal where magnetic field was calculated. Value s is non-dimensional parameter (from -1 to 1) for the terminal. This parameter can be used for graphics drawing. If terminal is composed from more then one boundary elements for each boundary element self parameter is used.

mr=*on/off* Use/Don't use mirror image technique for current distribution calculation in the presence of the ground plane. The default value is "off".

out=*0/1* 0 — inductances matrix is in a matrix form (for eqivalent circuit parameters extraction for example); 1 — the matrix is in a circuit oriented form, like SPICE. The output format of each line of output is

$$m \quad n \quad L_{mn} \quad t_{m0} \quad t_{m1} \quad t_{n0} \quad t_{n1}$$

Here n, m are indices of terminal paths (not necessary for SPICE data), L_{mn} is the inductance (self or mutual), $t_{m0} \quad t_{m1} \quad t_{n0} \quad t_{n1}$ are indices of the terminals which are used to inject or drain the currents. These indices of the terminals are easily connected with the SPICE input. The default value for this command is 0.

hc i n Sets the number of flux quanta n trapped in the hole number i (for **pb**=2) or the current circulating around the hole number i (for **pb**=3). The default n for all i is equal to 0.

Example: Calculate which field will create Φ_0 in a hole of the radius $R = 20\,\mu$m? $H = \frac{n\Phi_0}{\pi R^2 \mu_0} \approx 1.311 \times 10^{-3}\ \frac{\text{mA}}{\mu\text{m}}$.

tmpt T [T_c [p Define the temperature dependence of London penetration depth:

$$\lambda(T) = \frac{\lambda_0}{\sqrt{1-(T/T_c)^p}}$$

where default values are $T = 0$, $T_c = 1.0$ and $p = 2$.

cond n h_0 h_1 [λ [ah Default material parameters of conductor number n: h_0 is the height of the lower surface, h_1 is the height of the upper surface, λ is the London penetration depth (overwrites global λ), and ah is mesh size (overwrites global mesh size).

net Similar to **cond**.

patch *net* n h_0 h_1 [λ Part n of net *net* with material properties different from default. Default patch has number 0 and is defined by command *cond* (*net*). Last parameter is optional. When it is omitted default value is used.

lockd m n_1 n_2 ... n_m Locked domains list (locking principle).

lockdb k n_1 n_2 ... n_k Locked boundaries list (locking principle). Not similar to **lockd**. Single boundary can contain multiple domains.

biaspath m n_1 ... n_m List of numbers of bias currents terminal pathes. Locking principle, problem 4.

ms ***l s*** Sets the position (along z-axis) of the infinitely thin sheet current flowing in the conductor patch number l. The position of the sheet current is given by the following expression $z = h_0 + s(h_1 - h_0)$. The value of s should be from 0 (bottom surface) to 1 (upper surface). This command works if **avg**=off.

usetriangle Use known program Triangle for fast triangulation. In this case, possibly commands **pointinhole** and **pointinpatch** should be used. Global ah or local ah for conductors is used for mesh generation. Boundary mesh size is approximately equal to ah. The program try to make internal triangles slightly large keeping the triangulation quality (no very sharp edges, avoid if possible obtuse triangles).

pointinhole *nc np x y* Triangle needs help to identify holes. *nc* is the number of conductor. (x, y) is a point inside a hole.

pointipatch *nc np x y* Triangle needs this input to identify patches. (x, y) is a point inside patch *np* for conductor *nc*.

5.3 Command *ell*

This is the main command. The command defines a part (a line) of the conductor boundary. The command has the form

```
ell k form g1 g2 g3 g4 [g5] [type [id
```

The lines which describe the conductor boundaries are oriented. The orientation ("from" point and "to" point) should be chosen so that the conductor stays on the left side of the boundary, when we move from the starting point to the ending point of a line.

Here is the definition of the arguments of the command:

k The index of conductor;

form The form of the conductor part:

- **0** Straight segment (part of a strait line);
- **1** Arc defined by center, radius, starting and ending angles;
- **2** Arc defined by 3 points;
- **3** Circle;

g1 g2 g3 g4 g5 Real values. Depends of **form**:

- **form=*0*** The segment with origin $(g1, g2)$ and end $(g3, g4)$.
- **form=*1*** Arc with center $(g1, g2)$, radius $g3$, and given by angles from $g3$ to $g4$.
- **form=*2*** Three-point arc with first point $(g1, g2)$, second point $(g3, g4)$, and last point $(g5, g6)$.
- **form=*3*** Circle with center $(g1, g2)$, radius $g3$, anti-clockwise order for positive g4 and clockwise in other case.

type Character. Define the type of the component part:

- **b** Impenetrable edge. Default;
- **h** Hole with number *id*;
- **t** Terminal with number *id*;
- **i** Interface element. Next parameters are n_{left} n_{right} [r . The parameters are left patch number, right patch number and optionally reverse sign for locking principle.
- **id** Number of a hole or a terminal.

5.4 Drawing geometry using polylines

Using `ell k 0 ...` (straight segments) has a drawback that one has to specify the coordinates of each node twice — ones in the current ell command and the second time in the next ell command. To avoid this repetition and make input/editing easier the following polyline commands are added.

```
polyh nc nh [x1 y1] [x2 y2] ...
polyb nc [x1  y1] t1 [x2 y2] [x3 y3] t4 [x4 y4] ...
```

where `polyh` defines a boundary of the hole, while `polyb` defines the usual boundary with optional terminals. Each polyline is automatically closed **EG: What about terminal on the closing segment?**. The square brakets are not?? essential and can be substituted by round barces or skept???. In the above example, the terminal t1 corresponds to the segnment from [x1 y1] to [x2 y2]. The other parameters are defined as usual: `nc` is a conductor number, `nh` is a hole number `t`n is a terminal number.

5.5 Command *tp*

This command sets the current paths for the terminal currents. In general, the paths can not be closed. The command has the form

```
tp [J=value] i1->i2 i3->i4 ... im->in
```

where $i1, i2, \ldots, im, in$ - the terminal numbers. One tp-command sets one current path. Use several tp-commands to set all required paths. The terminals with the numbers connected by symbols "`->`" should belong to the same conductor. It means that the symbol "`->`" defines the current path within certain conductor for given current path. This command is very flexible. It allows to specify very complex current flow configurations.

Statement $J = value$ is valid for pb=2 or 3 only (no inductances, but current patterns are calculated) and define the full current within the current path.

5.6 Terminals with non constant current density

It is possible to specify J_n profile across a terminal.

It is assumed that terminal is presented by single boundary element. To set the profile we assume the boundary element is parametrically presented using parameter x, $-1 \leq x \leq 1$. Then full current J across a terminal is given as

$$\int_{\Gamma} J_n ds = J. \tag{3}$$

The profile is given by function $f(x)$ in formula

$$J_n = \alpha f(x) J \tag{4}$$

where α is normalizing coefficient. If

$$\int_{-1}^{1} f(x) dx = 1 \tag{5}$$

then $\alpha = 1$. This coefficient is calculated by the program automatically.

First it is necessary to define a current profile by command

```
termprofile n m {other data}
```

Here n is the number (label) of profile and m is the type of profile. $m = 0$ is default type and means constant distribution. No other parameters are necessary in this case. $m = 1$ is simple formula $f(x) = a_0 + a_1 x + a_2 x^2$ where a_0, a_1, a_2 are specified as "other data".

Second it is necessary to assign a current profile to a terminal. It is done by command

```
attprofile termnumber profilenumber
```

6 Automatic generation of equivalent circuits and terminal paths

The **3D-MLSI** allows to generate automatically a simple natural equivalent circuit and assign inductance values for the given layout. The equivalent circuit for each conductor is the circuit in which each terminal is connected to all other terminals of this conductor. All possible self and mutual inductances are calculated and all screening currents are taken into account. As a result, the matrix of self an mutual inductances contains a complete set of data for a given layout. If terminals present nodes of netlist then the matrix can be directly stamped into circuit simulator.

If no terminal paths commands are found the necessary data for *pb*=0,1,2 is generated automatically. The command **out**=1 can be used to represent the inductance matrix in a form close to the SPICE format. It can be used in circuit simulator which allows matrix component equations.

7 *DXC* input

7.1 *DXC* format

DXC format is reduced AutoCAD's *DXF* format. It was developed for the popular program **LMETER** [3]. The author of *DXC* writes:

```
DESCRIPTION

This is a simple flat geometry description format to specify a set
of lines and polylines attached to different layers.

FILE FORMAT

$ENTITIES
$POLYLINE
layerName
termNumber
x0 y0
x1 y1
...
xn yn
$POLYLINE
...
$EOF

NOTES

As many spaces/tabs/newlines may be used as necessary.

Layer names should be declared in technology description file (see
lmeter(5)). They may not start from the dollar sign (it marks
```

```
keywords).

While each polyline is assigned an integer number, this number has
meaning only for terminals (that are drawn using layer TERM). If
several polylines in layer TERM share the same number they are
considered to be different parts of one terminal.

The majority of polylines in designs are closed and they should
have the last point xn yx the same as the first one. Terminals are
built from open polylines sometimes (usually when a terminal is
just a straight line on the end of a microstrip line), in this
case they should not have the first point repeated at the end.

While coordinates may be floating point numbers, they should be
integer multiples of the declared gridsize (see lmeter(5)).

BUGS

This format looks ugly in general... ;-)

SEE ALSO

lmeter(1), lmeter(5).

AUTHOR

Paul Bunyk
```

7.2 D2D converter

For compatibility we have a program **D2D** for conversion of *DXC* data to our own **3D-MLSI** format, but some manual work still is necessary.

The input files for **D2D** are: *name.dxc*, *name.trm* and *def.thn*.

The file *def.thn* contains description of the layers (e.g. the description of the fabrication process). The **D2D** looks for this file according to the rules of the operational system and opens it automatically.

The file *name.dxc* (or just *name*) has to be specified in the command line as the first argument. This file contains the photo-mask information in *DXC* format.

The file *name.trm* contains information about the terminals. It can be specified as the second argument in the command line. If it is missing, the **D2D** looks for the file with the same name as *DXC* file but with extension "*trm*".

If **D2D** executes successfully, the file **name.dat** is created. This file contains all geometrical and material data, but does not contain information about the current flow paths. The heights of the conductors (in commands *cond...*) have to be checked and corrected manually, if necessary.

7.3 D2D *DXC* file restrictions

DXC drawing have to be prepared as follows:

- One polyline for one single connected conductor. Really it is not a restriction: polylines can be merged in schematic editor.
- The terminals can be represented by a single segments only (polylines with one segment). This is specific for the high-T_c problems **D2D** was developed.

In future **D2D** will be substituted by other preprocessor.

7.4 *def.thn* file

THN file contains the following commands (case sensitive):

```
physical_layer name who thickness [lambda [flag
```

and

```
CAD_layer name invert_flag phys_name_1 ... phys_name_n
```

Here the `name` is the user defined name of physical or CAD layer. The parameter `who` is a character which can be either 's' (superconductor) or 'i' (insulator). The `thickness` of the layer is given in μm, *lambda* is the London penetration depth. If `invert_flag` = 1 (default) then this physical layer always exists. If `invert_flag` = 0 then this physical layer is present only if it is drawn in some CAD layer. If `invert_flag` = 1 this CAD layer is inverted and not inverted in case `invert_flag` = 0. `phys_name_1` ... `phys_name_n` are physical members of CAD layer.

The order of physical layers is given by order of `physical_layer` commands. The first command means the lowest layer.

The data presented in the technology definition file is treated in the program **D2D** by simple but not general algorithm. As a result the heights of conductors in a DAT file need to be checked and corrected, if necessary.

7.5 *TRM* file

Since *DXC* file does not contain information about the layers where the terminals are, this information is presented in *TRM* file. *TRM* file contains the (case sensitive) strings of the following form:

```
terminal n CAD_layer_name
```

where n is terminal number.

8 Output

3D-MLSI is solving a specified problem in terms of the so-called stream function $\psi(x, y)$ in each conductor and shows the result in **WPM** as flowing currents. The currents are related to the stream function as

$$J_x = \frac{\partial \psi}{\partial y}, \quad J_y = -\frac{\partial \psi}{\partial x}. \tag{6}$$

When looking at solution in **WPM** the status line shows the values of currents J_x, J_y, $|J|$, its direction and the stream function ψ at the point under the mouse pointer.

8.1 Export of the magnetic field

One can export the values of magnetic field in horizontal and vertical planes. This is done using toolbar buttons "H1" (field in a plane parallel to the plane of the structure) and "H2" (field in plane perpendicular to the plane of the structure).

For H1 export, one should specify the extents of the export grid $x_{\min}$, $x_{\max}$, $y_{\min}$, $y_{\max}$, the number of divisions N and M in the grid in x and y directions, as well as the height (level) h of the export plane. The resulting output matrix will have the sizes $(N+1) \times (M+1)$. The format of output file is

```
x y Hx Hy Hz |H|
```

One can also check "center field point". In this case the field values will be exported in the centers of the grid cells rather than at the nodes and the size of the output array will be $N \times M$. This is useful when the hole size (e.g. 5x5) is known.

The calculation of the field is rather time consuming and can last about a minute for 50×50 matrix. When the export is complete you will see a messge box, which also reports the total flux through the exported area. If some points of the export grid are inside conductor, the field cannot be calculated. In this case, the export file will contain "*" instead of field values for corresponding points, and the flux will be calculated only through the area outside conductors.

8.2 CalcH: calculating magnetic field and related quantities

CalcH is a command-line utility for (batch) calculations of magnetic field, vector potential and fluxes (not fluxoids) from already obtained solution. This means that you should first run **WPM**, **UPM** and **MLW** to obtain the triangulation (*.trg*) and the solution (*.psi*) files.

The **CalcH** command line is:

```
CalcH -n nm [-Hpl HplInfile] [-Hpt HptInfile] [-Hz HzInfile]
            [-Apt AinFile]   [-Phi FluxInfile] [other options]
```

The options are described below. One of the options: `-Hpl`, `-Hpt`, `-Hz`, `-Apt`, `-Phi` must be present, otherwise **CalcH** will have nothing to do. If more than one option is present, **CalcH** will do all the specified requests. **EG: Misha, why do you use mandatory parameter nm as a switch? make it non-switch**

-n nm Obligatory option, gives the name (path, no extension) to **3D-MLSI** input. The input must be in the same directory with *nm.trg* and *nm.psi* files.

-Hpl HplInfile Asks to calculate magnetic field in plane(s) parallel to plane of all conductors for one or several rectangular patches. *HplInfile* is the input file with the description of the patches. The default name is *Hp.in*. The file can contain any number of lines of the form

```
X1 X2 Y1 Y2 Z NX NY
```

and $H(x, y, z)$ will be calculated in $(NX + 1) \times (NY + 1)$ equally spaced points on planar $[X1, X2] \times [Y1, Y2]$ patch situated at height Z. The output corresponding to k-th patch is written in a file *HplInfile.#k.dat*. Format of each output file is

```
x y z Hx Hy Hz |H|
```

If the magnetic field can't be calculated at some points due to singularity or because the point is inside a conductor, the output file will contain a '*' symbol instead of the field values.

-Hpt HptInfile Asks to calculate magnetic field in a set of points with coordinates given in *HptInfile*. The default name for *HptInfile* is *Hpt.in*. The file may contain any number of lines of the form

```
x y z
```

An output will be written in a file *HptInfile.dat* in the following format

```
x y z Hx Hy Hz |H|
```

If the magnetic field can't be calculated at some points due to singularity or because the point is inside a conductor, the output file will contain a '*' symbol instead of the field values.

-Hz HzInfile Asks to calculate magnetic field in plane(s) perpendicular to conductors within one or several rectangular patches. *HzInfile* is the input file with the patches. The default name is *Hz.in*. The file can contain any number of lines of the form

```
X1 X2 Y1 Y2 Z1 Z2 N1 N2
```

and $H(x, y, z)$ will be calculated in $(NX + 1) \times (NY + 1)$ equally spaced points on planar patch with diagonal line $[X1, Y1, Z1]$ to $[X2, Y2, Z2]$. The output corresponding to the k-th patch is written to the file *HzInfile.#k.dat*. Format of the output file is

```
x y z Hx Hy Hz |H|
```

If the magnetic field can't be calculated at some points due to singularity or because the point is inside a conductor, the output file will contain a '*' symbol instead of the field values.

-Apt Aptinfile Asks to calculate vector potential in a set of points with the coordinates given in *AptInfile*. The default name for *AptInfile* is *Apt.in*. The file may contain any number of lines with the following format

```
x y z
```

The output is written in a file *AptInfile.dat* with lines formatted as follows

```
x y z Ax Ay Az |A|
```

-Phi FluxInfile Asks to calculate magnetic field flux across the surface(s) spanned on closed polylines with coordinates given in *FluxInfile*. The default name for *FluxInfile* is *Phi.in*. The file may contain any number of polylines (each input line describes one polyline)

```
[x0 y0 x0] [x1 y1 z1] [x2 y2 z2] ..... [xn yn zn]
```

where $n + 1$ is the number of corners. The output is written in a file *AptInfile.dat* with the value of flux across corresponding polyline in each line, i.e.,

```
Phi_1
Phi_2
....
Phi_k
```

where k is the number of polylines. The first and the last points of each polyline should be different. The polyline will be closed automatically.

-h, -help Short help

-sl Forces *avg*=off.

-centered For *-Hpl* field will be calculated in $NX \times NY$ center grid points.

-sn k Take solution number k for calculations. *nm.psi* file can contain several solutions. By default, $k = 0$ is used for calculations.

-ah v Take v as mesh step for vector potential contour integration. By default this value is taken as *ahb*. Relevant only when `-Phi` is specified. **EG: Collides with -h for help!**

9 Student version

Student version has some limitations. The number of conductors $nc \leq 2$, there can be only few **tp**-commands, and only few holes.

References

[1] K. Gaj, Q. P. Herr, V. Adler, A. Krasniewski, E. G. Friedman, M. J. Feldman, "Tools for the Computer-Aided Design of Multigigahertz Superconducting Digital Circuits," *IEEE Trans. Appl. Superconductivity*, vol. 9, pp. 18–38, March 1999.

[2] T. Van Duzer and C. W. Turner, *Principles of Superconductive Devices and Circuits*, North Holland: Elsevier, 1981.

[3] *http://pavel.physics.sunysb.edu/RSFQ/RSFQ.html*

[4] M. Khapaev, "Inductance Extraction of Multilayer Finite Thickness Superconductor Circuits," *IEEE Trans. Microwave Theory and Techn.*, January 2001.

[5] M. M. Khapaev, "Extraction of inductances of plane thin film superconducting circuits," *Supercond. Sci. Technol.*, vol. 10, pp. 389–394, 1997.

[6] P. P. Silvester and R. L. Ferrari, *Finite Elements for Electrical Engineers.* Cambrige University Press, 1983.

[7] P. Antognetti, G. Massobrio, "Semiconductor Device Modeling with SPICE," McGraw-Hill, Second Edition, 1993.

[8] http://www.maplesoft.com

[9] C. L. Lawson, R. J. Hanson, "Solving Leasr Squares Problems," Prentice-Hall, Inc, Englewood Cliffs, N.J. 1974.

[10] S. Polonsky, http://pws.prserv.net/usinet.svpolon/