J.Sochor

3d magnetostatic and eddy current calculations in the time domain

A quick guide with examples

https://github.com/JNSresearcher/ECMS3D
January, 2026

1 Example for magnetostatics.

Here are some guidelines for constructing a magnetostatic calculation problem.

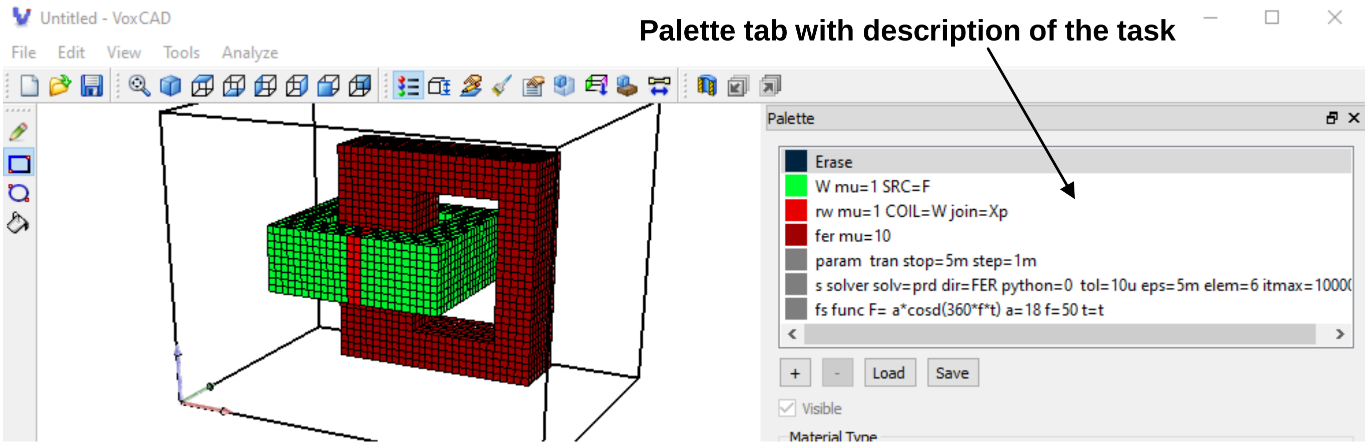


Fig.1. General view of the editor ***VoxCAD***.
On the left is the workspace; on the right is the palette tab.

The text data of the task is entered in the **Palette** tab.

General notes. First, the strings defining the geometry and properties of the domains are placed. Domains are displayed in the workspace. Then come the lines containing the modeling information. (various functions, setting time intervals, etc.). The line begins with a name. Except for lines describing coils, this name is not used. In strings containing modeling information, the line name must be followed by a keyword defining the purpose of the line (e.g. solver, function, etc.) and then sequence of keyword-value pairs. These strings should not be displayed in the work area.

The case of the characters does not matter, as the characters are converted to uppercase during processing.

In our case we have the following text:

```
W mu=1 SRC=F
rw mu=1 COIL=W join=Xp
fe mu=10
t tran stop=5m step=1m
s solver solv=prd dir=FER python=0 tol=10u eps=5m elem=6 itmax=10000
fs func F=a*cosd(360*f*t) a=18 f=50 t=t
```

* **W** string. This is where the coil information is located. The first word is the coil name, **W**. Next, the keyword **mu** is the relative magnetic permeability. Since the coils must always be located in an air environment, then **mu=1**. Then the keyword **SRC** and a reference to a function named **F**. Function **F** determines the amplitude of the current in the coil.
* **rw** string. The main purpose of this string is to indicate the direction of current in the coil. The domain must consist of two cells and be located in the coil's cross-section, in a plane perpendicular to the current direction. In this case, the cross-section is in the **YZ** plane (perpendicular to the **X** axis). The string must contain the keyword **COIL** indicating the name of the coil and the keyword **JOIN** indicating the direction of the current. In this example, a reference is made to a coil named **W**, and the direction of current in this section is along the positive direction of the **X** axis, and is designated as **Xp** (for the opposite direction, **Xm** must be specified). If the sections are located in other planes, then the designations **Yp**, **Ym**, **Zp**, **Zm** are acceptable for the directions (see section *2*).
* **fe** string. **mu=10**. Information about the relative magnetic permeability of the core. In this case $\mu = 10$.
* **t** string. After the keyword **tran -** the parameters of the transient process in seconds are set:

  The keyword **stop** is the time of the transient process. This is a variant of the value with a prefix: ***5m***, i.e. *5e-3 sec*. (More about *prefixes*);

  The keyword **step** - is the time step;

  You can use the keyword **jump -** is the time jump for outputting the results to a file;
* **s** string. After the word **solver -** the solver parameters are set:

  The keyword **solv** is solver type. There are three options. **PRD** for *Intel mkl pardiso* , **GMR** for *GMRES*, **BCG** for *BiCGSTABwr*. If the solver **prd** is specified, the values of the remaining parameters (**tol, eps, elem, itmax**) will be ignored. If the solver parameter **BCG** is specified, only the **tol** and **itmax** parameters are used; the **eps** and **elem** values will be ignored. (It should also be noted that the **BCG** solver does not work for magnetostatics problems).

  The keyword **dir** - specifies the name for the output directory: **FER**.

  The keyword **python**. To decompress the domains map. After the **python** keyword, you can specify the *Python* version: *2* or *3*. If **python=0** (this is the default value), then **gcc** will be used to decompress geometric data. If the data is saved in *ASCII* format, this key is ignored.
* **fs** string. After the word **func**, the description of the function **F** is entered: after the **"="** sign, without spaces, the symbolic expression of the function is written, then the names and values of the arguments. For example, the symbolic expression: **a*cosd(360*f*t)**

has 3 arguments **a**, **f, t**. Numeric values are assigned to the arguments immediately after the function. The **a** argument specifies the amplitude of the current in the coil (*18 A*). The **f** argument set the frequency of the current in the coil. The keyword **t** is assigned to the argument **t** "t". During the calculation, the calculated time will be assigned to this argument.

Note:

* Magnetostatics is calculated using a transient process. In this case, the cosine function initially determines the maximum current. During testing, magnetic field patterns can be obtained at other amplitudes.

* In **ECMS3D** the magnetic core cannot be positioned at a *45*-degree angle.

* Function arguments may contain algebraic expressions enclosed in quotation marks. These expressions may include numbers (without prefixes) and symbolic constants. The following symbols may be used to denote constants:

dx, **dy**, **dz** - grid steps along **x,y,z**.

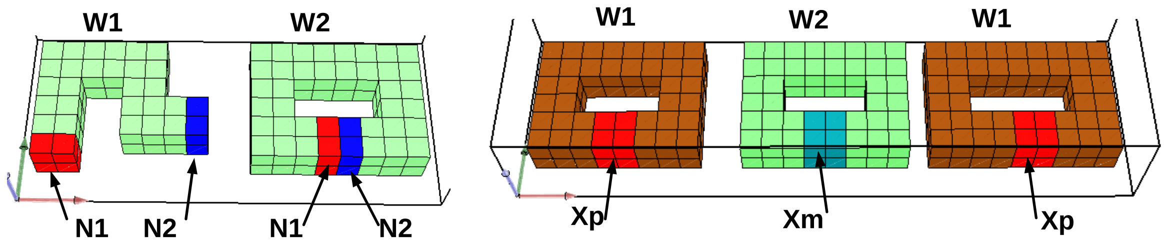
Nx, **Ny**, **Nz** - size of the calculation area along **x,y,z**.

time, **dt, t** - transient time, time step and current time.

pi, e, **mu0**, **e0** : $\pi = 3.1415...$, $e=2.7182...$, $\mu_0=0.12566...10^{-5}$, $\varepsilon_0=0.88541...10^{-11}$, respectively.

2 Creation of direction of external current sources in conductors

In the cross-section of the conductor, it is necessary to place "contacts". There are two methods: in the first, a contact is placed at the beginning (designated by the keywords **join=N1**) and end (designated by the keywords **join=N2**) of the conductor. Each such contact must have a thickness of *1* cell. In the second method, intended only for closed conductors (coils), only one contact, two cells thick, is placed in the coil's cross-section. The direction of the current is specified along the coordinate axes using the keywords **join=** and one of the following: **Xm, Xp, Ym, Yp, Zm, Zp**. If the coils are identical in shape, they can be connected in parallel. After the keyword **par**, the number of parallel coils is indicated.



a) Setting the start (**N1**) and end (**N2**) of current-carrying conductors

b) A double-layer contact in the coil cross-section
Coil **W1** consists of two parts connected in parallel.

Fig.2. Creating a direction of current in conductors

Below are parts of the tasks for Fig. 2. A constant current of *10 A* is set in all conductors.

part of the task for Fig.2 a):

```
W1 mu=1 SRC=F
r11 mu=1 COIL=W1 JOIN=N1
r12 mu=1 COIL=W1 JOIN=N2
W2 mu=1 SRC=F
r21 mu=1 COIL=W2 JOIN=N1
r22 mu=1 COIL=W2 JOIN=N2
......
f func F=a a=10
```

part of the task for Fig.2 b):

```
W1 mu=1 SRC=F par=2
r1 mu=1 COIL=W1 JOIN=Xp
W2 mu=1 SRC=F
r2 mu=1 COIL=W2 JOIN=Xm
......
f func F=a a=10
```

Note. When drawing a coil, aim for a uniform cross-section along its entire length. If the coil's cross-section is too small (usually at "bends"), this may result in warning messages. The explanation is given in Fig.3.

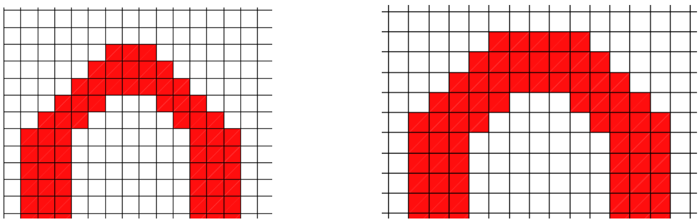


Fig.3. On the left, the coil width at "the bends" is reduced (to 2 cells).
On the right, the coil width at all "bends" is 3 cells.

3 Linear induction machine (LIM).

The magnetic core contains three windings shifted in space by *120* degrees, and a current shifted in time by *120* degrees must flow through them. The conducting plate can move with a given velocity (in Euler coordinates).

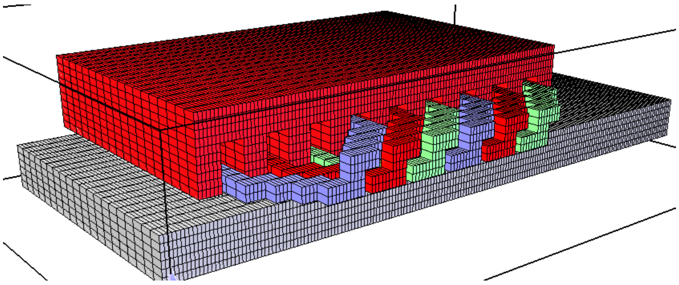


Fig.3. Screenshot of **LIM** in ***VoxCAD***

Data for calculation:

```
A mu=1 SRC=Ia par=2
B mu=1 SRC=Ib par=2
C mu=1 SRC=Ic par=2
rA mu=1 COIL=A JOIN=Yp
rB mu=1 COIL=B JOIN=Yp
rC mu=1 COIL=C JOIN=Yp
Fe mu=100
plate mu=1 COND=37.26e6 Vex=Vx
p1 tran stop=100m step=1m
p2 solver solv=prd dir=Lim1 python=0 tol=10u eps=5m elem=6 itmax=10000
f1 func Ia=a*cosd(360*fs*t) a=10 fs=28.6 t=t
f2 func Ib=a*cosd(360*fs*t-120)  a=10 fs=28.6 t=t
f3 func Ic=a*cosd(360*fs*t+120)  a=10 fs=28.6 t=t
f4 func Vx=tau*fs*impl2(sind(360*fm*t)) tau='36*dx' fs=28.6 fm='1/time' t=t
```

A, **B**, and **C** strings correspond to regions of the three-phase winding and contain a reference to the name of the function that determines the current in the coil (after the keyword **SRC**), as well as an indication of the parallel connection of the two coils of each phase (after the keyword **par**). The string begins with the name of the coil. The coils must be located in air, so **μ** is always equal to *1*.

rA, **rB**, **rC** strings correspond to the cells located in the cross-section of the corresponding coils and contain a reference to the coil name (after the keyword **COIL**) and an indication of the positive direction of the current relative to the **Y**-axis (after the keyword **join**).

Fe string correspond to magnetic core cells with a relative magnetic permeability of *100*.

plate string contains the plate conductivity that follows the **COND** keyword. In this case, it is *37.26e6 S/m*. The **Vex** keyword is followed by the name of the function that calculates the plate velocity in the **x**-axis direction (**Vex=Vx**). Velocities in three directions can be specified. The keywords are **Vex**, **Vey**, and **Vez**.

Note. The size of the conductive domain in any direction must not be less than three cells due to the approximation of the derivative at the edges using a three-point stencil.

The following strings do not apply to domains:

p1 string. After the keyword **tran -** the parameters of the transient process in seconds are set:

The keyword **stop** is the time of the transient process. This is a variant of the value with a prefix: *100m*, i.e. *100e-3 sec*. (More about *prefixes*). The keyword **step -** is the time step (*1m*), that is, there will be an output of *100* points. You can use the keyword **jump** - is the time jump for outputting the results to a file, For example, **jump=2m**, the results will be displayed every *2msec*, that is, we will get not *100* points but *50* points;

p2 string. After the word **solver** - the solver parameters are set:

The keyword **solv** is solver type. There are three options. **PRD** for *Intel mkl pardiso* , **GMR** for *GMRES*, **BCG** for *BiCGSTABwr*. If the solver **prd** is specified, the values of the remaining parameters (**tol, eps, elem, itmax**) will be ignored. If the solver parameter **BCG** is specified, only the **tol** and **itmax** parameters are used; the **eps** and **elem** values will be ignored. (It should also be noted that the **BCG** solver does not work for magnetostatics problems).

The **dir** keyword specifies the name of the output directory: **FER**.

The **python** keyword . To decompress the domains map. After the **python** keyword, you can specify the **Python** version: *2* or *3*. If **python=0** (this is the default value), then **gcc** will be used to decompress geometric data. If the data is saved in *ASCII* format, this key is ignored.

f1, **f2**, and **f3** strings describe functions of time that correspond to the currents in the coils. These are cosine curves, shifted in time by *120* degrees, with amplitude of *10A* and a frequency of *28.6 Hz*.

f4 string contains a function that specifies the plate velocity. This function, **impl2**, converts a sine wave into rectangular pulses with positive and negative amplitudes ***tau*fs***. This means that the plate initially moves at a constant positive velocity, and then, after half the specified transient time, the velocity changes to a constant negative velocity. ***fs*** is current frequency in the coils, ***tau*** is the pole pitch.