



COMSOL Multiphysics 3.4



Gambar A.1 Logo COMSOL Multiphysics

COMSOL Multiphysics is a powerful interactive environment for modeling and solving all kinds of scientific and engineering problems based on partial differential equations (PDEs). With this software you can easily extend conventional models for one type of physics into multiphysics models that solve coupled physics phenomena—and do so simultaneously. Accessing this power does not require an in-depth knowledge of mathematics or numerical analysis. Thanks to the built-in *physics modes* it is possible to build models by defining the relevant physical quantities—such as material properties, loads, constraints, sources, and fluxes—rather than by defining the underlying equations. COMSOL

Multiphysics then internally compiles a set of PDEs representing the entire model. You access the power of COMSOL Multiphysics as a standalone product through a flexible graphical user interface, or by script programming in the COMSOL Script language or in the MATLAB language.

As noted, the underlying mathematical structure in COMSOL Multiphysics is a system of partial differential equations. We provide three ways of describing PDEs through the following mathematical application modes:

- *Coefficient form*, suitable for linear or nearly linear models
- *General form*, suitable for nonlinear models
- *Weak form*, for models with PDEs on boundaries, edges, or points, or for models using terms with mixed space and time derivatives. (The weak form provides many additional benefits, and we review them in the context of specific models in other books in this documentation set.)

Using these application modes, you can perform various types of analysis including:

- Stationary and time-dependent analysis
- Linear and nonlinear analysis
- Eigenfrequency and modal analysis

When solving the PDEs, COMSOL Multiphysics uses the proven *finite element method (FEM)*. The software runs the finite element analysis together with adaptive meshing and error control

(Lanjutan)

using a variety of numerical solvers. A more detailed description of this mathematical and numerical foundation appears in the *COMSOL Multiphysics User's Guide* and in the *COMSOL Multiphysics Modeling Guide*.

PDEs form the basis for the laws of science and provide the foundation for modeling a wide range of scientific and engineering phenomena. Therefore you can use COMSOL Multiphysics in many application areas, just a few examples being:

- Acoustics
- Bioscience
- Chemical reactions
- Diffusion
- Electromagnetics
- Fluid dynamics
- Fuel cells and electrochemistry
- Geophysics
- Heat transfer
- Microelectromechanical systems (MEMS)
- Microwave engineering
- Optics
- Photonics
- Porous media flow
- Quantum mechanics
- Radio-frequency components
- Semiconductor devices
- Structural mechanics
- Transport phenomena
- Wave propagation

Many real-world applications involve simultaneous couplings in a system of PDEs — *multiphysics*. For instance, the electrical resistance of a conductor often varies with temperature, and a model of a conductor carrying current should include resistive-heating effects. This book provides an introduction to multiphysics modeling in the section [“Thermal Effects in Electronic Conductors” on page 33](#). In addition, the *COMSOL Multiphysics Modeling Guide* covers multiphysics modeling techniques in the section [“Creating Multiphysics Models” on page 270](#). The “Multiphysics” chapter in the *COMSOL Multiphysics Model Library* also contains several examples.

Along these lines, one unique feature in COMSOL Multiphysics is something we refer to as *extended multiphysics*: the use of coupling variables to connect PDE models in different geometries. This represents a step toward system-level modeling.

(Lanjutan)

Another unique feature is the ability of COMSOL Multiphysics to mix domains of different space dimensions in the same problem. This flexibility not only simplifies modeling, it also can decrease execution time.

In its base configuration, COMSOL Multiphysics offers modeling and analysis power for many application areas. For several of the key application areas we also provide optional modules. These application-specific modules use terminology and solution methods specific to the particular discipline, which simplifies creating and analyzing models. The COMSOL 3.2 product family includes the following modules:

- Chemical Engineering Module
- Earth Science Module
- Electromagnetics Module
- Heat Transfer Module
- MEMS Module
- Structural Mechanics Module

The CAD Import Module provides the possibility to import CAD data using the following formats: IGES, SAT (Acis), Parasolid, and Step. Additional add-ons provide support for CATIA V4, CATIA V5, Pro/ENGINEER, Autodesk Inventor, and VDA-FS.

You can build models of all types in the COMSOL Multiphysics user interface. For additional flexibility, COMSOL also provides its own scripting language, COMSOL Script, where you can access the model as a Model M-file or a data structure. COMSOL Multiphysics also provides a seamless interface to MATLAB. This gives you the freedom to combine PDE-based modeling, simulation, and analysis with other modeling techniques. For instance, it is possible to create a model in COMSOL Multiphysics and then export it to Simulink as part of a control-system design.

Dielectric Constant Reference Guide

Dielectric Constant (k) is a number relating the ability of a material to carry alternating current to the ability of vacuum to carry alternating current. The capacitance created by the presence of the material is directly related to the Dielectric Constant of the material.

Knowing the Dielectric Constant (k) of a material is needed to properly design and apply instruments such as level controls using radar, RF admittance, or capacitance technologies. There are also analytical reasons to know the (k) of a material.

How to use this guide

CLIPPER CONTROLS has compiled an extensive list of products with Dielectric Constants. Many of these Dielectric Constants are given at specific temperatures. If your product's temperature is significantly different from those listed there is a good chance that the Dielectric Constant may be different from the values listed.

The products in this reference are listed in alphabetical order and are grouped in sections by the first letter of their name. Proper chemical names were used, and any trade names are the trademark of their respective owners. If you know the correct spelling of the name of the product you wish to review then use the "Search" feature on the web browser to locate the name in the list. You may also click on the letter from the alphabetical table to go directly to the beginning of that alphabetic section.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	#
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

-P-

Parrafin Chloride 2.0-2.3

Pine Tree Resin, Powder 1.5-1.8

Pinene (68° F) 2.7

(Lanjutan)

Piperidine (68° F) 5.9

Plaster 2.5 - 6.0

Plastic Grain 65-75

Plastic Pellets 1.1-3.2

Plastic Sulphur, Unground 1.5

Platinum Catalyst 6.5 - 7.5

Poly Propylene 1.5

Polyacetal 3.6-3.7

Polyacetol Resin 2.6-3.7

Polyacrylic Ester 3.5

Polyamide 2.5-2.6

Polybutylene 2.2-2.3

Polycaprolactam 2.0 - 2.5

Polycarbonate 2.9-3.0

Polycarbonate Resin 2.9 - 3.0

Polyester Resin 2.8 - 4.5

Polyester Resin (Flexible) 4.1 - 5.2

Polyester Resin (Glass Fiber Filled) 4.0 - 4.5

Polyester Resin (Ridgid Cast) 2.8 - 4.1

Polyether Chloride 2.9

Polyether Resin 2.8-8.1

Polyether Resin, Unsaturated 2.8-5.2

(Lanjutan)

Polyethylene 2.2-2.4

Polyethylene, Pellet 1.5

Polymide 2.8

Polymonochloro Pifluoroethylene 2.5

Polypropylene 1.5

Polypropylene Powder 1.25

Polypropylene, Pellet 1.5-1.8

Polystyrene Resin 2.4 - 2.6

Polystyrol 2.0-2,6

Polysulphonic Acid 2.8

Polytetra Fluoroethylene 2.0

Polyvinyl Alcohol 1.9-2.0

Polyvinyl Chloride 3.4

Polyvinylchloride Resin 5.8 - 6.8

Porcelain 5.0-7.0

Porcelain With Zircon 7.1 - 10.5

Potassium Aluminum Sulphate 3.8

Potassium Carbonate (60° F) 5.6

Potassium Chlorate 5.1

Potassium Chloride 4.6

Potassium Chloronate 7.3

Potassium Iodide 5.6

(Lanjutan)

Potassium Nitrate 5.0

Potassium Sulfate 5.9

[Return to top](#)



Valeraldehyde (58° F) 11.8

Valeric Acid (68° F) 2.6

Valeronitrile (70° F) 17.7

Vanadium Oxybromide (78° F) 3.6

Vanadium Oxychloride (78° F) 3.4

Vanadium Sulfide 3.1

Vanadium Tetrachloride (78° F) 3.0

Vaseline 2.2-2.9

Veratrol (73° F) 4.5

Vinyl Alcohol Resin 2.6-3.5

Vinyl Butyral 3.3 - 3.9

Vinyl Chloride (Acetate) 3.0 - 3.1

Vinyl Chloride (Flexible) 3.5 - 4.5

Vinyl Chloride (Ridgig) 2.8 - 3.0

Vinyl Chloride Resin, Hard 5.8-6.4

Vinyl Chloride Resin, Soft 2.8-4.0

Vinyl Ether (68° F) 3.9

(Lanjutan)

Vinyl Formal 3.0

Vinylidene Chloride 3.0 - 4.0

Vycor Glass 3.8

[Return to top](#)

-W-

Water 4-88

Water (32° F) 88.0

Water (68° F) 80.4

Water (212° F) 55.3

Water (390° F) 34.5

Water (80° F) 80.0

Water (Steam) 1.00785

Wax 2.4-6.5

Wheat Flour 3.0 - 5.0

Wheat Flour (Dry Powder) 1.6

White Mica 4.5-9.6

Wood, Dry 2-6

Wood, Pressed Board 2.0-2.6

Wood, Wet 10-30

LAMPIRAN C

Berikut tampilan script mencari sensitivitas:

```

clear all
close all
clc
format long
whos
load('C:\MATLAB\U_8KOTAK_SINGLEPLANE\sensor1x.txt','-ASCII');
nx=32;ny=32;nz=32;
b=sensor1x(:,4);
b=squeeze(b);
figure(1)
plot(b)
countV1=0;
Ex1(1:nx,1:ny,1:nz)=0;
for i=1:nx;for j=1:ny;for k=1:nz;
    countV1=countV1+1;
    Ex1(i,j,k)=b(countV1);
end;end;end
figure(2)
for i=1:28
subplot(4,7,i)
mesh(Ex1(:, :, i))
title(['Level=' num2str(i)])
end
load('C:\MATLAB\U_8KOTAK_SINGLEPLANE\sensor1y.txt','-ASCII');
b=sensor1y(:,4);
b=squeeze(b);
figure(3)
plot(b)
countV1=0;
Ey1(1:nx,1:ny,1:nz)=0;
for i=1:nx;for j=1:ny;for k=1:nz;
    countV1=countV1+1;
    Ey1(i,j,k)=b(countV1);
end;end;end
figure(4)
for i=1:28
subplot(4,7,i)
mesh(Ey1(:, :, i))
title(['Level=' num2str(i)])
end
load('C:\MATLAB\U_8KOTAK_SINGLEPLANE\sensor1z.txt','-ASCII');
b=sensor1z(:,4);
b=squeeze(b);
figure(5)
plot(b)
countV1=0;
Ez1(1:nx,1:ny,1:nz)=0;
for i=1:nx;for j=1:ny;for k=1:nz;
    countV1=countV1+1;
    Ez1(i,j,k)=b(countV1);
end;end;end

```

(Lanjutan)

```

figure(6)
for i=1:28
subplot(4,7,i)
mesh(Ez1(:, :, i))
title(['Level=' num2str(i)])
end
load('C:\MATLAB\U_8KOTAK_SINGLEPLANE\sensor8x.txt', '-ASCII');
b=sensor8x(:,4);
b=squeeze(b);
figure(7)
plot(b)
countV2=0;
Ex2(1:nx,1:ny,1:nz)=0;
for i=1:nx;for j=1:ny;for k=1:nz;
    countV2=countV2+1;
    Ex2(i, j,k)=b(countV2);
end;end;end

%b=reshape(b,32,32,32);
figure(8)
for j=1:28
subplot(4,7,j)
mesh(Ex2(:, :, j))
title(['Level=' num2str(j)])
end
load('C:\MATLAB\U_8KOTAK_SINGLEPLANE\sensor8y.txt', '-ASCII');
b=sensor8y(:,4);
b=squeeze(b);
figure(9)
plot(b)
countV2=0;
Ey2(1:nx,1:ny,1:nz)=0;
for i=1:nx;for j=1:ny;for k=1:nz;
    countV2=countV2+1;
    Ey2(i, j,k)=b(countV2);
end;end;end

%b=reshape(b,32,32,32);
figure(10)
for j=1:28
subplot(4,7,j)
mesh(Ey2(:, :, j))
title(['Level=' num2str(j)])
end
load('C:\MATLAB\U_8KOTAK_SINGLEPLANE\sensor8z.txt', '-ASCII');
b=sensor8z(:,4);
b=squeeze(b);
figure(11)
plot(b)
countV2=0;
Ez2(1:nx,1:ny,1:nz)=0;
for i=1:nx;for j=1:ny;for k=1:nz;
    countV2=countV2+1;
    Ez2(i, j,k)=b(countV2);
end;end;end

```

(Lanjutan)

```

end;end;end

%b=reshape(b,32,32,32);
figure(12)
for j=1:28
subplot(4,7,j)
mesh(Ez2(:,:,j))
title(['Level=' num2str(j)])
end
S=-1*((Ex1.*Ex2)+(Ey1.*Ey2)+(Ez1.*Ez2));
whos
figure(13)
for k=1:28
subplot(4,7,k)
mesh(S(:,:,k))
title(['Level=' num2str(k)])
end
save S28 S nx ny nz -mat

```

Setelah sensitivitas didapat maka digabungkan menjadi satu dalam suatu file. Berikut contoh scriptnya:

```

clear all
clc
NameOfDirectory=['C:\MATLAB\U_8KOTAKTWIST45'];
fname=['MatrikSensitifitas'];
ne=8;
m=28;
nx=32;
ny=32;
nz=32;
n=nx*ny*nz;
figure
for p=1:28
filename=['S' num2str(p)];
load(filename, '-mat')
subplot(4,7,p)
mesh(S(:,:,p))
end
clear S
S=S2;
save(['MatrikSensitifitas.smt'], 'S', 'n', 'm', 'nx', 'ny', 'nz', '-mat')
figure
plot(S)

```

Setelah digabungkan menjadi satu maka langkah berikutnya adalah menormalisasi sensitivitas tersebut. Setelah normalisasi didapat maka data yang diperoleh diolah dengan hardware ECVT untuk kemudian mengambil data.