

# The Ultimate Guide to *GridMorph* (Version 1.0)

© 2004 by **MatSim GmbH**, Switzerland

Welcome to *GridMorph*! This user's guide is not like other software manuals because *GridMorph* is much easier to use than any other software for scientific computing.

If you follow the first two examples in this tutorial you will be familiar with *GridMorph* in half a day! You will be able to calculate **physical properties** of a large variety of heterogeneous composite materials and to visualize structures of multi-phase composites.

**Start the tutorial now...**

## Table of content 1

<b>1</b>	<b>Preface</b> .....	<b>4</b>
1. 1.	Composite topologies .....	4
1. 2.	The GridMorph approach to calculate composite properties .....	5
1. 3.	Physical properties that can be calculated. ....	5
<b>2.</b>	<b>My first project</b> .....	<b>7</b>
2. 1.	Getting started and understanding the user interface .....	7
2. 1. 1.	Rotating the model . . . . .	7
2. 1. 2.	Zooming the model . . . . .	8
2. 1. 3.	Flying through the model . . . . .	8
2. 2.	Loading a project and understanding the main window. ....	9
2. 3.	Periodic Boundary Conditions .....	10
2. 3. 1.	Example: Hexagonal fiber packing . . . . .	11
2. 4.	Tetrahedron mesh of an orthogonal grid .....	12
2. 5.	Viewing the model .....	13
2. 5. 1.	Perspective vs. orthographic mode. . . . .	13
2. 5. 2.	Viewing coordinate axes . . . . .	13
2. 5. 3.	Graphics modes . . . . .	13
2. 6.	View Background Color .....	14
2. 7.	Viewing Options .....	14
2. 8.	Calculating properties of the composite .....	14
2. 8. 1.	Store Property Fields . . . . .	15
2. 8. 2.	Property field data files . . . . .	16
	Elastic Constants (cc) . . . . .	16
	Thermal Expansion Coefficients (te) . . . . .	16
	Thermal Conductivity (tc) . . . . .	16
	Electric Conductance (ec) . . . . .	16
	Dielectric Constants (dc) . . . . .	17
	Gas Permeability (gp) . . . . .	17
	Gas Diffusivity (gd) . . . . .	17

2. 8. 3.	Understanding the deformation data . . . . .	17
2. 8. 4.	Interrupting long operations (STOP BUTTON) . . . . .	18
2. 9.	Saving the project and results . . . . .	18
2. 10.	View Results . . . . .	19
2. 11.	Viewing a cut plane through the volume mesh . . . . .	19
2. 11. 1.	Cut Plane of Periodic Mesh: Geometric Considerations . . . . .	21
2. 11. 2.	Visualization of Property Fields . . . . .	21
2. 11. 3.	Checkerboard effect in local property display . . . . .	21
2. 12.	Quitting the program . . . . .	22
<b>3.</b>	<b>My second project</b> .....	<b>23</b>
3. 1.	Creating a new project . . . . .	23
3. 2.	Import an arbitrary grid morphology . . . . .	23
3. 2. 1.	Averaging method . . . . .	23
3. 2. 2.	Load phases and assign materials . . . . .	24
3. 3.	Load an arbitrary grid morphology . . . . .	25
3. 4.	Editing material properties of the components . . . . .	25
3. 4. 1.	Editing elastic constants . . . . .	26
3. 4. 2.	Editing thermal expansion coefficients . . . . .	27
3. 4. 3.	Editing thermal conductivity . . . . .	27
3. 4. 4.	Editing dielectric constants . . . . .	27
3. 4. 5.	Editing electric conductivity . . . . .	27
3. 4. 6.	Editing gas permeability . . . . .	28
3. 4. 7.	Editing gas diffusivity . . . . .	28
3. 4. 8.	Editing material colors . . . . .	28
3. 5.	Tips and tricks to calculate related properties . . . . .	28
3. 5. 1.	Liquid permeability and diffusivity . . . . .	28
3. 5. 2.	Refraction index . . . . .	29
3. 6.	Reorganizing the Materials Database . . . . .	29
3. 7.	Assigning material properties to phases . . . . .	29
3. 8.	Calculating composite properties . . . . .	30
<b>4.</b>	<b>Information, warning, and error messages</b> .....	<b>31</b>
4. 1.	Error and information messages . . . . .	31
4. 2.	Warning messages . . . . .	37
<b>5.</b>	<b>File Formats</b> .....	<b>39</b>
5. 1.	File Dependencies . . . . .	39
5. 2.	General conventions . . . . .	40
5. 3.	Script File . . . . .	40
5. 4.	Log (Project) File . . . . .	40
5. 5.	Materials Database File . . . . .	41
5. 5. 1.	Content . . . . .	41
5. 5. 2.	Format . . . . .	41
5. 6.	Configuration File . . . . .	41
5. 6. 1.	Content . . . . .	41
5. 6. 2.	Format . . . . .	42

5. 7.	Phase File for Grid Morphologies	42
5. 7. 1.	Content	42
5. 7. 2.	Format	43
5. 8.	Materials Definition File	43
5. 8. 1.	Content	43
5. 8. 2.	Format	45
5. 9.	Output File	47
5. 9. 1.	Content	47
5. 9. 2.	Format	47
5. 10.	Result Files for Property Fields	48
5. 10. 1.	Content	49
	Elastic Constants ( <i>cc</i> )	49
	Thermal Expansion Coefficients ( <i>te</i> )	49
	Thermal Conductivity ( <i>tc</i> )	50
	Electric Conductance ( <i>ec</i> )	50
	Dielectric Constants ( <i>dc</i> )	50
	Gas Permeability ( <i>gp</i> )	50
	Gas Diffusivity ( <i>gd</i> )	50
5. 10. 2.	Format	50
5. 11.	Backup Files	51
<b>6.</b>	<b>Batch Processing</b>	<b>52</b>
6. 1.	Batch Mode	52
6. 1. 1.	Interactive mode	52
6. 1. 2.	Batch mode	52
<b>7.</b>	<b>Script commands</b>	<b>54</b>
7. 1.	Varia	54
7. 1. 1.	Comments	54
7. 1. 2.	Line-continuation character	54
7. 1. 3.	System calls	54
7. 1. 4.	Quit	55
7. 1. 5.	Parameter assignments	55
7. 2.	Project commands	55
7. 2. 1.	Project – Load	55
7. 2. 2.	Project – Save	56
7. 2. 3.	Project – Save As	56
7. 3.	Materials commands	56
7. 3. 1.	Material – Assign Properties	56
7. 3. 2.	Material – Save Composite	56
7. 4.	Solver commands	57
7. 4. 1.	Solver	57
<b>8.</b>	<b>Literature</b>	<b>58</b>

# 1. Preface

New demanding applications require high-performance materials. Composite materials that combine the best properties of their homogeneous phases are optimal candidates for such applications. Heterogeneous composite materials consisting of several phases provide an unlimited variety of morphologies and properties. Because the microstructure highly affects the composite properties it is crucial to find the morphologies with the optimal overall properties.

## 1. 1. Composite topologies

There are many composite materials with topologically complicated structures:

**Particulate topologies** can be found in all kind of composites where particles such as whiskers, platelets or otherwise shaped object are surrounded with a continuous matrix material. Examples are fiber or otherwise reinforced polymers or Swiss cheese (where the particles are air hollows) (see also list of [literature references](#)).

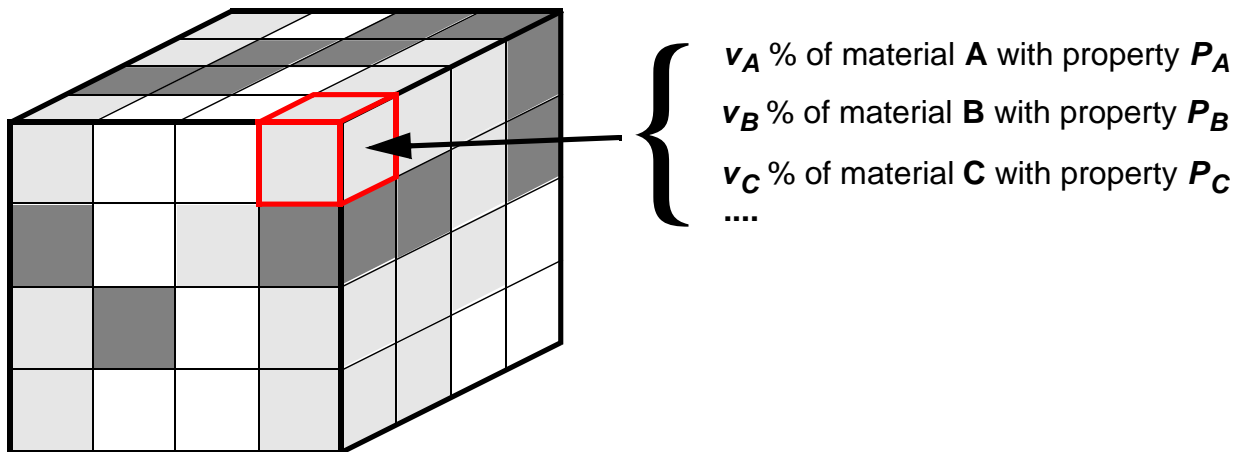
Poly crystalline single phase metals are mechanically heterogeneous due to the non-isotropic elastic and plastic behavior of crystals with various orientations. Such **granular topologies** correspond to a space-filling solution where different phases are contiguous but none can be seen as containing the other.

Polymer blends, block-copolymer melts, austeno-ferritic steels, metallic foams or porous silicon often are of **bi-percolated topology** because the phases are continuous so that they can be seen as containing each other.

## 1. 2. The GridMorph approach to calculate composite properties

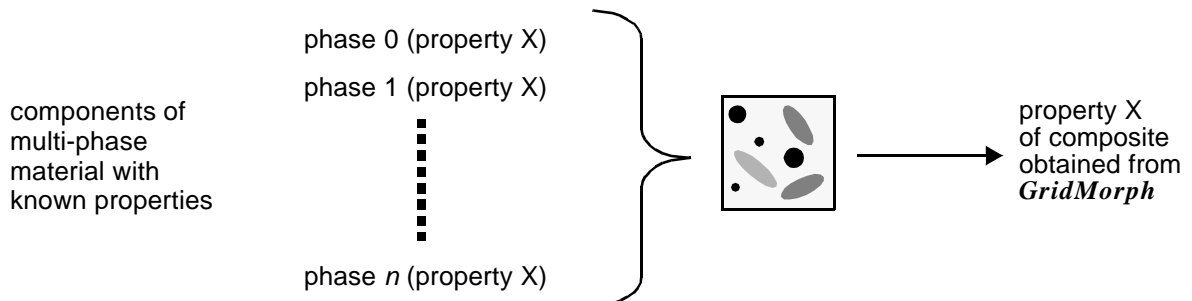
*GridMorph* uses a numerical method to determine the overall properties of composites with arbitrary morphologies from the properties of the components based on small homogeneous grid elements. The morphology is defined by a number of phases in a periodically continued base cell of cubic or orthorhombic shape where the phases may consist of any material. The resolution depends solely on the

number of grid elements used. For each of the grid elements it is possible to specify the fraction of each phase contained in that particular position.



While the grid elements are cubic, the number of elements in each direction can be different in order to adjust the “resolution”. Using a small number of grid elements (minimum 3) in one or two directions allows also to simulate pseudo-planar (two-dimensional) and layered (one-dimensional) structures.

By applying a displacement-based finite element method to the volume mesh, the responses to external deformations are calculated (Gusev 1997). Graphical visualization of the local properties allow to spot out the “weak points” of the structures. The computed overall properties of the composite can be used as input to calculate a new super-composite.



### 1. 3. Physical properties that can be calculated

The algorithms implemented in *GridMorph* allow to easily calculate the following physical properties:

- elastic constants (stiffness and compliance)
- thermal expansion coefficients
- thermal conductivity
- electrical conductivity
- dielectric constants
- gas diffusion

- gas permeability

More properties can be calculated by applying **some tricks**:

- refraction index
- magnetic moments
- swelling in solvents
- liquid diffusion
- liquid permeability

Some practical applications of the calculations of above properties can be found in the **Literature**.

## 2. My first project

### 2. 1. Getting started and understanding the user interface

To start *GridMorph* on your computer simply type `GridMorph` or double-click the *GridMorph* icon. If neither of these is successful you may have to **install the program** first.

At start-up the user interface of *GridMorph* consists of two windows: the viewer, which shows an empty simulation cell, and the main window with a menu bar, an information section and a scrollable text box that shows log information of the program.

The user interface is written for the **X/Motif™ platform** and behaves according to the implicit navigation model as described in the OSF/Motif™ Style Guide. This means that the mouse must always be somewhere inside the window which you want to be active. For input fields (where you enter text or numbers) you must first click with the mouse inside this field. To replace a default value in the input field, double-click the number and type the new value. To get to the next input field in the same dialog you can use the TAB key. By pressing Shift + TAB the input focus goes to the previous field. If you are in the last input field and press TAB again, the focus will be on the OK button, which can be activated by pressing the Space bar.

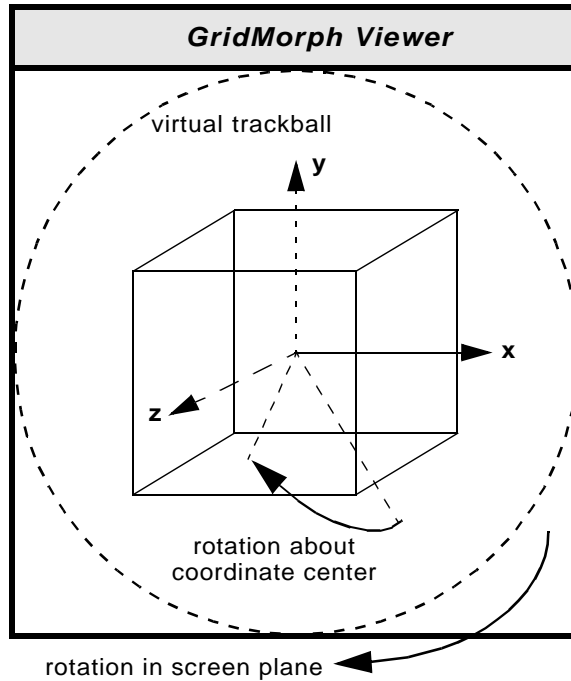
Input fields are generally checked if you click with the mouse pointer in another field or if you press the TAB key. This allows *GridMorph* to tell you immediately if an input value does not make sense and gives you the opportunity to correct it quickly.

#### 2. 1. 1. Rotating the model

In the viewer window you notice the wire-frame of a cube, which represents the basic simulation cell. In the center of the cube are coordinate axes x (solid line), y (dotted line), and z (dashed line). To look at the cube from different sides you can rotate it by clicking with (or just moving) the mouse pointer in the viewer and then using the following keys to rotate about the coordinate axes: The up and down arrows on the cursor keypad rotate the cube clockwise (up) and anti-clockwise (down) about the x axis, by looking at the axis from the positive direction. Arrows to the left and right rotate the cube about the y axis in clockwise (right) and anti-clockwise (left) direction. A rotation about the z axis is possible by pressing the "<" (anti-clockwise) or ">" (clockwise) keys on the keyboard.

An alternative way of rotating the model is through using the mouse: Assume that a trackball is installed at the center of the coordinate system and touches the borders of the viewer window. You can click somewhere on the surface of the trackball (the position is given by a parallel projection of the mouse pointer's po-

sition onto the virtual trackball) and rotate the ball about the coordinate center. To rotate the model in the plane of the screen you can click in one of the corners of the viewer (outside the virtual trackball) and move the mouse outside the trackball.



The trackball itself is not visible, however you can easily find out where it is by moving the mouse from a corner of the viewer towards the coordinate center. As soon as the mouse "touches" the trackball the model will rotate out of the screen plane.

### 2. 1. 2. Zooming the model

To zoom the simulation cell the "Page Up" (zoom in) and "Page Down" (zoom out) keys are used and to return the cube to its initial orientation and size you can press "Home". Zooming the model with the mouse is possible by pressing the right mouse button and moving the mouse up (zoom in) or down (zoom out).

### 2. 1. 3. Flying through the model

Additionally to rotating and zooming of the simulation cell there is a flight option available. If ticked in the **View – Options** dialog, pressing the middle mouse button changes the mouse cursor to a little airplane and it allows you to translate the simulation cell in the plane of the screen in opposite direction of the mouse cursor. The airplane means that your cursor behaves like a control stick of your plane and you fly in the direction where you move the cursor. In combination with the right mouse button (for zooming) you can fly forth and back in any point of the viewer.

If you get lost and nothing can be seen any more, press the Home key on the keyboard in order to bring the simulation cell back to the center of the viewer. If flight mode is on the trackball rotation does not work properly any more because

the rotation and translation are not compatible. In order to switch back to rotation mode, untick the “flight mode” button in **View – Options**.

Toggling flight mode can also be done through pressing the F key when the mouse cursor is inside the viewer. Pressing the middle mouse button shows a pointer that makes clear whether you are in flight mode or not.

## 2. 2. Loading a project and understanding the main window

A quick hint before you get started: *GridMorph* issues two types of messages, which appear either in an orange or a green window.

- Orange messages are errors that require a repetition of the last step. If the message itself is not clear enough for you to understand what to do, simply **check the list of errors** (in alphabetical order) with additional explanations
- Green messages are warnings that require a decision “yes” or “no”. If you are not sure about the answer it is advisable to **check the list of warnings** below.

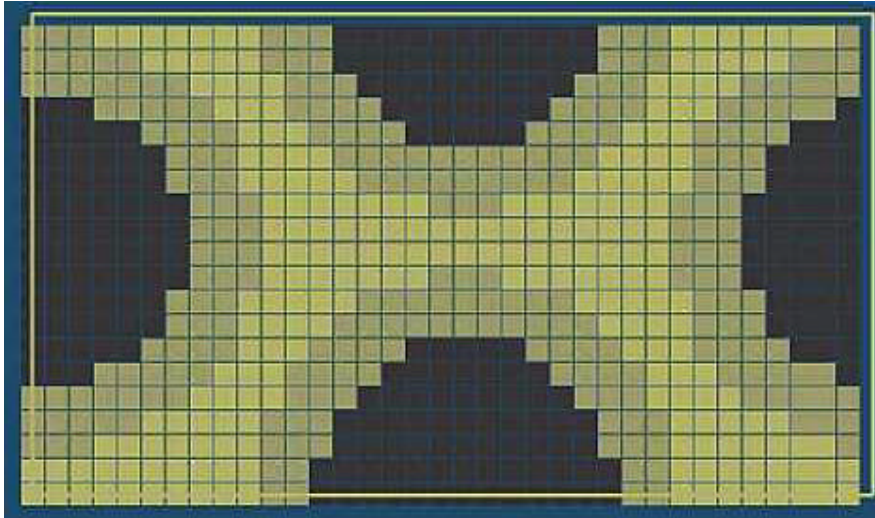
To work yourself through a simulation project you can intuitively follow the sequence of topics in the menu bar. First, place the mouse on the menu bar over the word **Project** and hold the left mouse button. A pull-down menu with the options **New, Open, Save, Save As, Import** and **Quit** appears. To start with you might want to have a look at an already existing project: in the **Project** pull-down move the mouse down (always holding the left mouse button) until the word **Open** is highlighted. Now release the left mouse button and wait for the file selection box to pop up.

The file selection box consists of a “Filter” field which is set by default in a way that all project files (those with filename extension `.log`) in the current directory are displayed in the “Files” box. To select a file, click on it with the mouse (it will then appear in the “Filename” box) and press the **OK** button on the bottom.

If you want to change the directory you click on the desired line in the “Directories” box (it will then appear in the “Filter” box) and click the **Filter** button. To discard the file selection box you simply click **Cancel**.

To load the example project you click now in the “Files” box inside the **Open Project** dialog on the file name `3Phase.log` and then on the **OK** button. The viewer immediately displays a non-cubic grid, made up of 5 x 35 x 20 cubic grid elements, that looks “from top” (from the direction of the x axis, with the y axis pointing to the right) as shown in the following figure.

As soon as the project is loaded the main window changes. In the top left field the states of the project are displayed: the project name is `3Phase` and the status of the configuration is “loaded” (that is why you can see it in the viewer). Analysis data is not yet available for this project.



In the top right field of the main window the configuration is briefly described with the dimension of the grid and the number of grid nodes and mesh elements. The latter are six times the number of grid nodes.

While loading the project number of phases and the volume fraction of these in the simulation cell. Below that the current working directory is indicated.

The bottom part of the main window shows the log box with runtime information of *GridMorph* as it is also stored in the **project file** (with the extension `log`). Among other information you can find there the number of phases, their origins (data file) and the volume fraction of these in the simulation cell. You can display different parts of the log info by moving up and down the scroll bar on the right side of the log box with the mouse, or by pressing “Page Up” and “Page Down” with the mouse pointer positioned in this box.

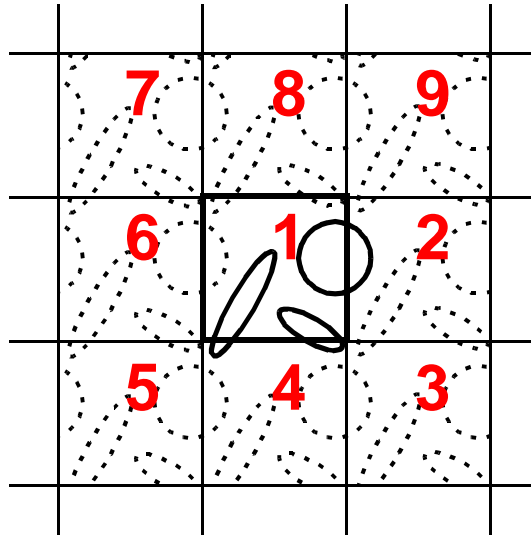
The next two chapters explain why this project actually simulates a system of densely packed parallel fibers and why the number of mesh elements is six times the number of grid points. If you want to have just a quick look at this example you can skip this section and continue with **2. 5. Viewing the model**.

## 2. 3. Periodic Boundary Conditions

Simulating properties of materials is very different from simulating properties of a complex-shaped structural part consisting of a homogeneous material. While the latter has a clearly defined shape here we merely want to avoid any boundaries by simulating pure material properties. For this reason we consider the simulation cell as periodically continued, which means that the right face of the cell corresponds exactly to the left, the top face corresponds to the bottom, and the front face corresponds to the back. With this construction we can attach copies of the cell to all its faces and we will not notice any interfaces between the cells. This construction can be continued infinitely in order to get a large bulk of material with-

out any boundaries. Rather than simulating an “infinite” system, however, it is sufficient to consider just a small part, represented by the basic cell.

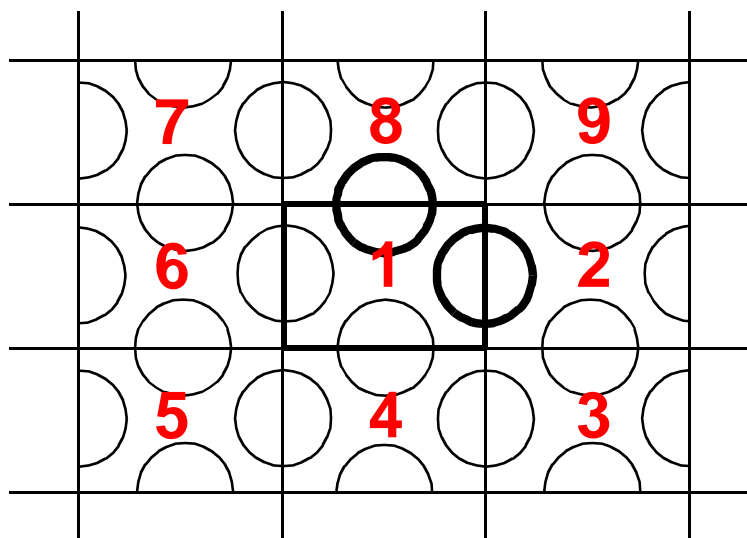
The construction of a so called periodic cell with periodic boundary conditions (PBC) requires that a phase that overlaps the cell borders on one side has to be continued on the other side of the cell by reentering the cell from the opposite direction.



The above picture shows a two-dimensional projection of a basic simulation cell (1) in the center with three inclusions representing different phases and the nearest images of this cell (2-9). All inclusions are overlapping the central cell thus their images enter the cell from the opposite side.

### 2. 3. 1. Example: Hexagonal fiber packing

Coming back to our example of parallel fibers of infinite length (in x direction) in a hexagonal packing, the above picture looks as follows:



The central cell is now not rectangular but one side (in y direction) is  $3^{1/2}$  longer than the other (in z direction) and the two bold fibers are periodically continued in all neighboring cells, thus giving a perfect hexagonal packing of circles. In our ex-

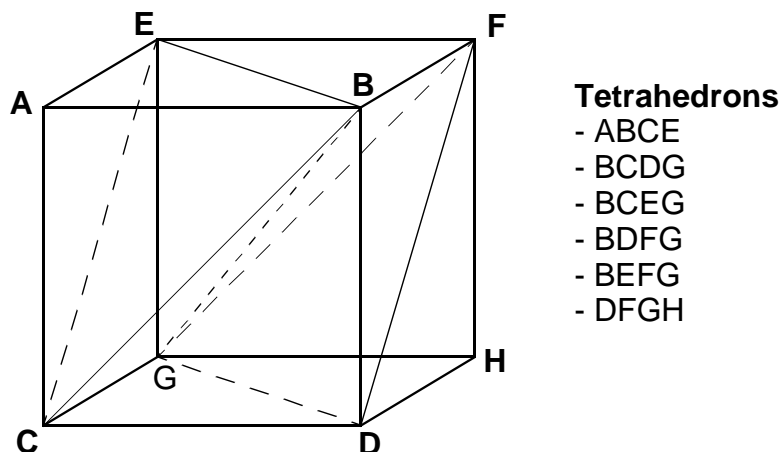
ample the system has a “thickness” of 5 identical layers that are periodically continued in the x direction. As the layers are identical a minimum of three layers would be sufficient.

*GridMorph* calculates with discrete systems based on a regular grid structure. Therefore it is not possible to perfectly represent the ratio of  $3^{1/2}$  for the cell dimensions. In this simple example the dimensions are 20 x 35. Also the fiber cross sections are not perfectly round but rather jagged due to the representation by square boxes. Both “inaccuracies” can be reduced by increasing the total number of grid elements. *GridMorph* can handle systems with a few million grid points depending on the computer’s memory, thus allowing a very high precision for the property calculations.

You will have noticed that in the example on your screen the fibers have a thin surface layer. This could be a surface coating to improve adhesion between the fibers and the matrix and is actually the third phase (number 2) of the sample system.

## 2. 4. Tetrahedron mesh of an orthogonal grid

Although *GridMorph* models are represented as a regular cubic lattice, the solver works on the basis of a space-filling tetrahedral mesh, i.e. without voids. The mapping of cubic grid elements to a tetrahedral mesh results in six *isochoric* (of the same volume) tetrahedrons for each cubic grid element:



For this reason the program shows that the number of mesh elements is six times the number of grid nodes, which is equal to the number of grid elements. There is only one dissection of a cube that results in fewer tetrahedrons, resulting in four identical elements of the shape ABCE and a regular tetrahedron BCEH of double volume. Although this would result in a 16 % reduction of mesh elements, *GridMorph* uses the above dissection because it is the most simple one that is periodic, i. e. the sides ACGE and BDHF, ABDC and EFHG, and ABFE and CDHG can be joined together under **periodic boundary conditions**.

The tetrahedron mesh can be **visualized** following the instructions in the next section. By scanning through the mesh with the **Cut Plane** it is possible to follow the edges of these tetrahedrons.

## 2. 5. Viewing the model

### 2. 5. 1. Perspective vs. orthographic mode

You already know how to rotate and zoom the model in the viewer. Now, after loading a project including geometric configuration and mesh data, you can try the different visualization options. By default the model is displayed in perspective view; if you activate the **View** pull-down you notice before the word “Perspective” an activated toggle. To switch to an orthographic view you have to select **Perspective**. The model appears now in a parallel projection. Try to rotate the cube in a way that you see it exactly from front.

### 2. 5. 2. Viewing coordinate axes

To get rid of the coordinate axes in the viewer, deactivate the toggle button before **Coordinate Axes** in the **View** pull-down. To redisplay the axes, make sure the toggle button in **Coordinate Axes** is ticked.

### 2. 5. 3. Graphics modes

The configuration can be displayed in nine different graphics modes and for the cell there are three different options available, which results in a total of 27 different views. The following list explains what you should be able to see depending on the **Graphics mode**:

- **Nodes**: Show all mesh nodes; these are the grid points of the regular cubic grid. The size of the nodes can be adjusted in **View - Viewing Options** to give a better picture. This mode can also be activated by pressing F2 while the focus is on the viewer window.
- **Mesh**: Show the full volume mesh by drawing all tetrahedrons (if the mesh is available). A better mesh visualization is provided by the **View - Cut Plane** facility. This mode can also be activated by pressing F3 while the focus is on the viewer window.

The colors of the nodes or mesh elements depend on the material type of the corresponding phases. If no materials are defined the phases appear artificially painted. To modify the colors of different materials you can use the **Materials - Edit Material Properties** Dialog for the respective material.

Another graphics mode allows to display plane cuts through the mesh. This Cut Plane mode is describe in detail in [2. 11. Viewing a cut plane through the volume mesh](#).

## 2. 6. View Background Color

The **View - Background Color** dialog allows to change the background color of the graphics viewer. The color is composed of the three RGB components red, green and blue given in percent. Use the sliders for each color component or directly enter the numbers to modify the background color. The **Apply** button makes the changes visible in the viewer. **Reset** brings back the previously stored color scheme. After applying the **Close** button the current background color will become the default color.

The initial *GridMorph* color is 0 % red, 18.43 % green, 30.59 % blue (Pantone® 309C, the MatSim color).

## 2. 7. Viewing Options

To emphasize certain elements of the model in the graphics viewer it is possible to set the sizes for nodes, the line width for volume mesh elements, the line width of the coordinate axes (can be switched on and off in **View - Axes**), and the line width of the simulation cell.

The shape of the points can vary between different OpenGL versions (points can be squares or circles). In some OpenGL implementations overlapping points cause interference patterns that make visibility of points difficult. The optimal point size can be found through playing around with this parameter.

## 2. 8. Calculating properties of the composite

If your configuration has material properties for all phases **defined** and **assigned** (this is already the case in the `3Phase` example), you can start the **Solver - Composite Properties** to calculate the composite properties. The dialog that pops up hides actually two different solvers: in order to calculate mechanical properties and thermal expansion coefficients a thermo-elastic solver is used. It applies six different infinitesimally small deformations to the composite and minimizes the total strain energy for each of these deformations in order to calculate the elastic composite properties. To calculate thermal expansion of the composite a seventh “deformation” (an increase of temperature by 1 Kelvin) is applied in order to obtain the linear thermal expansion coefficients ([Gusev 2001-1](#)).

For other **physical properties** such as conductivities, dielectric constants, and transport properties a **Laplace solver** is used, that applies a field in the three main directions to the finite element mesh and minimizes the energy of the composite.

To calculate some properties of the 3Phase example select the buttons *Elastic constants*, *Thermal expansion coefficients* and *Thermal conductivity*. As the calculation will be reasonably short you can leave all other parameters as they are.

The time required to minimize the energy for both the thermo-elastic and the Laplace solver (by means of a conjugate-gradient method) depends on a number of factors:

- the **number of grid elements**;
- the **number of different phases** because the phase properties have to be averaged (see **3. 2. 1. Averaging method**) for every grid element;
- the **convergence parameter** specifies by which factor the initial gradient should be reduced: if you just want to get a rough estimate of the composite properties and the properties of the components are not very different, it may be enough to specify a convergence of 0.001. The maximum number of minimization steps makes sure that the solver will not run forever even if the required convergence can not be achieved;
- the **maximum number of minimization steps** limits the minimization time to a constant value depending on the number of mesh elements and independent of the convergence parameter;

The types of deformations applied to the composite in order to calculate thermo-elastic behavior are:

- normal strain in the x-direction
- normal strain in the y-direction
- normal strain in the z-direction
- shear in the yz plane
- shear in the xz plane
- shear in the xy plane
- thermal strain corresponding to a small change in temperature.

For each of the Laplace properties a corresponding field is applied in

- field direction 1 (corresponding to the x-direction)
- field direction 2 (corresponding to the y-direction)
- field direction 3 (corresponding to the z-direction)

For each type of deformation, *GridMorph* reports the total energy and the gradient reduction. The Solver automatically stores these results in the **output file** (with extension `out`) so that they are available immediately in another session and therefore do not have to be recomputed for **visualization purposes**.

### 2. 8. 1. Store Property Fields

Beside calculating the overall properties of a composite material you can also store the property fields for postprocessing such as later visualization in **View – Cut Plane**.

If the toggle button **Store property fields** is pressed in the solver dialog (de-

fault setting), *GridMorph* stores the following data for each tetrahedron mesh element in a separate file. The extensions of these property field files start with the letter “F” for “field” or “flux”. A two-letter abbreviation specifies the field type (usually the property code). The last letter of a 4-letter extension stands for the field or deformation direction; the last two letters of a 5-letter extension stand for a shear deformation as listed in the Table in [5. 10. Result Files for Property Fields](#).

When calculating big systems you may get the “Disk full” error because the property field files use a lot of memory. In this case it may be helpful to run the solver for each property separately and store the property fields only for those properties where you are really interested in the detailed fields.

## 2. 8. 2. Property field data files

The details about the data stored in the property field files can be found in the [file description](#). The property codes in the following sections are shown in parentheses, these codes are used for the extension of the [property field files](#).

### Elastic Constants (cc)

Deformations in x, y, z direction; shear in xy (gamma), yz (alpha), xz (beta) plane. For each of the 6 deformation types the local hydrostatic pressures (pr) and von Mises (vm) stresses are stored. See also [2. 8. 3. Understanding the deformation data](#).

The elastic constants are usually specified in GPa.

### Thermal Expansion Coefficients (te)

Strain through heating the cell by 1 K. The local hydrostatic pressures (pr) and von Mises stresses (vm) are stored for this temperature change.

Thermal expansion coefficients are specified in  $K^{-1}$ .

### Thermal Conductivity (tc)

Heat field in x, y, z direction. For the 3 “deformation” types the local heat fluxes are stored.

The units of thermal conductivity are  $Wm^{-1}K^{-1}$ . Since the applied field is in K/m (kelvin per meter) the heat field is given in  $W/m^2$  (watt per square meter).

### Electric Conductance (ec)

Electric field in x, y, z direction. For the 3 “deformation” types the local electric fluxes (currents) are stored.

The units of the electric conductance are S/m (Siemens per meter), which is equivalent to  $A/(Vm)$  (ampere per volt-meter). Since the applied field is in V/m the electric field is given in  $A/m^2$ .

### Dielectric Constants (dc)

Dielectric field in x, y, z direction. For the 3 “deformation” types the local electric fields are stored.

Dielectric constants are dimensionless; the applied field is in V/m, therefore the field induced is also in V/m.

### Gas Permeability (gp)

Permeability field in x, y, z direction. For the 3 “deformation” types the local gas fluxes are stored.

The units of the permeability of gases are barrer (equals to  $10^{-16} \text{ m}^3$ ). The applied field is in c/m (concentration per meter). The units of the permeation field are therefore  $10^{-16} \text{ c/m}^2$ .

### Gas Diffusivity (gd)

Diffusivity field in x, y, z direction. For the 3 “deformation” types the local gas fluxes assuming the same solubility for all phases are stored.

Diffusion constants are given in  $\text{cm}^2/\text{s}$ ; the applied field is in c/m (concentration per meter). The units of the flux field are therefore c/s.

## 2. 8. 3. Understanding the deformation data

For thermo-elastic properties (mechanical properties and thermal expansion coefficients) the total elastic energy  $F$ , which is a quadratic function of the displacements of the nodes, is defined as the sum over the element contributions:

$$F = \sum_{t=1}^{n_{Tetr}} f_t = \frac{1}{2} \sum_{t=1}^{n_{Tetr}} V_t C_{ik} e_i e_k$$

where  $f_t$  is the elastic energy of tetrahedron  $t$ ,  $V$  its volume,  $C_{ik}$  the elastic constants, and  $e_j$  the local strains. In general, the  $e_j$  are made up of mechanical and thermal parts. The units of the energy  $F$  are  $[\text{L}^3 \cdot \text{GPa}]$ , where  $L$  is the length unit of the simulation cell.

The local stresses are calculated from the nodal displacements at the end of each minimization run assuming linear elastic behavior. They correspond to an applied mechanical strain of 1 % or a temperature change of 1 K.

The local hydrostatic pressures  $P$  are calculated for each tetrahedron as the sum of the principal components of the stress tensor of that volume element

$$P = (\sigma_{xx} + \sigma_{yy} + \sigma_{zz})/3$$

These values can be calculated with a spread sheet program by loading the appropriate property field files and performing the right calculations between their values.

The deviatoric von Mises stress for a mesh element is defined as

$$\tau = \sqrt{\frac{((\sigma_{xx} - \sigma_{yy})^2 + (\sigma_{yy} - \sigma_{zz})^2 + (\sigma_{zz} - \sigma_{xx})^2)/2 + 6(\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2)}{2}}$$

This value is listed in the output file for every applied deformation; the units are the same – [GPa] – as for the elastic constants.  $\tau$  typically controls the appearance of first inelastic events in metals and organic polymers. For example, in many organic polymers the first inelastic events such as plasticity start occurring when the von Mises stress reaches a level of 50-100 MPa.

The von Mises stresses calculated by *GridMorph* are based on a deformation of 100 %. If you assume for your composite a deformation of 3 % you have to multiply  $\tau$  with 0.03 in order to get the effective von Mises stress. For most organic composites and brittle materials one can assume that failure occurs if the local von Mises stress exceeds 100 MPa.

#### 2. 8. 4. Interrupting long operations (STOP BUTTON)

Running the solver may take some time, depending on the **parameters described before**. For such long-running tasks *GridMorph* offers a red **STOP BUTTON** in the dialog window that can be clicked with the mouse in order to interrupt the current operation. After an interrupt or normal termination of the operation the stop button disappears automatically.

Many other operations such as Monte Carlo packing, meshing, file loading, etc. can be interrupted in the same way (if the STOP BUTTON is available). The operation may not terminate immediately because some internal bookkeeping has to be done in order to keep the program in a stable state. After interrupting an operation it is usually necessary to restart it – a continuation at the halting point is often not possible.

#### 2. 9. Saving the project and results

After performing all minimization steps the Solver automatically stores the output file which has the name of the current project and the extension `out`. This file contains all necessary information about the composite material. However, if you want to use these properties for a new material in another project, you have to save it separately in a material file: In the **Materials** pull-down you find as the last option **Save Composite Material**; if you select it the properties will be stored in the materials database file with the name of the project and extension `mat`.

To save all other data of the current project you have to choose **Save** from the **Project** pull-down or **Save As** if you want to store the data under a new project name. With the **Save** option *GridMorph* automatically stores all project data that was modified since you saved it last time; the program also creates a backup copy

of each file if a file with the same name is already available in the current working directory. The backup files are marked with the extension `_bk`.

The **suite of files** for a project "`pname`" consists of the following:

- `pname.log`: this is the main **project file** without which the project is not defined. It contains the log book of a project run and can be used to trace back the operations that were performed during the project's session.
- `pname.mdb`: the **Materials Database file** contains the names of all material files that are required in the current project. In the material files the properties of a particular material are stored. These files have an arbitrary file name (usually a descriptive name for the material they describe) followed by the extension `mat`.

The materials database file is automatically **reorganized** upon a save operation. After this the entries (material properties) will be sorted according to their appearance in the configuration (properties of phases in ascending order, starting with phase 0).

- `pname.cfg`: the **ConFiGuration file** holds the information about the different phase files and how the phases are weighted and averaged. The configuration file also contains information about the material of each phase as an index number in the **materials database** (`mdb`) file.
- `pname.out`: the **OUTput file** is created after a successful run of the Solver. It summarizes the main information about the configuration and the volume fractions of different materials in the composite. All properties that could be determined during the Solver run are listed in an easily readable form.
- `pname.mat` is another **material file** that contains the extracts from the out file that are sufficient to fully describe the properties of the composite material. This file can be used in another project as the material of a "homogeneous" phase.

In addition to the files above there can be a set of files containing the **property field data**.

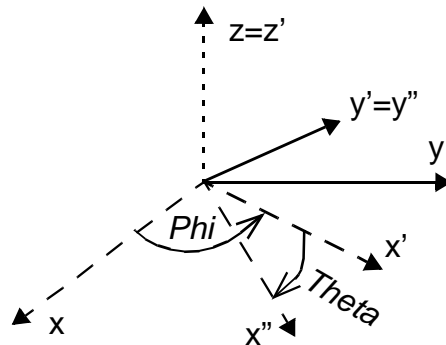
## 2. 10. View Results

With **Solver - View Results** you can display the latest **output file** (extension `out`) of the current project in a separate window.

## 2. 11. Viewing a cut plane through the volume mesh

Beside the mesh node and element views selected by **View - Graphics Mode** it is possible to visualize the volume mesh by plane cuts through the mesh. In comparison to the mesh graphics mode described before this facility allows to analyze the volume mesh in more details. In the **View** menu select **Cut Plane** to pop up the **View Cut Plane** dialog.

The orientation of the cut plane is defined by its normal vector, given by the angles **Phi** and **Theta**. If both angles are zero the normal vector corresponds to the x axis of the simulation cell – the axes can be switched on with **View - Coordinate Axes** – and the cut plane is parallel to the yz-plane. Angle **Phi** determines the rotation of the coordinate system about the z-axis (a positive angle meaning an anti-clockwise rotation about z if z is pointing towards the viewer). By this rotation the former x- and y-axes are rotated by the same angle into the new axes x' and y' respectively. The angle **Theta** specifies an anti-clockwise rotation of the coordinate system about the new y'-axis. After the two rotations the normal vector of the cut plane corresponds to the x'' axis.



To scan the mesh with the cut plane the distance to the center of the simulation cell can be specified (selecting the toggle button **Distance from center**) or a point on the cut plane (selecting **Through point**). The distance from the coordinate center is specified in percent, meaning that 100 % is the biggest possible distance of the plane from the center so that at least one point of the plane is still within the simulation cell. This center distance is therefore depending on the orientation of the cut plane.

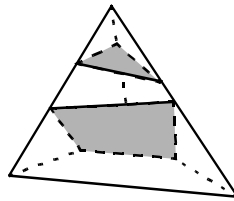
By specifying a point of the cut plane (the point must be within the simulation cell) the plane will be rotated about this point by changing the orientation angles. If the cut plane is once not visible any more it may be that it goes through one of the corners of the cell. To get it back into the latter you can always press the **Reset** button.

With complex meshes the redrawing of the plane cut may take some time. For this reason it is possible to switch automatic redrawing of the plane on or off by ticking the **Automatically redraw** field in the cut plane dialog. If the cut plane is shown it is still possible to rotate and zoom the simulation cell by clicking in the graphics viewer and using the cursor keys as described in [moving the model](#). If **Automatically redraw** is off, the cut plane can be redrawn any time by pressing the **Redraw** button on the bottom of the dialog window. If **Automatically redraw** is on, the **Redraw** button has no effect.

### 2. 11. 1. Cut Plane of Periodic Mesh: Geometric Considerations

Volume meshes in *GridMorph* are periodic, obeying the **periodic boundary condition**. The cut plane view shows cuts of all tetrahedrons and their images if their centers of mass are within the simulation cell.

The periodicity of the mesh can be observed in the cut as well if the cut plane is perpendicular to one of the x, y, or z axis. A plane cut of a tetrahedron may result in different polygon shapes such as a point, a line, a triangle or a quadrilateral (see figure below). By small shifts of the cut plane any of these polygon shapes can be changed into one of the others and an acute triangle does not mean that the corresponding tetrahedron is also acute.



### 2. 11. 2. Visualization of Property Fields

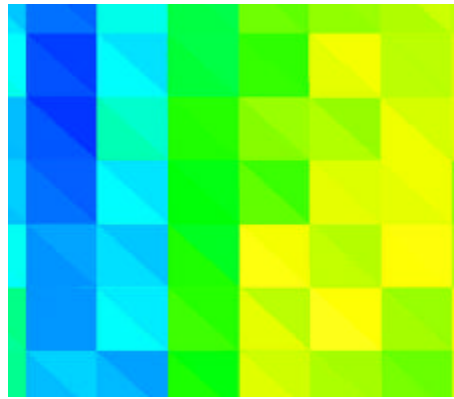
In **View - Cut Plane** the local properties from the property field files can be visualized. For this purpose, however, the properties have to be computed first according to the description in **2. 8. Calculating properties of the composite**. Due to the big amount of data these files are loaded upon request, which may take a few seconds. Properties can be visualized if they have been previously **stored during a Solver** run.

The color coding for each tetrahedron element can be adjusted in a histogram. The histogram also shows the magnitudes for ranges of the property, i. e. the number of elements with the same property. Using the histogram you can determine quite accurately the amount of elements of a certain magnitude of the displayed property by adjusting the minimum and maximum of the histogram's range. The minimum and maximum values must not exceed the effective minima and maxima of the property values. With the number of bins (minimum 3) you can adjust the resolution of the histogram.

### 2. 11. 3. Checkerboard effect in local property display

When visualizing a property field in the Cut Plane mode you may notice a picture like in the following figure: instead of getting unicolor squares these may consist of two triangles with obviously different properties. This so-called

checkerboard effect appears is due to the fact that *GridMorph* uses not a hexagonal solver but a solver based on a tetrahedral mesh:



Although the grid elements are cubic boxes they consist of six tetrahedrons each and these tetrahedrons may experience different strains or fluxes even though they have identical local properties. For this reason it is recommended to average local properties of the six tetrahedrons within the same grid element.

Despite of these local deviations of the properties it was shown through comparison of such calculations with measurements that the overall properties (stored in the output file) are highly accurate ([Goussev 2000](#)) ([Gusev 2004](#)).

## 2. 12. Quitting the program

After saving the project with **Project - Save** you can safely leave *GridMorph* or start a new project. To quit the program select **Quit** from the **Project** pull-down menu.

## 3. My second project

In the first example project you learned all about the user interface and visualization modes of *GridMorph* as well as how to calculate the composite properties. Now you will learn how to create a composite from existing phase data files, how to design material properties and how to assign them to the different phases of a composite.

### 3. 1. Creating a new project

For this tutorial you can use the same phase data files as in the first example or if you have output files of **C<sup>2</sup>-MesoDyn**<sup>®</sup> by *Accelrys*<sup>®</sup> available, you can use those. The files must correspond to the format described in [Chapter 5. 7. 2.](#)

First you have to start a new project: from the **Project** pull-down in the main window choose **New**. Type the desired name for your project in the “Filename” field of the file selection box. If you want to change the directory select the appropriate directory name in the “Directories” box and click the **Filter** button. In the “Files” box you may find a list of the log files (with extension `log`) of already available projects. This extension will be added automatically to your project name, so you do not have to type it explicitly.

After typing the project name in the filename field click the **OK** button. The main window will immediately show the name of the project and the current working directory. Before any grid information is defined the window shows “Arbitrary Grid Morphology in 0 x 0 x 0 grid” and neither a configuration file nor analysis data is available. The number of grid nodes and mesh elements is initially also zero.

### 3. 2. Import an arbitrary grid morphology

If phase files for an arbitrary morphology are available in the proper format the **Project - Import** command can be used to put the morphology together and to define the properties of every phase. Initially the number of phases – this equals the number of phase files to be imported – is specified together with the desired averaging method:

#### 3. 2. 1. Averaging method

With arbitrary morphologies each grid element can be a combination of different phases (see also the Figure in the [Preface](#)): for example grid element  $j$  can consist of 20 % of material A, 70 % of material B and 10 % of material C as specified in the [phase files](#) for materials A, B, and C.

Two methods can be chosen for calculating the properties of each grid element that is a combination of different phases: **serial (arithmetic) averaging** or **parallel averaging**. These names come from the meaning of the method in electronic circuits, where resistors can be connected in series or in parallel. The method is stored in the [configuration file](#).

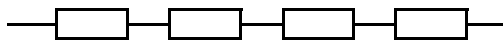
### Serial averaging

arithmetic averaging with weights  $w_i$

$$\rho_j = \sum_{i=1}^M v_{ji} w_i \rho_i$$

( $j$ : element index;  $i$ : phase index;  
 $M$ : number of phases)

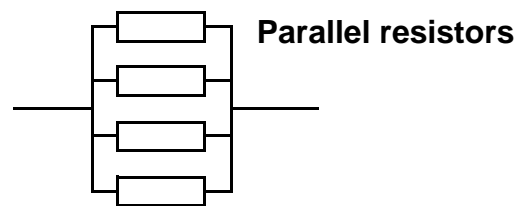
#### Resistors in a line (serial)



### Parallel averaging

with weights  $w_i$

$$\frac{1}{\rho_j} = \sum_{i=1}^M \frac{v_{ji} w_i}{\rho_i}$$



The weights  $w_i$  are used to weight the phases differently. If one of the property values  $\rho_i$  is zero, this property is set to zero for all grid elements.

**There are no clear rules available yet about which method should be used for multi-component mixtures and MatSim does not claim scientific correctness for any of the two averaging methods.**

## 3. 2. 2. Load phases and assign materials

In the lower part of the **Project - Import** dialog there is a table of the phases, their materials and volume fractions, which is initially empty. The buttons for the phase file name (starting with the first entry) must be pressed for a file selection box to appear. The material for each phase can be assigned by selecting a material in the pull-down menu at the bottom of the import dialog and pressing the **Material** button of the phase to which this material should be assigned.

Phases are numbered from zero and it is important to start the definition of the 0<sup>th</sup> phase first and define the phases in increasing order.

After importing all phase files and assigning a material to each phase the **Save** button must be pressed in order to store this configuration on the disk. Make sure that the total volume fraction is 100 % otherwise you might have loaded the wrong set of phase files and the property results may become unpredictable.

Once the project is saved it is not possible to change the phases through the user interface. It will then be necessary to modify the configuration (*cfg*) file in a text editor by strictly keeping the formatting rules in Sections [5. 5. Materials Database File](#) and [5. 6. Configuration File](#).

### 3. 3. Load an arbitrary grid morphology

If the above file suite has been put together it will be available until the program is terminated or another project is loaded. This project can be reloaded into *GridMorph* by selecting **Project - Open** and choosing the corresponding `log` file.

With the **View - Options** you can adjust the sizes of grid nodes, thickness of the mesh lines and the simulation cell. **View - Background** allows you to find the optimal background color to give best contrast of the model.

### 3. 4. Editing material properties of the components

Before you can calculate properties of the composite it is necessary to specify the properties of each component. If you do not want to rely on the materials provided you can design your own by choosing **Edit Material Properties** in the **Materials** pull-down. To define a new material select **Not yet defined** in the material option menu. If you want to modify an existing material select it from the current database or – if the database is empty – load the respective material file.

In the “Descriptor” field you first write a short description of your material, specify its density and the crystal symmetry class. The latter can be selected out of nine different classes with the option menu:

- **isotropic materials**, defined by two independent elastic constants for the mechanical properties and one parameter for each of the other physical properties.
- **cubic** symmetry, defined by three independent elastic constants and at least one parameter for each of the other properties.
- **hexagonal** or **fibre symmetry** for transversely isotropic materials is defined by five independent elastic constants and at least one parameter for each of the other properties.
- **orthorhombic** symmetry: nine independent elastic constants and one to six constants for each of the other properties.
- **trigonal**: seven independent elastic constants and one to six constants for each of the other properties.
- **tetragonal**: seven independent elastic constants and one to six constants for each of the other properties.
- **mono2clinic** (monoclinic symmetry with diad parallel to the y axis): 13 independent elastic constants and one to six constants for each of the other properties.
- **mono3clinic** (monoclinic with diad parallel to z): 13 independent elastic constants and one to six constants for each of the other properties.
- **triclinic materials** (fully anisotropic) are specified by 21 independent constants for the mechanical properties and six independent constants for each of the other properties.

It is possible that a material is isotropic for one property but anisotropic for another one. In this case the most complex crystal symmetry class should be chosen.

To specify the parameters for certain physical properties of the material the respective check boxes must be selected and a new dialog window pops up in which the specific property has to be defined. If the check box contains already a tick mark this property was specified earlier for this material.

Each of the property dialogs have four activity buttons: **Apply** (in this case the properties are stored in the local material database), **Reset** (undo the changes done in the current dialog), **Close** (the changes are not stored), or **Undefine** (an already defined property will be removed from this material). By pressing **Undefine** the tick mark in the check box for this property will disappear. If you press **Close** without first pressing **Apply**, the property will not be modified for this material – an appropriate warning will be displayed to confirm your action.

After entering the required properties according to the description below it is necessary to save the changes in the material file by pressing **Save** (to save the material under the same name) or **Save As** (to save it under a new name). If a **Not yet defined** material is to be **Saved** or if an existing material is to be renamed pressing **Save As**, the file selection box is popped up in which you enter the appropriate file name (the extension `mat` will be automatically added by the program). The file name will be used as the name of the material itself as it will appear in the materials database of your project.

Clicking **Close** discards the **Edit Material Properties** dialog without saving the last changes of the properties. If you did not save the property or if you switch to another material without saving the current one first, a warning is displayed before the dialog disappears or before the data for the new material is displayed.

As soon as the material file is saved it is available in the internal material database and can be **assigned to the desired phase**. If you want to modify another material you have to select **Load** in the material option menu and choose the appropriate material file from the file selection. If this material is already loaded in the database the warning message will appear whether to load it again. If you save the modified property (**Save As**) with the same name, the materials database and the properties of the phases consisting of this material will be updated.

### 3. 4. 1. Editing elastic constants

Depending on the symmetry class different parts of the stiffness or compliance matrix are editable (the white fields). The non-editable fields contain either a constant value or they are dependent on one or two other parameters. Such dependent fields are always updated when the mouse pointer leaves the input field of an underlying parameter.

For the mechanical properties you can specify up to 21 independent elastic constants or elastic compliances for fully anisotropic materials (triclinic crystal class), which are shown in the upper right part of the symmetric stiffness matrix

(parameters  $c_{ij}$ ). To display the reciprocal compliance matrix with parameters  $s_{ij}$  you have to click the toggle button on the left of the triangular matrix.

The matrix indices of the elastic constants correspond to the main axis directions: index 1 stands for x, 2 for y, and 3 for z. The internal coordinate system of the phases correspond to the simulation cell's coordinate system.

For isotropic materials two constants are sufficient to describe the elastic behaviour. These are the Young's modulus  $E$  [units: GPa] and Poisson's ratio  $\nu$ , the bulk  $K$  [GPa] and shear  $G$  [GPa] moduli, the elastic constants  $c_{11}$  and  $c_{12}$  [both in GPa], or the elastic compliances  $s_{11}$  and  $s_{12}$  [1/GPa]. They are connected by the following equations for isotropic materials:

$$E = \frac{3KG}{K + G/3} = \frac{(c_{11} - c_{12})(c_{11} + 2c_{12})}{c_{11} + c_{12}} \quad \nu = \frac{3K - 2G}{6K + 2G} = \frac{c_{11}}{c_{11} + c_{12}}$$

$$K = \frac{E}{3(1 - 2\nu)} = \frac{c_{11} + 2c_{12}}{3} \quad G = \frac{E}{2(1 + \nu)} = \frac{c_{11} - c_{12}}{2}$$

$$c_{11} = \frac{E(1 - \nu)}{(1 - 2\nu)(1 + \nu)} \quad c_{12} = \frac{E\nu}{(1 - 2\nu)(1 + \nu)} \quad c_{44} = \frac{E}{2(1 + \nu)}$$

$$c_{11} = K + \frac{4}{3}G = 2\mu + \lambda \quad c_{12} = K - \frac{2}{3}G = \lambda \quad c_{44} = G = \mu$$

In the last equations  $\lambda$  and  $\mu$  are the first and second Lamé constants, respectively.

### 3. 4. 2. Editing thermal expansion coefficients

The linear thermal expansion coefficients,  $te$  [units:  $K^{-1}$ ], are given by six independent elements of the symmetric  $3 \times 3$  expansion tensor. For isotropic materials only one coefficient has to be specified.

### 3. 4. 3. Editing thermal conductivity

The thermal conductivity,  $tc$  [units:  $Wm^{-1}K^{-1}$ ], is given by its symmetric  $3 \times 3$  tensor. For isotropic materials only one coefficient has to be specified.

### 3. 4. 4. Editing dielectric constants

The dielectric constants,  $dc$  [dimensionless], is given by its symmetric  $3 \times 3$  tensor. For isotropic materials only one coefficient has to be specified.

### 3. 4. 5. Editing electric conductivity

The electric conductivity,  $ec$  [units: Siemens/m], is given by its symmetric  $3 \times 3$  tensor. For isotropic materials only one coefficient has to be specified.

### 3. 4. 6. Editing gas permeability

The permeability of gases (or liquids),  $gp$  [units: barrer], is given by its symmetric  $3 \times 3$  tensor. For isotropic materials only one coefficient has to be specified.

### 3. 4. 7. Editing gas diffusivity

The diffusivity of gases (or liquids),  $gd$  [units:  $\text{cm}^2\text{s}^{-1}$ ], is given by its symmetric  $3 \times 3$  tensor. For isotropic materials only one coefficient has to be specified.

### 3. 4. 8. Editing material colors

Only for visualization purposes it makes sense to specify colors for a material. *GridMorph* allows to edit all parameters that define the appearance of three-dimensional objects. Inside a grid-based composite morphology however, the surface color parameters are not relevant and are therefore ignored so far. It is not straight-forward to set these parameters in order to make an object appear in “natural” colors. Please refer to the computer graphics literature for complete guidance.

Transparency can be obtained by setting the alpha values smaller than 1. However, the OpenGL graphics format (using a depth buffer for RGB colors but not for alpha values) can not always deal properly with transparent materials.

While editing material colors a small viewer appears containing two spheres of different resolutions. After pressing the **Apply** button the material must be **Saved** in order to see the effects of the changes in the main viewer.

## 3. 5. Tips and tricks to calculate related properties

In addition to the properties that can be entered straight using *GridMorph*'s user interface there are other physical properties that can be calculated by applying some tricks.

### 3. 5. 1. Liquid permeability and diffusivity

The calculation of the transport properties of liquids works in the same way as for gases. The difference is usually that liquids diffuse much slower than gases and therefore the liquid diffusion is often specified in  $\text{mm}^2\text{s}^{-1}$ . Since *GridMorph* expects values with units  $\text{cm}^2\text{s}^{-1}$  the numbers have to be converted appropriately.

### 3. 5. 2. Refraction index

The refraction index of a medium,  $n_m$ , is a wave property and related with the dielectric constant as

$$n_m = \frac{c_{\text{air}}}{c_m} = \sqrt{\frac{\epsilon_m}{\epsilon_{\text{air}}}} = \sqrt{\epsilon_m}$$

where  $c_{\text{air}}$  and  $c_m$  are the speeds of light in air or the medium, respectively. Since the dielectric constant of air is 1.0, the refraction index of the medium is equal to the square root of the dielectric constant of the medium.

### 3. 6. Reorganizing the Materials Database

After editing or creating many different materials, you may get an error message that the materials database is full. In this case you should do **Materials - Reorganize Database**: with this command all materials that are not assigned to a phase are discarded from the internal database. No material files will be lost because they are available on the disc as a precondition for acceptance in the database.

The **materials database file** (with extension `mdb`) is not affected by this command unless you perform a **Project - Save** operation. Upon **Save** the materials database is always automatically reorganized.

### 3. 7. Assigning material properties to phases

Properties of a composite can only be obtained if the same properties of all phases are defined. Therefore you have to assign material properties to each phase. You can assign material properties while importing external grid data with **Project - Import** or you can do it all at once by opening the **Materials - Assign to Phases** dialog window.

This popup dialog contains a list of the phases of the composite starting with zero, which is the first phase. For each of them the material type and the phase file name are displayed (or **Not yet defined**, if no material has been assigned).

If the list of phases does not fit in the dialog window you can scroll it with the scroll bar on the right. To select one of the phases simply click on the corresponding line; to select a series of phases click on the first and move the mouse down to the last without releasing the mouse button. Alternatively you can click on the first phase, press the Shift key, and then click on the last phase that you want to select. To select or deselect multiple phases you can press the Ctrl key when clicking a phase. The Shift and Ctrl keys can also be combined.

After finishing your selection you choose a material from the options menu on the bottom. By default the button shows **Not yet defined**. As soon as you click this button the option **Load Material** appears and usually a list of materials from the database. If you find the desired material in the list select it or choose **Load Material** to load another material into the database. Make sure that the option button shows the materials name you want to assign to the selection. Then simply click **Assign** and the previously marked list of phases will be updated appropriately.

Before clicking the **Close** button, make sure that all phases have a material assigned. Otherwise you will get error messages from the Solver.

### 3. 8. Calculating composite properties

With the **Solver** you are already familiar. Just a few hints before you start it:

- **Save the project** before you continue to make sure you do not lose the project data in case of an unexpected interrupt of *GridMorph*.
- **Assign material properties** to all phases.
- Check if for the component materials all properties are defined that you want to calculate for the composite. Use **Materials - Edit Material Properties**.

**Now you proceed as in your first project...**

## 4. Information, warning, and error messages

Here you find all possible messages produced by *GridMorph*. Information and error messages appear in an orange pop up window, which will disappear if you press the **OK** button. Information messages usually are less severe than error messages and the cause can be fixed quickly. Warning messages appear in a green window and require a decision between **YES** or **NO**.

In the explanations below the messages produced by *GridMorph* are printed in `Courier` font. Some of them contain texts in `< >` brackets, which stands for arbitrary names or descriptions. The number (`#`) symbol stands for any number related to the message.

Below each message is the explanation what it means and hints how to avoid the error or what to do with a warning. Bold face expressions are operations you can directly apply in *GridMorph*, such as **Project - Open**, which means to click the “Project” pull-down in the main menu and select “Open”.

### 4. 1. Error and information messages

These messages appear in an orange window and require confirmation by pressing the **OK** button.

#### `<filename> not found`

The file `<filename>` could not be found.

If `<filename>` corresponds to what you just typed in a file selection box, correct the spelling of the filename;

Otherwise: if `<filename>` has the extension `mat` you should check the contents of the materials database file (with extension `mdb`) of you current project. There is a reference to a material file that does not exist any more. You can either recreate this material file or manually change the entry in the `mdb`-file.

In all other cases the file is part of the project and it was available at the moment you started the current project but now it is not available any more. Did you or somebody else remove it or do you have a network problem?

#### `Allocation of # bytes for <name of data structure> failed`

Your problem is too big to fit into the computer’s memory.

You have to decrease the size of your system by using a smaller grid.

Another option is to increase the swap space of your computer or to plug in additional RAM (Random Access Memory).

#### `Assign a material to phase #`

Phase `#` has no properties allocated; therefore the solver cannot calculate the composite properties.

Check the **Materials - Assign** dialog for phases whose material is “not defined”.

**Could not create backup of <filename>**

Probably there is a network error or the file was accidentally removed while *GridMorph* was running. This error is not severe for the current project but maybe you lost some data of a previous stage.

**Define phases in increasing order.**

In **Project - Import** for an arbitrary morphology the phase files must be specified starting with the first phase (number zero).

**Double entry in materials database file <filename>, line #**

The name of a material file appears twice in the materials database file (with extension `mdb`). The second occurrence is in the specified line of the `mdb` file.

Edit the `mdb` file manually and type in another valid material filename (with extension `mat`) at the erroneous line. After loading the project again you can use **Materials - Assign** to load the material which you wanted to assign to the phases.

**Elastic constants must be specified for <material name>**

Elastic constants are not defined in the file of the specified material (the file with extension `mat`), therefore the solver cannot calculate elastic properties of the composite.

Use the **Materials - Edit Properties** dialog, **Load** the material in question, specify its elastic constants, and **Save** the material under the same name.

**End of file <filename> reached at line #**

The specified file is probably corrupted or some essential information is missing.

Open the file in an editor. If it looks reasonable to you add a blank line at the end and try to open it again in *GridMorph*. If it still does not work, remove the file or, if there is a backup (the same filename with extension `_bk`) available, overwrite the corrupted file with its backup (by removing the extension `_bk`).

**Error closing <filename>. Disk full?**

The disk is full, your disk quota is exceeded, or a network error occurred.

If the disk is full or your quota limit reached you may have to remove some older backup files (those with extension `_bk`).

**Error reading <filename>**

The file system was not able to read the already opened file, probably due to a network error.

Try it again, if the error persists, check the file for illegal characters. These are usually characters which can not be typed on the keyboard.

**Error writing <filename>. Disk full?**

The disk is full, your disk quota is exceeded, or a network error occurred.

If the disk is full or your quota limit reached you may have to remove some older backup files (those with extension `_bk`).

**Filename <filename> too long! Max <# of characters>!**

You specified a filename with too many characters.

Depending on your operating system there is a limitation of the number of characters that can be used for the **filename including pathname**.

**Format error in configuration file <filename>, line #!**

The **configuration file** (with extension `cfg`) does not comply with the expected format (probably after manual changes).

Correct the formatting of the line in question according to the specification.

**Format error in materials database file <filename>, line #!**

The **materials database file** (with extension `mdb`) does not comply with the expected format (probably after manual changes).

Correct the formatting of the line in question according to the specification or remove the `mdb` file and reassign materials to the configuration with **Materials - Assign**.

**Format error in material file <filename>, line #!**

The **material property file** (with extension `mat`) does not comply with the expected format (probably after manual changes).

Correct the formatting of the line in question according to the specification or remove the `mat` file and redefine the properties of this material with **Materials - Edit**.

**Format error in phase file <filename>, line #!**

The specified **phase file** for the morphology contains a syntax error. Correct it according to the **phase file format**.

**Format error in script file, line #!**

The syntax of the script file is wrong. There might be an unexpected end-of-line. Correct the syntax according to the rules for **script commands**.

**Grid morphology can not be imported into existing project! Do Project-New first.**

**Project - Import** for an arbitrary morphology works only in a new project environment.

**Illegal argument for parameter <name> in script file, line #!**

The parameter value for a **script commands** as described in chapter 7. is wrong, e. g. a float value where an integer is expected.

**Illegal filename <filename>**

The specified filename is not allowed in this operating system.

For **filenames** you should only use letters, numbers, and underscores.

**Illegal keyword <wrong command> in script file, line #!**

A command keyword is misspelled. Use **script commands** as described in the corresponding chapter 7.

**Illegal material assignment to phase for this morphology!**

Only phase properties can be modified. See also the detailed description of parameters for this **script command**.

**Illegal phase number # in Material-Assign statement in script file <filename>!**

The phase number is out of range: phases are numbered from 0 to  $n_{Phases}-1$

**Illegal or missing name for material in script file <filename>, line #!**

A material name was either not specified or misspelled, i. e. it could not be found in the database. Check if the file exists (and has the extension `mat`) and specify in the script file the full file name including the relative path name if it is not in the directory where *GridMorph* was started.

**Illegal parameter name <name> in script file, line #!**

A parameter name for a **script commands** is misspelled or used in the wrong command context. The commands and their corresponding parameters are described in chapter 7.

**Illegal syntax for parameter <name> in script file, line #!**

An unexpected end of line appeared after a parameter in a **script command**.

**Inconsistent grid dimension (#) in phase file <file name>!**

The specified **phase file** for the **arbitrary grid morphology** contains not as many numbers per line, lines, or blocks of lines as specified in the first line of the file.

**Input error in character field: <field name>**

Illegal characters or nothing was entered in the **specified field**.

Some input fields require an entry; do not use composed characters.

**Input error of floating-point number: <erroneous number>**

You used a **wrong format** to specify a number.

A floating-point number must be of the form  $s\#.es\#$  where  $s$  stands for an optional sign (+ or -),  $\#$  stands for a sequence of digits without spaces, and  $.$  stands for the decimal point (commas are not allowed).  $e$  and everything thereafter is optional. The number after the letter  $e$  specifies the exponent of base 10 by which the number before  $e$  is multiplied.

**Input error of integer number: <erroneous number>**

You used a **wrong format** to specify a number.

An integer number must be of the form  $s\#$  where  $s$  stands for an optional sign (+ or -) and  $\#$  stands for a sequence of digits without spaces.

License expired # days ago!

License expires in # days!

Your license for using *GridMorph* expired or will expire soon.

**Order a new license**, which includes full support and free upgrade service.

**Materials database file <filename>: entries should be consecutively numbered in line #**

The entries in the **materials database file** (with extension `mdb`) are wrong.

Correct the formatting of the line in question according to the specification or remove the `mdb` file and reassign materials to the configuration with **Materials - Assign**.

**Materials database is full with # elements! Saving is possible but material will not be available for current project. Perform Materials - Reorganize Database.**

The number of different materials for this project was exceeded.

Do **Materials - Reorganize Database**. If it does not help, remove the materials database file (with extension `mdb`) of this project and do **Materials - Assign** again.

**Mismatch between materials name and filename in materials database file <filename>, line #!**

The name of a material is specified by the name of its properties file (with extension `mat`).

Do not rename `mat` files that are used in any of your projects. If the error persists, remove the **materials database file** (with extension `mdb`) of this project and do **Materials - Assign** again.

**Missing crystal class (CC) in file <filename>!**

A **material properties file** (with extension `mat`) must have an entry that specifies its crystal symmetry class. If this entry, which is introduced by "CC", is missing, the file can not be used.

Redefine the material with **Materials - Edit Properties**.

**Missing material for phase #. Select Material and press Assign.**

In **Project - Import** for an arbitrary morphology every phase must have a material assigned.

**Missing phase definition #. Press Load Phase.**

In **Project - Import** for an arbitrary morphology every phase must be defined by a phase file.

**Missing <property>! Used for phase #.**

The <property> (e. g. thermal expansion coefficient, electric conductivity) is not available for the specified phase. Check, which material is used for it with **Materials - Assign** and go to **Materials - Edit Mat. Properties** to define the requested property for this material.

Another possibility is to tell the solver not to calculate this property for the composite by unmarking the property button in the **Solver** window.

**No analysis data available!**

The output file (with extension `out`) is missing for this project.

Assign material properties to all phases of your composite and run the **Solver**.

**No composite material available!**

Composite properties have not been stored in the material file (with extension `mat`).

Run the **Solver** first and store the output with **Materials - Save composite material**.

**No elastic constants are defined for <material>!**

In the **Material - Analyze Elasticity** dialog it is possible to analyze the stiffness tensor of triclinic materials. However, if the selected material does not have elastic properties assigned, this information message appears.

**No field data available for <property>! Run Solver first.**

In View - Cut Plane you tried to visualize property fields of a property whose field has not been stored in a previous run of the Solver.

Run the **Solver** and do not forget to tick the **Store full property fields** button.

**No project is defined!**

For most operations in *GridMorph* you must have a project defined.

Do **Project - Open** to open an existing project or **Project - New** to create a new one.

**No read/write permission for <filename>**

The operating system does not allow you to access the specified file.

Check if the file belongs to another user or if it is read/write protected.

**Nothing to save!**

No project is loaded or nothing has been modified since last save.

**Please fill in all input fields correctly!**

Some input field in the currently active dialog window contain wrong values.

Move the mouse pointer over all **input fields** (white) to find out which of them has to be corrected.

**Some properties require elastic constants! Please add them now or later.**

To calculate properties such as thermal expansion coefficients it is necessary to know the mechanical properties given by the elastic constants. Specify those for the currently used materials in **Materials - Edit Properties**.

**Undefined material (#) in configuration file <filename>.**

**Assuming no material.**

The specified material's identifier in the **configuration** (`cfg`) file does not exist in the **materials database** (`mdb`) file. *GridMorph* simply ignores this identifier

and assumes that the inclusion or cell with this material's identifier is not defined yet.

Use **Materials - Assign** to assign valid properties to all phases.

**Unknown property <property> or no property specified!**

To start the **solver** at least one property button must be checked or the "prop" parameter must be specified for the Solver command in the **script file**.

**Wrong number of parameters in line # of <material file>**

There are not enough properties parameters in the specified line of the **material file**.

Remove the `mat` file and regenerate it by doing **Materials - Edit**.

**Wrong number: <erroneous number>! Enter value <allowed range>**

Most numbers must be within a certain limit.

Choose **another number** within the allowed range.

**Wrong or missing keyword in material file <filename>, line #**

Each property of materials is defined by a keyword and the corresponding parameters. The current material file is syntactically incorrect.

Remove the `mat` file and regenerate it by doing **Materials - Edit**.

**Wrong value for <property> in material file, line #**

A parameter of the specified material property is beyond the allowed limits.

Remove the `mat` file and regenerate it by doing **Materials - Edit**.

**Your license does not allow this operation. Please contact your support center.**

If you did not purchase the full license there will be packages that are not supported in your license scheme. To get more packages, contact your vendor.

## 4. 2. Warning messages

This type of messages appear in a green window. They require an immediate answer (usually **YES** or **NO**) from the user.

**<material name> already loaded! Overwrite?**

A material with this name is already in the internal materials database. This messages occurs if you try to load a material properties file, which was already loaded before.

**Current project was not saved. Quit anyway?**

Some parts of the current project have been modified but not saved.

If you do not want to loose any data of the current project do **Project - Save** prior to **quitting GridMorph**.

**Data file was not saved. Close anyway?**

After editing a data table, the information should be written in an appropriate file otherwise it will be lost upon closing the edit dialog.

**Do you really want to quit?**

Appears upon selecting **Project - Quit**.

**Do you really want to undefine this property for <material>?**

In **Materials - Edit Properties** you may want to delete a single property from the database of a material.

**File <filename> already exists. Overwrite?**

If you try to save a material file or a project with the same name as an already existing file, this message appears.

**Material with zero density! Is that correct?**

By defining new materials in **Materials - Edit Properties** you should specify the density of the material. If it is not available you can omit it but you have to take into account that the total density will then not be properly computed.

**Project <project name> already in memory. Dismiss without save?**

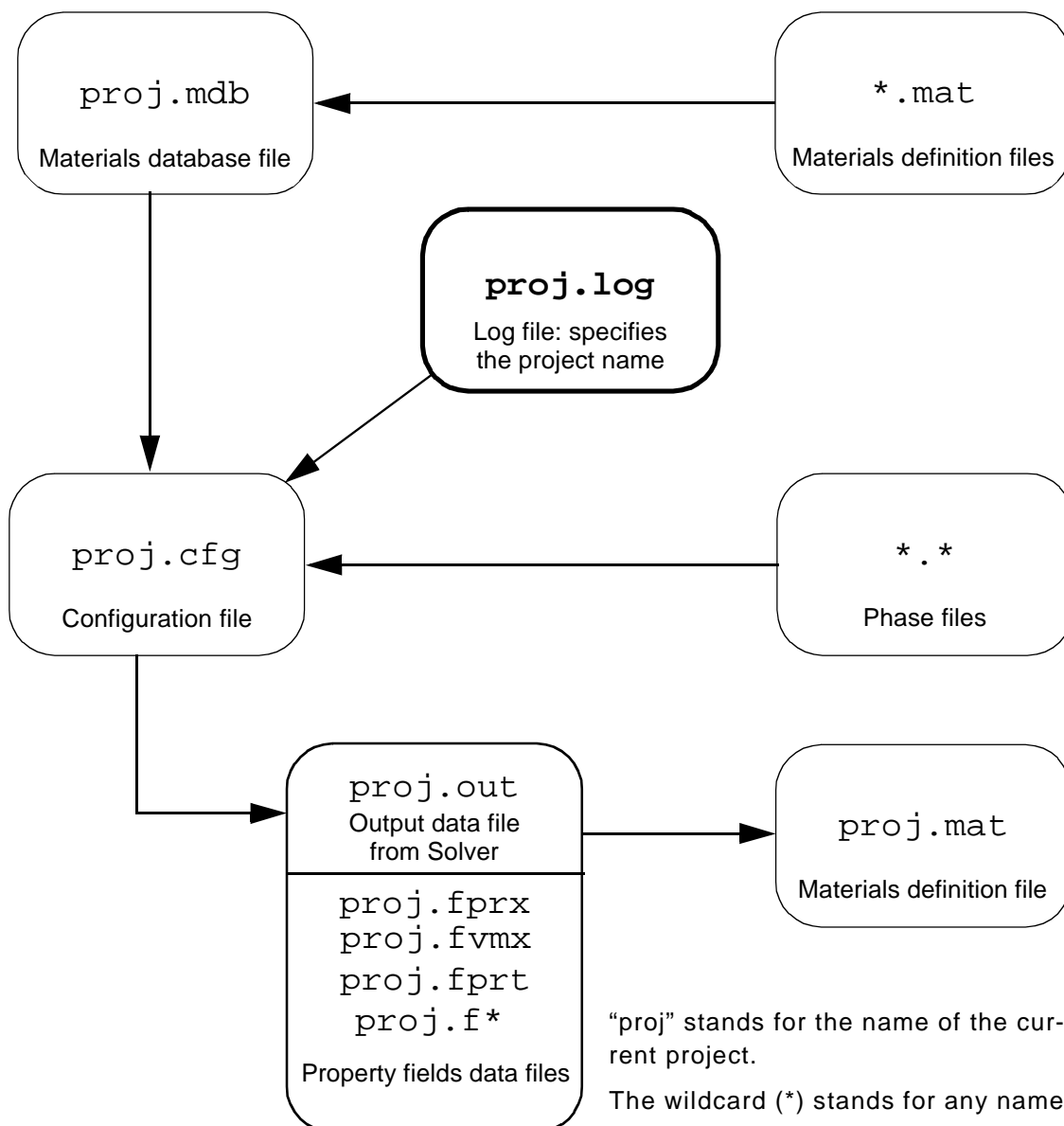
A project with this name is already loaded.

## 5. File Formats

Warning: do not modify files that were generated by *GridMorph* unless you know exactly what you are doing. Syntactically incorrect files can not be read by *GridMorph* any more.

### 5. 1. File Dependencies

*GridMorph* uses the following file suite for input or output. They depend on each other as shown in the figure. The central file of the suite is the `log` file, which determines the name of the project. If for whatever reason the log file is not available any more, an empty file with the corresponding name can be used in order to recover the other files of the project..



## 5. 2. General conventions

The following conventions apply to all files created by *GridMorph*:

- File names (including directory path names) may contain the following characters:  
A-z, 0-9, ., , \_ + - = \$ % / ~ \*
- Input data fields must be delimited by space or tab
- The syntax of the file formats is specified in a type of Bacchus-Naur formalism: Identifiers are defined by the ':=' assignment. "" (quotes) enclose exact strings. The selection operator '|' separates different options. The content of square brackets [ ] is optional while the content of curly brackets { } can be repeated many times (or never). Round brackets ( ) are used for organizing expressions. The carriage return symbol '<' stands for a new line. Variables are printed in italics and must be replaced in the file by a number or a string of the specified type, which is defined after the colon ':'.
- Do not expect the file formats be logical. However, the following explanations should help you to understand the file formats.

## 5. 3. Script File

See the detailed syntax description for the *GridMorph script language* in the corresponding chapter.

## 5. 4. Log (Project) File

Filename: `project.log` (where `project` is an arbitrary name for the current project).

Contains information about the current project such as names of files involved (creation, opening, saving, backup creation) and history log information that appears in the main window (including error and warning messages). The name of the project is given by this file. No **backup** is produced of the log file, therefore it can become long if a project was reopened many times. In this case it can be emptied by the user.

If for whatever reason the log file is lost, an empty file with the corresponding name of the `cfg` and `mdb` files can be used in order to recover other files.

After long operations the computation time is printed in the log file. The basic results of the solvers are also printed here and in the main window.

There is no particular formatting of this file since it is not machine readable.

## 5. 5. Materials Database File

Filename: `project.mdb` (where `project` is the same name as for the information file in `project.log`)

Contains a list of material files (their names with extension `mat` and locations) with indices that are used in other files such as `project.cfg`. The database is reorganized before saving it, thus it contains no more than the materials that are really in use in the current project.

### 5. 5. 1. Content

The `mdb` file contains an indexed list of material files, which are given by their name and the full path name where the materials data itself can be found. The index is starting at zero.

The materials name must correspond to the base name of the file, i. e. the file name without path and extension `mat`.

This file is updated whenever a new material is loaded within the project or a new material is created by the materials editor.

### 5. 5. 2. Format

```
{ index           : integer
  materials-name  : string
  filename        : string           <
}
```

## 5. 6. Configuration File

Filename: `project.cfg` (where `project` is the same name as for the information file in `project.log`).

This file is used to define the names of the phase files, their properties and the averaging method. For each phase  $X$  in the current configuration there must be a **phase file** containing the fractions  $v_{j,X}$  of this phase for each grid element  $j$ .

### 5. 6. 1. Content

The first line is an optional comment line (starting with "#"), which contains – if created by the program – the version of the program that has stored the configuration (e. g. `GridMorph 1.0`). The next line (starting with ">M") declares the number of phases and a code for the averaging method: 0 stands for **serial averaging**, 1 for **parallel averaging** as described in [Chapter 3. 2. 1.](#)

Then follow the descriptions of every single phase: each of these lines starts with the keyword “File”, the number (in increasing order, starting with 0) of the phase (corresponding to the entry in the **materials database**, `mdb`) and the **weight** of the respective phase as described in the picture above. The last parameter on this line is the name of the file describing the phase (**phase file**), i. e. describing the density of this phase in every grid element.

### 5. 6. 2. Format

```
[ "#" program_version      : string           <
]
"> M" nPhase avgMethod    : integer         <
{
  "File" phase/matId       : integer (pointer to mat.DB)
    weight                  : float
    filename                : string           <
}
```

## 5. 7. Phase File for Grid Morphologies

The filename can be chosen ad libitum. It is accessed through the “File” entry in the **configuration file** (with extension `cfg`).

### 5. 7. 1. Content

The phase files are usually produced by other software such as **C<sup>2</sup>-MesoDyn**<sup>®</sup> by **Accelrys**<sup>®</sup> ([www.accelrys.com/cerius2/MesoDyn.html](http://www.accelrys.com/cerius2/MesoDyn.html)). For a composite consisting of  $N$  phases,  $N$  phase files are required with identical syntax and parameter description. The phase file format corresponds to the **C<sup>2</sup>-MesoDyn**<sup>®</sup> (**Accelrys**<sup>®</sup>) format. The file starts with the keyword `#GRID`, followed by the dimensions of the grid,  $nx$ ,  $ny$ ,  $nz$ , i. e. the number of cubic grid elements in the x, y, and z directions. The last parameter of the first line specifies the size (*length*) of a cubic grid element.

The *length* parameter specifies the length of the cubic grid element. To reduce floating point precision inaccuracies and to get a nice picture in the viewer, the product of the biggest dimension (maximum of  $nx$ ,  $ny$ ,  $nz$ ) with the *length* should not exceed the range between 0.1 and 10.

After this follow  $nx$  blocks of  $ny$  lines of  $nz$  numbers per line, each number specifying the fraction of this phase in the respective grid element. The minimum size for  $nx$ ,  $ny$ ,  $nz$  is 3; the upper limit is restricted only by the memory capacity of the computer. The  $nx$  blocks are separated by an empty line. Although these numbers are fractions and therefore assumed to have a value between zero and one, they can actually be bigger to represent areas of higher local pressure. With the weight

parameter specified in the **configuration file** the weight of all the fraction numbers in the same phase file can be adjusted appropriately.

The phase files of the same morphology must have identical syntax and therefore identical values for  $nx$ ,  $ny$ ,  $nz$  and the *length*. The sum of the fraction values  $V_{x,y,z}$  of all the phase files of one morphology should be one. However, as described above, it is possible to have a total phase fraction above one if the local pressure is higher than in another area.

### 5. 7. 2. Format

```

"#GRID"
  nx  ny  nz                : integer
  length                       : float           <
V1,1,1 V1,1,2 ... V1,1,nz : float           <
V1,2,1 V1,2,2 ... V1,2,nz : float           <
  :   :   :   :           :   :           <
V1,ny,1 V1,ny,2 ... V1,ny,nz : float           <
                                     (empty line)      <
  :   :   :   :           :   :           <
  :   :   :   :           :   :           <
Vnx,1,1 Vnx,1,2 ... Vnx,1,nz : float           <
Vnx,2,1 Vnx,2,2 ... Vnx,2,nz : float           <
  :   :   :   :           :   :           <
Vnx,ny,1 Vnx,ny,2 ... Vnx,ny,nz : float           <

```

## 5. 8. Materials Definition File

Filename: `matname.mat` (where `matname` is an arbitrary identifier for a certain material).

For each material there is a separate file. These files are related to the current project by the materials database `project.mdb`, where each `mat`-file is specified by an index and its full file name.

### 5. 8. 1. Content

The first two lines are optional comment lines (starting with "#"). The first line contains – if created by the program – the version of the program that stored the materials parameters (e. g. `GridMorph 1.0`). The second line contains a textual description of the material, starting with the key word "REM" (REMark).

The *crystal symmetry class* is introduced by the key word "CC" and defined by a string descriptor for the following classes:

- **isotropic**: 2 independent elastic constants. In the file they are defined by Young's modulus  $E$  [GPa] and Poisson's ratio  $\nu$ . The conversion into other pa-

rameters such as bulk and shear modulus or the stiffness and compliance matrices is performed upon reading the data.

Only one element of the other properties such as linear thermal expansion coefficient ( $te_{11}$  [1/K]) or electric conductivity ( $ec_{11}$  [S/m]) is stored.

- **cubic**: 3 independent elastic constants; 21 lower left elements of the stiffness matrix  $C$  [GPa] and 6 values of the other properties (represented by symmetric 3x3 matrices) are stored in Voigt notation (on one line of the material file).
- **hexagonal** (transversely isotropic): 5 independent elastic constants; 21 lower left elements of the stiffness matrix  $C$  [GPa] and 6 values of the other properties (represented by symmetric 3x3 matrices) are stored in Voigt notation.
- **orthorhombic**: 9 independent elastic constants; 21 lower left elements of the stiffness matrix  $C$  [GPa] and 6 values of the other properties are stored in Voigt notation.
- **trigonal**: 7 independent elastic constants; 21 lower left elements of the stiffness matrix  $C$  [GPa] and 6 values of the other properties are stored in Voigt notation.
- **tetragonal**: 7 independent elastic constants; 21 lower left elements of the stiffness matrix  $C$  [GPa] and 6 values of the other properties are stored in Voigt notation.
- **mono2clinic** (monoclinic with diad parallel to  $x_2$ ): 13 independent elastic constants; 21 lower left elements of the stiffness matrix  $C$  [GPa] and 6 values of the other properties are stored in Voigt notation.
- **mono3clinic** (monoclinic with diad parallel to  $x_3$ ): 13 independent elastic constants; 21 lower left elements of the stiffness matrix  $C$  [GPa] and 6 values of the other properties are stored in Voigt notation.
- **triclinic**: 21 independent elastic constants; 21 lower left elements of the stiffness matrix  $C$  [GPa] and 6 values of the other properties are stored in Voigt notation.

Among the material properties the following are available, depicted by the appropriate key words:

- **ElasticConst**: 21 stiffness constants  $C$  [GPa] arranged in Voigt notation in a lower-left triangular matrix (it is optional to write the full symmetric matrix but the values above the diagonal are simply ignored).
- **Density**: 1 value for the density  $\rho$  [kg/m<sup>3</sup>].
- **ThermalExp**: 1 (for isotropic materials) or 6 (for anisotropic) values for the thermal expansion coefficients  $te$  [1/K]. For anisotropic behavior the six  $te$  values are stored in Voigt notation. If only one value is specified for anisotropic materials, isotropy in the thermal behavior is assumed. These parameters can only be stored if "ElasticConst" are available.
- **ThermalCond**: 1 (for isotropic materials) or 6 (for anisotropic) values for the thermal conductivity  $tc$  [w/(mK)]. For anisotropic behavior the six  $tc$  values are stored in Voigt notation. If only one value is specified for anisotropic materials, isotropy in the thermal behavior is assumed.
- **DielectricConst**: 1 (for isotropic materials) or 6 (for anisotropic) values for the dielectric constant  $dc$  [dimensionless]. For anisotropic behavior the six  $dc$  val-

ues are stored in Voigt notation. If only one value is specified for anisotropic materials, isotropy in the dielectric behavior is assumed.

- **ElectricCond:** 1 (for isotropic materials) or 6 (for anisotropic) values for the electric conductivity  $ec$  [S/m]. For anisotropic behavior the six  $ec$  values are stored in Voigt notation. If only one value is specified for anisotropic materials, isotropy in the electric behavior is assumed.
- **GasPermeability:** 1 (for isotropic materials) or 6 (for anisotropic) values for the permeability of gases and liquids  $gp$  [cm/s]. For anisotropic behavior the six  $gp$  values are stored in Voigt notation. If only one value is specified for anisotropic materials, isotropy in the permeability behavior is assumed.
- **GasDiffusivity:** 1 (for isotropic materials) or 6 (for anisotropic) values for the diffusivity of gases and liquids  $gd$  [cm<sup>2</sup>/s]. For anisotropic behavior the six  $gd$  values are stored in Voigt notation. If only one value is specified for anisotropic materials, isotropy in the diffusivity behavior is assumed.
- **RGBColor:** Red-Green-Blue color codes for the visualization of the materials. The full definition of the OpenGL color coding must be specified. These are four values (red, green, blue, and alpha contribution) for the color, the ambient, diffusive, and specular reflectance attributes, emitted light intensities, and one value for the shininess. All values apart from the shininess must be numbers between 0 (darkest) and 1 (brightest).

All non-mechanical properties are defined by a maximum of six parameters, which are stored in the `mat` file as a Voigt vector.

Properties must have their units specified in square brackets [] just after the property code (for dimensionless properties the square brackets are simply left empty). For the time being the units according to the SI standard are assumed and no transformations take place.

An optional comment can follow the property code and units definition. A more legible description is automatically added by *GridMorph*. Other comments can be placed anywhere in the file on separate lines if introduced with the keyword "REM" (REMark). These comments are not preserved if the file is modified and saved by *GridMorph*.

## 5. 8. 2. Format

```
[ "#" Program_version           : string           <
]
"REM" comments and remarks      : string           <
"CC" Symmetry_Class            : string           <
{ Property
}

Symmetry_Class =
( "isotropic" | "cubic" | "hexagonal" | "orthorhombic" |
  "trigonal" | "tetragonal" | "mono2clinic" | "mono3clinic" |
  "triclinic" )
```

```

Property =
( "ElasticConst" "[" units "]" [comment] <
  c11 : float <
  c21 c22 : float <
  c31 c32 c33 : float <
  c41 c42 c43 c44 : float <
  c51 c52 c53 c54 c55 : float <
  c61 c62 c63 c64 c65 c66 : float <
) |
( "Density" "[" units "]" <
  rho : float <
) |
( "ThermalExp" "[" units "]" [comment] <
  te11 [ te22 te33 te23 te13 te12 ] : float <
) |
( "ThermalCond" "[" units "]" [comment] <
  tc11 [ tc22 tc33 tc23 tc13 tc12 ] : float <
) |
( "DielectricConst" "[" units "]" [comment] <
  dc11 [ dc22 dc33 dc23 dc13 dc12 ] : float <
) |
( "ElectricCond" "[" units "]" [comment] <
  ec11 [ ec22 ec33 ec23 ec13 ec12 ] : float <
) |
( "GasPermeability" "[" units "]" [comment] <
  gp11 [ gp22 gp33 gp23 gp13 gp12 ] : float <
) |
( "GasDiffusivity" "[" units "]" [comment] <
  gd11 [ gd22 gd33 gd23 gd13 gd12 ] : float <
) |
( "RGBColor" [comment] <
  cr cg cb ca : float (GLfloat) <
  ar ag ab aa : float (GLfloat) <
  dr dg db da : float (GLfloat) <
  sr sg sb sa : float (GLfloat) <
  er eg eb ea : float (GLfloat) <
  shine : float (GLfloat) <
)

```

## 5. 9. Output File

Filename: `project.out` (where `project` is the same name as for the project file).

Contains all necessary information about the current system (configuration, surface and volume mesh) including the data (all properties) calculated by the solver. The file is written in an easily understandable way. Some parts however follow a well defined syntax because this data is read again for visualization purpose.

### 5. 9. 1. Content

- Description of the configuration. For each component material the volume fraction is displayed.
- Information about the surface and volume mesh.
- Density of the composite.
- Tensor of elastic constants of the composite in Voigt matrix format:  $C_{ik}$  ( $i, k = 1..6$ ); overall Young's moduli, Poisson's ratios, shear and bulk moduli.
- Tensors for other properties of the composite in  $3 \times 3$  matrix format:  $te_{ij}$  ( $i, k = 1..3$ ) and the average and dispersion of the three diagonal elements of the property tensor.
- The Appendix contains one-to-one copies of all material files (extension `mat`) that were used for the calculation of the properties of this project.

### 5. 9. 2. Format

```
***** Composite Name *****
```

```
Configuration data
```

```
Mesh data
```

```
Density
```

```
Elastic constants
```

```
C11 C12 C13 C14 C15 C16 : float <  
C21 C22 C23 C24 C25 C26 : float <  
: : : : : : : : : : :  
C61 C62 C63 C64 C65 C66 : float <
```

```
Elastic compliances
```

```
S11 S12 S13 S14 S15 S16 : float <  
: : : : : : : : : : :  
S61 S62 S63 S64 S65 S66 : float <
```

Overall elastic moduli (engineering constants)

Property tensors (for properties  $t_e$ ,  $t_c$ ,  $d_c$ ,  $e_c$ ,  $g_p$ ,  $g_d$ )

```

te11 te12 te13           : float           <
te21 te22 te23           : float           <
te31 te32 te33           : float           <
Average and dispersion of te11, te22, te33

```

## 5. 10. Result Files for Property Fields

Filename: `project.eee`, where "eee" is the extension specifying the property field and the field direction as follows:

eee	meaning
fprx	Pressure from deformation in x direction
fpry	Pressure from deformation in y direction
fprz	Pressure from deformation in z direction
fpryz	Pressure from shear in yz plane
fprxz	Pressure from shear in xz plane
fprxy	Pressure from shear in xy plane
fvmx	Von Mises stress from deformation in x direction
fvmy	Von Mises stress from deformation in y direction
fvmz	Von Mises stress from deformation in z direction
fvmyz	Von Mises stress from shear in yz plane
fvmxz	Von Mises stress from shear in xz plane
fvmxy	Von Mises stress from shear in xy plane
fprt	Pressure from thermal expansion
fvmt	Von Mises stress from thermal expansion
ftcx	Heat flux in x direction
ftcy	Heat flux in y direction
ftcz	Heat flux in z direction
fdcx	Dielectric field in x direction
fdcy	Dielectric field in y direction
fdcy	Dielectric field in z direction
fecx	Electric current in x direction

<b>eee</b>	<b>meaning</b>
f <sub>ecy</sub>	Electric current in y direction
f <sub>ecz</sub>	Electric current in z direction
f <sub>gpx</sub>	Flux in x direction
f <sub>gpy</sub>	Flux in y direction
f <sub>gpz</sub>	Flux in z direction
f <sub>gdx</sub>	Diffusivity in x direction
f <sub>gdy</sub>	Diffusivity in y direction
f <sub>gdz</sub>	Diffusivity in z direction

### 5. 10. 1. Content

For each tetrahedron element the property due to a certain deformation or field direction is stored.

#### Elastic Constants (cc)

Deformations in x, y, z direction; shear in xy (gamma), yz (alpha), xz (beta) plane (6 deformation types)

##### Pressure (fpr):

$$fpr_{x,y,z} = \sigma_0 + \sigma_1 + \sigma_2$$

stored in files `project.fprx`, `.fpry`, `.fprz`, `.fpryz`, `.fprxz`, `.fprxy`

##### Von Mises stress (fvm):

$$tau_{x,y,z} = fvm_{x,y,z} = \sqrt{\frac{1}{2} \cdot ((\sigma_0 - \sigma_1)^2 + (\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_0)^2) + \frac{3}{4} \cdot (\sigma_3^2 + \sigma_4^2 + \sigma_5^2)}$$

stored in files `project.fvmx`, `.fvmy`, `.fvmz`, `.fvmyz`, `.fvmxz`, `.fvmxy`

#### Thermal Expansion Coefficients (te)

Strain through heating the cell by 1 K.

##### Pressure (fprt):

$$fprt_{x,y,z} = \sigma_0 + \sigma_1 + \sigma_2$$

stored in file `project.fprt`

##### Von Mises stress (fvmt):

$$tau_{x,y,z} = fvmt_{x,y,z} = \sqrt{\frac{1}{2} \cdot ((\sigma_0 - \sigma_1)^2 + (\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_0)^2) + \frac{3}{4} \cdot (\sigma_3^2 + \sigma_4^2 + \sigma_5^2)}$$

stored in file `project.fvmt`

## Thermal Conductivity (*tc*)

Heat field in x, y, z direction (3 “deformation” types).

**Heat flux (*ftc*):**

$$ftc_{x,y,z} = \sqrt{\phi_0^2 + \phi_1^2 + \phi_2^2}$$

stored in files `project.ftcx`, `.ftcy`, `.ftcz`

## Electric Conductance (*ec*)

Electric field in x, y, z direction (3 “deformation” types)

**Local current (*fec*):**

$$fec_{x,y,z} = \sqrt{e_0^2 + e_1^2 + e_2^2}$$

stored in files `project.fecx`, `.fecy`, `.fecz`

## Dielectric Constants (*dc*)

Dielectric field in x, y, z direction (3 “deformation” types)

**Local electric field (*fdc*):**

$$fdc_{x,y,z} = \sqrt{e_0^2 + e_1^2 + e_2^2}$$

stored in files `project.fdcx`, `.fdcy`, `.fdcz`

## Gas Permeability (*gp*)

Permeability field in x, y, z direction (3 “deformation” types)

**Field of permeability, i. e. gas flux (*fgp*):**

$$fgp_{x,y,z} = \sqrt{e_0^2 + e_1^2 + e_2^2}$$

stored in files `project.fgpx`, `.fgpy`, `.fgpz`

## Gas Diffusivity (*gd*)

Diffusivity field in x, y, z direction (3 “deformation” types)

**Field of diffusivity, i. e. the gas flux assuming the same solubility for all phases (*fgd*):**

$$fgd_{x,y,z} = \sqrt{e_0^2 + e_1^2 + e_2^2}$$

stored in file `project.fgdx`, `.fgdy`, `.fgdz`

## 5. 10. 2. Format

```
"#" Program_version           : string           <
"#" property, field direction : string           <
NObj                          : integer          <
NVNodes                        : long              <
NTetr                          : long              <
Nsteps                         : INTEGER          <
```

<i>convergence</i>	: float	<
<i>minValue</i> <i>maxValue</i>	: float	<
<i>data</i> <sub>0</sub>	: float	<
<i>data</i> <sub>1</sub>	: float	<
:	: :	:
<i>data</i> <sub><i>NTetr</i></sub>	: float	<

## 5. 11. Backup Files

Before overriding any existing file the program will rename the previous version by appending *\_bk#* to the filename extension, where # is the version number of the backup file (number 1 stands for the most recent backup; higher numbers are older backups). Of the previous backup files the version numbers will be increased by one up to 5. The previous file with extension *\_bk5* will be deleted.

Sequence of backup file renumbering of file "name" with extension "ext":

```

name.ext_bk5 is deleted
name.ext_bk4 is renamed to name.ext_bk5
name.ext_bk3 is renamed to name.ext_bk4
name.ext_bk2 is renamed to name.ext_bk3
name.ext_bk1 is renamed to name.ext_bk2
name.ext      is renamed to name.ext_bk1

```

## 6. Batch Processing

The *GridMorph* script language for batch processing is available in *GridMorph* from Version 2.4.

Automatization with the script language allows you to automatically run many simulations of a given composite configuration.

To get statistically better results it is recommended to run the solver on several configurations of the same composite that were produced in a Monte Carlo run. Each of these configurations requires meshing (surface and volume mesh) and performing a **Solver** run.

To determine the best matrix material for a given composite it is necessary to modify the material parameters and run the Solver every time.

The *GridMorph* script language allows to automatize the essential interactive processes. It is not a general programming language that allows to write conditional statements and loops. In order to do this it is recommended to use one of the standard scripting languages available for the UNIX operating system, such as C-shell, Bourne-shell, Perl, Tcl, etc., and to embed the *GridMorph* scripts into these UNIX scripts.

### 6. 1. Batch Mode

Beside the normal interactive mode, *GridMorph* can also be run in the background (in batch mode) by executing a series of **Script commands**.

#### 6. 1. 1. Interactive mode

```
BatchMode=BMinteract
```

Run the software fully interactive, i. e. in the foreground.

Start *GridMorph* without command line options; depending on the installation on your desktop this can be done by simply clicking the GridMorph icon.

#### 6. 1. 2. Batch mode

```
BatchMode=BMbatch
```

Run the software fully in the background without visualizing anything. Comments, errors and run-time information are printed in the `log` file of the projects.

Start *GridMorph* with option `-b batchfile.bat`

If a critical situation occurs for which in interactive mode a **warning message** would appear (in a green window), *GridMorph* automatically assumes a positive answer, as if the user would press the YES button of the warning message. The warning message is printed in the `log` file of the current project.

If an error occurs, *GridMorph* writes the **error message** into the `log` file and terminates. The return value of the program is available in the UNIX shell variable `status` and can be printed immediately after termination the command `echo $status`

is performed. Upon normal termination of the program, the status value is zero. Otherwise its positive value corresponds to one of the *GridMorph* **error messages** that are listed in alphabetical order and explained above.

## 7. Script commands

The batch file contains script commands (one instruction per line) to guide a task in *GridMorph*. The commands have similar (sometimes abbreviated) names as in the pulldown menus. Each command keyword is followed by a series of parameters to guide the task.

Parameters are of type *int* (positive and negative integer numbers), *float* (positive and negative real numbers), *string* (character sequences without spaces in-between), *bool* (boolean type **Y** for yes and **T** for true or **N** for no and **F** for false). For clarity it is recommended to use either **Y** and **N** or **T** and **F** for *bool* types.

### 7. 1. Varia

#### 7. 1. 1. Comments

Introduced by a **#** as the first character of the line.

Empty lines are allowed everywhere.

Syntax:

```
# <any comment>
```

#### 7. 1. 2. Line-continuation character

Command lines with many parameters can become very long. For better readability it is possible to continue them on a new line by putting the line-continuation character `\` preceded by a space after any parameter assignment.

Example:

```
Material-Assign phase:0=Epoxy phase:1=Glass.mat \  
                phase:2=Materials/CarbonFiber
```

The space before the line-continuation character is compulsory.

Comment lines cannot be split in this way.

#### 7. 1. 3. System calls

Allows to call UNIX commands from within *GridMorph*. For example to create new directories, rename or copy files, etc.

Syntax:

```
System <UNIX command> <options>
```

Parameters:

- UNIX shell command with optional parameters. The next end-of-line marks the end of the command. Quotes are treated as if the line had been typed in the UNIX shell.

Examples:

```
System sleep 5
System mkdir NewDirectory
```

#### 7. 1. 4. Quit

Stop execution of batch file immediately (acts like a break point in the script file). This is usually useful only in visualize mode (BMviewer) if the program should be terminated after execution of the script file (e. g. for endless demonstration runs).

Syntax:

```
Quit
```

#### 7. 1. 5. Parameter assignments

Assigns a default value to parameters below. Useful in visualize (parameter) mode and for establishing convenient default settings for the dialog parameters when repeating similar operations over and over again.

Syntax:

```
Assign <command> <parm> = <value> ...
```

<command> stands for one of the following *GridMorph* script commands that are described below with associated parameters:

- Solver: prop, storeFields, step, nRep, conv

## 7. 2. Project commands

### 7. 2. 1. Project – Load

Load a project if the corresponding log, cfg and mdb (if materials are specified in the cfg file) are available

Syntax:

```
Project-Load <parm> = <value> ...
```

Parameters:

- projName (string): project name (without extension log)

### 7. 2. 2. Project – Save

Save the current project under the same name including mesh information (if available).

Syntax:

```
Project-Save
```

Parameters: none

### 7. 2. 3. Project – Save As

Save the current project under a new name. The mesh is only saved if required. The output file is saved only if the mesh file is saved.

Syntax:

```
Project-SaveAs <parm> = <value> ...
```

Parameters:

- projName (string): project name (without extension log)  
previous project files with the same name are saved as backup files

## 7. 3. Materials commands

### 7. 3. 1. Material – Assign Properties

Assign different materials to the simulation cell, a set of objects or a set of phases. The latter only in grid morphologies.

Syntax:

```
Material-Assign <parm> = <value> ...
```

Parameters:

- phase:*phaseNum* (string): material name (filename) for phases given by a set of phase numbers (between 0 and the total number of phases minus 1) in *phaseNum*

The filename for the materials file must be specified including the path if the file is not in the directory from where *GridMorph* is run. The extension *mat* is optional.

Example:

```
Material-Assign phase:0=Epoxy phase:2= Glass.mat \  
                phase:1 = materials/CarbonFiber
```

### 7. 3. 2. Material – Save Composite

Save calculated composite material in *mat* file for later use in *GridMorph*.

Syntax:

```
Material-SaveComp <parm> = <value> ...
```

Parameters:

- matName (string): material name (without extension)  
previous material files with the same name are saved as backup files [default:  
same name as current project]

## 7. 4. Solver commands

### 7. 4. 1. Solver

Run the elastic or Laplace solver. The property acronyms are

- cc: elastic constants
- te: thermal expansion coefficients (cc required)
- tc: thermal conductivity
- dc: dielectric constants
- ec: electric conductivity
- gp: gas permeability
- gd: gas diffusivity

To calculate thermal expansion coefficients (te) the elastic constants (cc) must be calculated as well. If cc is not specified in the script file together with te, cc will be calculated automatically.

Syntax:

```
Solver <parm> = <value> ...
```

Parameters:

- prop (string): comma-separated list (without spaces) of property acronyms to be calculated.  
Example: `prop = ec,tc`
- storeFields (bool): Y(es) if property fields have to be stored [default]. N(o) otherwise. Property fields use up a lot of disk space but are useful for visualization in the **View – Cut Plane** mode
- step (int): maximum number of minimization steps for all properties specified [default: 1000]
- nRep (int): reporting frequency for all properties specified [default: 50]
- conv (float): convergence for energy minimization for all properties specified [default: 0.0001]

## 8. Literature

The following list of scientific papers relate to the methods used in GridMorph and some practical applications.

- (Gusev 1997) A. A. Gusev, "Representative Volume Element Size for Elastic Composites: A Numerical Study", J. Mech. Phys. Solids 45(9), 1449-1459 (1997).
- (Goussev 2000) O. A. Goussev, P. Richner, M. G. Rozman, A. A. Gusev, "Void-containing materials with tailored Poisson's ratio", J. Appl. Phys. 88(7), 4013-4016 (2000).
- (Gusev 2000) A. A. Gusev, P. J. Hine, I. M. Ward, "Fiber packing and elastic properties of a transversely random unidirectional glass/epoxy composite", Comp. Sci. Techn. 60, 535-541 (2000).
- (Gusev 2001-1) A. A. Gusev, J. J. M. Slot, "Non-Additive Morphology Effects in Thermal Expansion of Three-Phase Materials", Adv. Eng. Mater. 3(6), 427-427 (2001).
- (Gusev 2001-2) A. A. Gusev, H. R. Lusti, "Rational Design of Nanocomposites for Barrier Applications", Adv. Mater. 13(21), 1641-1643 (2001).
- (Wissler 2002) M. Wissler, H. R. Lusti, C. Oberson, A. H. Widmann-Schupak, G. Zappini, A. A. Gusev, "Non-additive effects in the elastic behavior of dental composites", Adv. Eng. Mater. 5(3), 113-116 (2002).
- (Gusev 2004) A. A. Gusev, "Finite Element Mapping for Spring Network Representations of the Mechanics of Solids", Phys. Rev. Letters 93(3), 034302-1-4 (2004).
- (Guseva 2004) O. Guseva, H. R. Lusti, A. A. Gusev, "Matching thermal expansion of mica-polymer nanocomposites and metals", Modelling Simul. Mater. Sci. Eng 12, 101-105 (2004).