PENFO User's Guide (Version 1.0)

Michal Kočvara Michael Stingl

PENOPT GbR
www.penopt.com

March 5, 2005

Contents

1 Program description

PENFMO (original name MOPED for **M**aterial **OP**timization in **E**ngineering **D**esign; **in this manual we stick to the original name**) is a software package that computes optimal material distribution/properties in an arbitrary two- and three-dimensional domain under single and multiple load conditions. A closer description of the methods used in the code can be found in [2]. For the introduction to the theory of free material optimization, see [5] for the single-load and [1] for the multiple-load problems. The optimization problem is solved by a penalty/barrier method introduced in [3].

2 Installation

2.1 Unpacking

UNIX versions
All files needed to get a running program are packed in `moped3.tar.gz`. Put this file in an arbitrary directory. After uncompressing the file to `moped3.tar` by command `gunzip moped3.tar.gz`, the files are extracted by command `tar -xvf moped3.tar`.

Win32 version
All files needed to get a running program are packed in `moped3.zip`. Put this file in an arbitrary directory and extract the files by `PKZIP`.

The unpacking takes quite a while (be sure to provide enough disk space!) and leaves the following directory tree: the main directory is called `Moped3`; it contains three subdirectories

Doc contains this documentation;

2d contains the two-dimensional version of MOPED;

3d contains the two-dimensional version of MOPED;

each of the `2d` and `3d` directories contain the following subdirectories

Mesh contains subdirectories with project files and the interface program NAMO;

bin contains the executable programs.

2.2 Compilation

At present, MOPED runs under the following operation systems:

HP-UX 10.20;

SUN Solaris 2.6;

SuSE Linux;

Microsoft Win32.

In this manual we describe the UNIX version. When needed, we comment on the differences in the Win32 version.
The recent distribution of MOPED3 includes only executable codes and there is no need to compile anything.

3 Preparing data

Go to directory `Moped3/<dim>/Mesh`

For each project, the user creates a directory with a specific name (e.g. `ex01`) under `Moped3/<dim>/Mesh`. All input and output data for this project are collected in this directory.

The code MOPED needs three input data files

`<project>.qua` with geometry and topology of the finite element mesh

`<project>.frc` with loads

`<project>.bdr` with Dirichlet boundary conditions

To make MOPED universal, we will prepare interface programs to various popular FE codes. The recent version includes an interface to NASTRAN called NAMO3 for $dim = 3$ and NAMO for $dim = 2$ (NAstran–MOped3 interface). NAMO3 (or NAMO) reads information about the finite element mesh in standard NASTRAN format and generates the three MOPED input files.

3.1 The NASTRAN input data file

Go to directory `Moped3/<dim>/Mesh/<project>`

Input file name (`<project>.<ext>`) The name of the input file must be given with an extension, the project name is then taken from the first part of the input file name; e.g., if `example.<ext>` is the NASTRAN file then "example" is the project name. The extension (`.<ext>`) should be different from `.qua`, `.frc`, `.bdr`, because these extensions are used by MOPED. Typical extension for NASTRAN input files are `.bdf` or `.dat`. Because of the MOPED directory structure the project name must match the name of the project subdirectory in the directory Mesh: `Moped3/Mesh/<project>/<project>.<ext>`

Since not all NASTRAN features are supported by MOPED3, the following FE modelling restrictions must be considered.

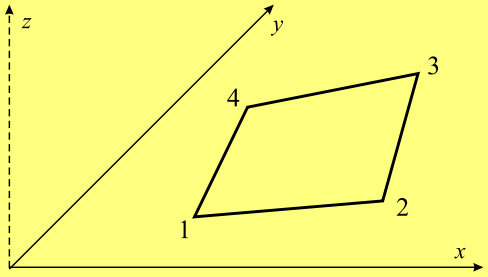
Model geometry

$dim = 2$ The model definition, i.e., the FE idealization, must be done in the 2-dimensional $x - y$ plane. All nodes are defined by `GRID` cards with x and y coordinates. The z coordinates may be defined but should be constant because the z components are not used in MOPED3. If the z coordinates are set to different values, only the projected model is considered. Local coordinate systems are not supported.

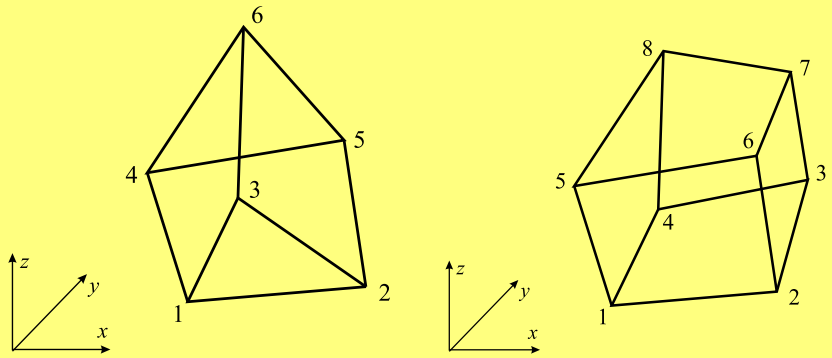
$dim = 3$ Local coordinate systems are not supported.

Elements

$dim = 2$ Elements must be defined by `CQUAD4` elements only. The mesh generation can be done by any meshing tools, e.g. IDEAS. In principle, the number of elements in MOPED is only limited by the available computer memory; in the recent version, it is limited to 20 000. The element connectivity, i.e., the sequence of the four grid numbers on the `CQUAD4` card, must be anti-clockwise with respect to the normal z axis (see Figure). In IDEAS, the so-called "first edge" displays the connectivity direction.



$\boxed{dim = 3}$ Elements must be defined by CHEXA and CPENTA elements only. The mesh generation can be done by any meshing tools, e.g. IDEAS. In principle, the number of elements in MOPED is only limited by the available computer memory; in the recent version, it is limited to 20 000. The element connectivity, i.e., the sequence of the four grid numbers on the CHEXA and CPENTA card, must be anti-clockwise with respect to the normal z axis. In IDEAS, the so-called "first edge" displays the connectivity direction.



Support The structural support may be done by SPC as well as by SPC1 cards within the Bulk data Deck. The set identification number (SPC$= n$) in the Case Control Deck may be defined but will be ignored by NAMO3 (NAMO).

Loads The load must be defined by force cards only. The number of load cases is given by the number of the SUBCASE cards in the Case Control Deck. The LOAD card in the Bulk Data Deck enables the user to make use of linear combinations of the FORCE cards: $L = S(\sum S_i L_i)$, where S is the overall scale factor, S_i is the individual scale factor and L_i is the FORCE card ID.

- If there are SUBCASE and LOAD cards, NAMO3/MOPED works in the same way as NASTRAN. Note, if there are more LOAD cards in the Bulk Data Deck then defined by SUBCASEs, they are then simply ignored.
- If there are SUBCASE cards but no LOAD cards, the forces are assigned to the load-cases by the load-identification number on the FORCE card.
- If there are no SUBCASE cards, all FORCE cards are taken into one (first) load-case, independent of their load-identification number on the FORCE card.

Other Bulk Data Deck cards Property, Material, and other cards are not needed by MOPED. All these cards may be defined to be able to run a singularity check by NASTRAN. The converters NAMO3 and NAMO ignore all input except those described above.

3.2 Generating the MOPED input files

MOPED needs the three files <project>.qua, <project>.frc, <project>.bdr which will be generated by the converter NAMO3or NAMO. Below we give the structure of <project>.frc, <project>.bdr for cases when the user would like to make changes manually. The structure is presented only for the three-dimensional case, the 2d case is analogous.

3.2.1 Forces

The forces are specified in the file <project>.frc created automatically by NAMO. *Here we show the format of this file only for cases when the user would like to make changes manually.*
The first row of this file contains information about the number of load cases (the key letter is "ell"):

l $node\_of\_load\_cases$

Forces defined at singular nodes are specified as follows:

f $load\_number$ $node\_number$ $force\_x$ $force\_y$ $force\_z$

3.3 Boundary conditions

Boundary conditions are specified in the file <project>.bdr created automatically by NAMO. *Here we show the format of this file only for cases when the user would like to make changes manually.*
Boundary conditions are given at singular nodes in the following format

b $node\_number$ $b.c.c.$

Here the boundary condition code (b.c.c.) can range from 1–7 and has the following meaning:

1 — x fixed
2 — y fixed
3 — z fixed
4 — xy fixed
5 — yz fixed
6 — xz fixed
7 — xyz fixed.

3.4 Unilateral contact conditions

Unilateral contact conditions are specified in the force file <project>.frc. In the recent verision, these conditions can only be specified "manually", so one has to edit the file <project>.frc.
Again, the conditions are specified at single nodes as follows:

o 1 $node\_number$ n_x n_y n_z

Here (n_x, n_y, n_z) is a vector pointing *from* the node *to* the obstacle. In the current version, the contact conditions are the same for all load cases (hence the fixed number 1 at the second position).
Below is a sample <project>.frc file, specifying two forces and two nodes in unilateral contact:

```
l   2
f   1       77    0.0   -10.0   0.0
f   2     3003    0.0    10.0   0.0
o   1       39    0.0    -1.0   0.0
o   1     2965    0.0     1.0   0.0
```

4 Solving a problem

4.1 Data placement

Place the NASTRAN input file <project>.<ext> discussed in the previous section in the project directory Moped3/Mesh/<project>/. The converter NAMO is located in the directory Moped3/Mesh/.

4.2 Executing NAMO3

Open a terminal window, go to directory Moped3/3d/Mesh/ and execute
namo3
without any parameters. When prompted, give the full name of the input file: <project>.<ext>. NAMO generates the three MOPED input files <project>.qua, <project>.frc, <project>.bdr, and locates these files in the project directory.
If needed (e.g. in case of unilateral contact conditions), edit manually the files <project>.frc and <project>.bdr in the project directory.
The 2d case is analogous.

4.3 Running the solver

Go to directory Moped3/<dim>/bin

There are two files with control parameters for the optimization program, in.txt and conf.txt.
File in.txt contains various parameters like stopping tolerances or output control. The meaning of the parameters is explained in the file. However, we recommend not to change these values, apart from the output level control.
File conf.txt contains just one parameter, the upper bound on the densities. The user has a choice to run the program without upper bound, with automatically computed upper bound (recommended) and with manually set upper bound. *Setting manually too low upper bound may lead to serious difficulties in the optimization code.*
There are two ways to run the solver, a fully automatic one and a "manual" one. In the manual way, one has the possibility to change the parameters in in.txt, the automatic way runs with default parameters.

Running the solver automatically

Start the solver by executing
solve <project>
Sometimes there are UNIX error messages appearing (like cp: cannot access ...) —just ignore them. Running project man1 in 2d, you should see

```
*****************************************
PENFMO 1.0
------------

 License will expire in 374 days.
*****************************************
Number of Nodes:         625
Number of Elements:      576
Number of Load Cases:      1

Scale Force: 0.100000
Upper bound: 0.040000
```

```
*************************************************************************
* it |     obj      | <U,G(x)> |  ||dF||  |   feas   |   pmin   | Nwt  | Fact |
*************************************************************************
|   0|  0.00000e+00 |  0.0e+00 |  9.8e-01 |  0.0e+00 |  1.0e+04 |    0 |    0 |
|   1| -2.09432e+06 |  1.1e+06 |  5.1e-07 |  6.4e+06 |  1.0e+04 |    7 |    7 |
|   2| -1.80996e+05 |  9.5e+04 |  1.9e-06 |  5.8e+05 |  3.0e+03 |    4 |    4 |
|   3| -1.44793e+04 |  8.3e+03 |  4.3e-04 |  5.2e+04 |  9.0e+02 |    4 |    4 |
```

If the above screen-shot does not appear there is probably something wrong with the file format of the forces or boundary conditions. In our example the computation should take few seconds. After 16 iterations, the optimization program stops. Below is the screen-shot you should see at the end of the computation:

```
|  16| -2.19862e+01 |  9.9e-09 |  4.7e-05 |  7.8e-04 |  1.4e-04 |    2 |    2 |
*************************************************************************
Objective                          -2.1986168041435679e+01
Relative Precision                  9.8532666470418917E-09
Gradient Augm. Lagrangian           4.7121926447633885E-05
Complementary Slackness             9.8532666470418917E-09
Feasibility                         7.7677998302552395E-04
Outer Iterations                                        16
Inner Iterations                                        54
Linesearch steps                                        69
Compliance                          6.1103703712018147E+00
Maximal Multiplier                  4.0000069737140091E-02
Start time                        Thu Jun 24 21:34:18 2004
End time                          Thu Jun 24 21:34:19 2004
Process time                                 0 h  0 min  1 sec
*************************************************************************
```

Now the postprocessing program starts automatically. It creates a GMV input file.

Running the solver manually

Start the data preparation phase by executing
`pre <project>`
The input data files from the directory `Moped3/<dim>/Mesh/<project>` are copied in `bin` as files `demo.qua, demo.frc` and `demo.bdr`.
At this moment the user can change parameters in `in.txt`.
Run the solver by executing
`PENFMOsl` ($dim = 2$)
or
`PENFMOsl3` ($dim = 3$)
for a single load problem or
`PENFMOml` ($dim = 2$)
or
`PENFMOml3` ($dim = 3$)
for a multiple load problem.
Finally, run the postprocessing phase by executing
`post <project>`.
Here we compute the strains, create the GMV file and copy the results in the directory `Moped3/<dim>/Mesh/<project>`.

WIN32 version:

The solver can only be run manually by a sequence
`pre` (give the project name when asked)
`PENFMO_sl3` *or* `PENFMO_ml3`
`post` (give the project name when asked)
called from the command window or by double-clicking on the corresponding icons in the `Moped3/<dim>/bin` directory.

5 Program limitations

The limitations on problem size are basically due to available memory. In the recent version, the upper bound on the number of elements is set to 20 000. However, in the multiple load case, we recommend to work with not more than 5 000–10 000 elements. The multiple load formulation is much more demanding on the CPU time, hence the first test runs for every problem should be done with, say, 300–500 elements.

References

[1] A. Ben-Tal, M. Kočvara, A. Nemirovski and J. Zowe. Free material optimization via semidefinite programming: the multiload case with contact conditions. *SIAM J. Optimization*, 9(4): 813–832, 1999 *and SIAM Review*, 42(4): 695–715, 2000.

[2] H.R.E.M. Hörnlein, M. Kočvara, and R. Werner. Material optimization: bridging the gap between conceptual and preliminary design. *Aerosp. Sci. Technol.*, 5: 541–554, 2001.

[3] M. Kočvara and M. Stingl. PENNON—a code for convex nonlinear and semidefinite programming. Preprint 290, Institute of Applied Mathematics, University of Erlangen, 2002.

[4] R. Werner. Free material optimization—Mathematical analysis and numerical solution. PhD thesis, Inst. of Appl. Math., University of Erlangen, 2001.

[5] J. Zowe, M. Kočvara, and M. Bendsøe. Free material optimization via mathematical programming, *Mathematical Programming*, 79:445-466, 1997.