

Parallelepiped whose Young's modulus is a function of the temperature Fino benchmark problem series

Number	Rev.		
SP-FI-17-BM-12F2	А		
Date	Hash	Author	
12-Jun-2017	44695ea	Jeremy Theler	jeremy@seamplex.com
Document type	Pages		
Benchmark problem	13		

Abstract

A benchmark thermal-mechanical problem that has an analytical solution was solved using fino. Even though the original problem is defined over a simple parallelepiped, this reports uses a fully unstructured grid to validate the code using arbitrary jacobians and not just structured hexahedra. The way fino handles dependence of parameters on space, in this case through intermediate distributions such as the temperature, is illustrated by the simplicity of the input files used to solve both the thermal and the mechanical problems. The comparison between the numerical and the analytical solutions are within the order of 10^{-6} even for coarse grids.



Keywords Fino, benchmark, CodeAster, thermal conduction, linear elasticity

Revision history

Rev.	Date	Author	
А	12-Jun-2017	gtheler	First issue

Contents

1	Introduction	4
•	1.1 About fino	4
2	Problem	4
3 Solution		
	3.1 Analytical solution	6
	3.2 Geometry discretization	6
	3.3 Thermal problem	8
	3.4 Mechanical problem	8
	3.5 Coarser and finer grids	11
4	Conclusions	13

1 Introduction

This reports shows the results of solving the benchmark problem stated in reference [1] that consists of a thermomechanical problem in which a parallelepiped whose Young's modulus depends on the temperature is subject to known heat fluxes on its surface and to an external traction pressure at the same time. The geometry, the material properties and the boundary conditions are selected such that the problem has analytical solutions for both the temperature and displacement distributions, as computed by solving the heat diffusion and the linear elasticity equations respectively.

The original problem (and even other solutions such as [2]) uses an structured grid to exploit the simplicity of the problem domain. However interesting this approach is, it fails to provide a satisfactory validation case because it may be the case that the computer code handles correctly structured hexahedra but fails to obtain good results for arbitrary geometries that need unstructured grids. So in order to partially remedy this issue, in this report we solve the simple geometry that gives rise to analytical solutions but using a not-so-fine unstructured grid to discretize the parallelepiped. This way, the results involve the evaluation of non-trivial jacobians and numerical integrations and if we do get the results, the conclusions about the validation of the code for more general cases is more satisfactory than if we employed just simple structured grids.

1.1 About fino

The problem is solved with the free ("Free" both as in "free speech" and in "free beer.") and open source finite-element analysis tool fino—developed by Seamplex—with a Gmsh-generated unstructured grid. More information about the programs, including documentation and downloads can be found at



(a) https://www.seamplex.com/fino



(b) http://gmsh.info/

It should be noted that, following Seamplex' principles,¹ only free and open source software was used in preparing this report.

2 Problem

The problem consists of finding the non-dimensional temperature T and displacements u, v and w distributions within a solid parallelepiped of length l whose base is a square of $h \times h$ and that is subject to heat fluxes and to a traction pressure at the same time. The geometry is shown in figure 1. The non-dimensional Young's modulus E of the material depends on the temperature T (which in turns depends on space) in a know algebraically way, whilst both the Poisson coefficient ν and the thermal conductivity k are uniform and do not depend on the spatial coordinates:

$$E(x, y, z) = \frac{1000}{800 - T(x, y, z)}$$
$$\nu = 0.3$$
$$k = 1$$

The thermal boundary conditions are:

¹https://www.seamplex.com/principles.html



Figure 1: The problem to be solved. Original figure from reference [1].

- Temperature at point A at (l, 0, 0) is zero
- Heat flux q'' through x = l is +2
- Heat flux q'' through x = 0 is -2
- Heat flux q'' through y = h/2 is +3
- Heat flux q'' through y = -h/2 is -3
- Heat flux q'' through z = h/2 is +4
- Heat flux q'' through z = -h/2 is -4

The mechanical boundary conditions are:

- Point O at (0, 0, 0) is fixed
- Point B at (0, h/2, 0) is restricted to move only in the *y* direction
- Point C at (0, 0, /h2) cannot move in the x direction
- Surfaces x = 0 and x = l are subject to an uniform normal traction equal to one

- $\begin{cases} T(l,0,0) = 0\\ -k \cdot \frac{\partial T}{\partial n} \Big|_{x=l} = +2\\ -k \cdot \frac{\partial T}{\partial n} \Big|_{x=0} = -2\\ -k \cdot \frac{\partial T}{\partial n} \Big|_{y=h/2} = +3\\ -k \cdot \frac{\partial T}{\partial n} \Big|_{y=-h/2} = -3\\ -k \cdot \frac{\partial T}{\partial n} \Big|_{z=h/2} = +4\\ -k \cdot \frac{\partial T}{\partial n} \Big|_{z=-h/2} = -4 \end{cases}$
 - $\begin{cases} u(0,0,0) = 0\\ v(0,0,0) = 0\\ w(0,0,0) = 0\\ u(0,h/2,0) = 0\\ w(0,h/2,0) = 0\\ u(0,0,h/2) = 0\\ p|_{x=0} = 1\\ p|_{x=l} = 1 \end{cases}$

3 Solution

The case is a thermo-mechanical problem in which the mechanical properties (namely the Young's modulus E) depends on the temperature T. But the thermal properties do not depend back on the mechanical properties. Under the assumption of linearity in the geometry (i.e. that the displacements due to the mechanical loads are negligible) the problem is not coupled. It can then be solved by first solving the standalone thermal problem and then using the resulting temperature distribution T to evaluate E. It should be noted that under these conditions, the problem is also linear in the material properties because E depends explicitly on x, y and z and does not depend on the unknowns u, v or w.

3.1 Analytical solution

The problem stated in section 2 has the following analytical solutions for the temperature and displacement distributions under the heat diffusion equation and the linear elasticity formulation:

$$\begin{split} T(x,y,z) &= -2x - 3y - 4z + 40\\ u(x,y,z) &= \frac{A}{2} \cdot \left[x^2 + \nu \cdot \left(y^2 + z^2 \right) \right] + B \cdot xy + C \cdot xz + D \cdot x - \nu \cdot \frac{Ah}{4} \cdot \left(y + z \right)\\ v(x,y,z) &= -\nu \cdot \left[A \cdot xy + \frac{B}{2} \cdot \left(y^2 - z^2 + \frac{x^2}{\nu} \right) + C \cdot yz + D \cdot y - A \cdot h/4 \cdot x - C \cdot h/4 \cdot z \right]\\ w(x,y,z) &= -\nu \cdot \left[A \cdot xz + B \cdot yz + C/2 \cdot \left(z^2 - y^2 + \frac{x^2}{\nu} \right) + D \cdot z + \frac{Ch}{4} \cdot y - \frac{Ah}{4} \cdot x \right] \end{split}$$

where A = 0.002, B = 0.003, C = 0.004 and D = 0.76. The reference results are the temperature at points O and D and the displacements at points A and D (table 1).

Point	Unknown	Reference value
0	T	40.0
D	T	-35.0
А	u	15.6
	v	-0.57
	w	-0.77
D	u	16.3
	v	-1.785
	w	-2.0075

Table 1: Reference results the original benchmark problem [1] asks for.

This problem is purely of mathematical interest because the assumption of the geometrically linearity implied in the linear elastic problem formulation clearly does not hold as the parallelepiped elongates 75% of its original length in the x direction. Moreover, this expansion would change the thermal problem conditions as well, rendering the problem mechanically-thermal coupled.

3.2 Geometry discretization

The geometry is fairly simple, although the points A, B, C, D and O need to be explicitly defined and embedded into the grid so boundary conditions can be attached to them. The following Gmsh input script creates a suitable second-order grid for both the thermal and the mechanical problem (figure 2):

```
SetFactory("OpenCASCADE");
h = 10; // base length
l = 20; // height
lc = l/20; // characteristic element size
Mesh.ElementOrder = 2;
Point(1) = {0, h/2, h/2, lc};
Point(2) = {l, h/2, h/2, lc};
Point(3) = {0, h/2, -h/2, lc};
Point(5) = {0, -h/2, h/2, lc};
Point(6) = {l, -h/2, h/2, lc};
Point(6) = {0, -h/2, h/2, lc};
Point(12) = {0, 0, 0, lc};
Point(12) = {0, 0, 0, lc};
Point(13) = {1, 0, 0, lc};
Point(15) = {0, 0, h/2, lc};
Line(1) = {5, 15};
Line(2) = {15, 1};
Line(3) = {1, 14};
Line(4) = {14, 4};
Line(5) = {4, 8};
```

```
Line(6) = {8, 5};

Line(10) = {5, 6};

Line(11) = {2, 3};

Line(12) = {3, 7};

Line(13) = {7, 6};

Line(15) = {4, 3};

Line(16) = {1, 2};

Line Loop(1) = {2, 3, 4, 5, 6, 1};

Plane Surface(1) = {1};

Line Loop(2) = {9, 10, -16, -2, -1};

Plane Surface(2) = {2};

Line Loop(3) = {-13, -10, -11, -12};

Plane Surface(3) = {3};

Line Loop(4) = {15, 12, -14, -5};

Plane Surface(4) = {4};

Line Loop(5) = {11, -15, -4, -3, 16};

Plane Surface(6) = {6};

Point {12} In Surface {1};

Point {13} In Surface {3};

Surface Loop(1) = {6, 4, 5, 3, 2, 1};

Volume(1) = {1};

// pretty names

Physical Point("0") = {12};

Physical Point("0") = {1};

Physical Surface("x=0") = {3};

Physical Surface("x=0") = {5};

Physical Surface("x=0") = {3};

Physical Surface("x=0") = {5};

Physical Surface("x=0") = {5};

Physical Surface("x=0") = {6};

Physical Surface("x=0") = {6};

Physical Surface("x=0") = {1};

Physical Surface("x=0") = {1};

Physical Surface("x=0") = {1};

Physical Surface("x=0") = {3};

Physical Surface("x=0") = {3};

Physical Surface("x=0") = {3};

Physical Surface("x=0") = {3};

Physical Surface("x=0") = {1};

P
```







🕭 SEAMPLEX

3.3 Thermal problem

The first step is to solve the thermal problem. With the grid from the last section, this relatively simple input²

- 1. reads the mesh (line 1),
- 2. selects the problem to be "thermal" (default is "mechanical" line 2),
- 3. defines the material properties, only the thermal conductivity k in this case (line 5)
- 4. defines the boundary conditions (lines 8-16),
- 5. solves the problem (line 18),
- 6. defines the analytical solution T(x, y, z) (line 20),
- 7. writes a VTK file with the result and the absolute error with respect to the analytical solution (line 22),
- 8. writes a MSH file with the computed temperature distribution (which can be read back by fino in the mechanical problem, line 23), and
- 9. prints to the standard output the results the original benchmark problem asks for [1] (lines 25-26).

1	MESH FILE_PATH parallelepiped.msh DIMENSIONS 3
2	FINO_PROBLEM BAKE # the problem type has to come after the mesh is loaded (default is BREAK = mechanical)
3 4 5	<pre># thermal conductivity (the problem is non-dimensional) k = 1</pre>
7	# houndary conditions
8	PHYSICAL ENTITY NAME A BC T=0 # temperature at point A
9	
10	# we give $q = -k dT/dn$ here, in the original reference the give +k Dt/dn, hence we have opposite signs
11	PHYSICAL_ENTITY NAME "x=l" BC q=+2
12	PHYSICAL_ENTITY NAME "x=0" BC q=-2
13	PHYSICAL_ENTITY NAME "y=h/2" BC q=+3
14	PHYSICAL_ENTITY NAME "y=-h/2" BC q=-3
15	PHYSICAL_ENTITY NAME "z=h/2" BC q=+4
16	PHYSICAL_ENTITY NAME "z=-h/2" BC q=-4
1/	
10	FIND_SIEK # SOLVE:
20	Treal(x, y, z) + analytical solution
21	reac(A))(2) = 2 × 5 y + 2.40
22	MESH POST FILE PATH thermal.vtk T T(x.v.z)-Treal(x.v.z)
23	MESH POST FILE PATH temperature.msh T
24	
25	PRINT "T(0),=," %.6f T(0,0,0) TEXT "abs,error,=" %.2e T(0,0,0)-Treal(0,0,0)
26	PRINT "T(D)_=_" %.6f T(20,5,5) TEXT "abs_error_=" %.2e T(20,5,5)-Treal(20,5,5)

Listing 2: thermal.fin

Running fino v0.5.63-gff06eb9 with thermal.fin as the main input file gives the desired results:

```
$ fino thermal.fin

T(0) = 39.999998 abs error = -1.54e-06

T(D) = -34.999992 abs error = 7.76e-06

$
```

Using the VTK output, some pretty figures illustrating the thermal results can be built as in figure 3. The errors with respect to the analytical solution of the temperature distribution are given as signed absolute values because giving relative errors as a measure of the degree of accuracy of the code is conceptually wrong because the temperature scale is defined up to an additive constant. Thus, as the zero of the scale can be defined by the value of the only Dirichlet boundary condition the problem has (first bullet in page 5), relative errors are ill-defined.

3.4 Mechanical problem

Once the temperature distribution is available, the mechanical problem can be solved. In this case, the Young's modulus E depends on temperature. The benchmark problem asks for the numerical temperature computed in the last section, although we may use the actual analytical solution also. The latter can be easily defined in fino³ it has already been done in the thermal problem as

²Fino's (actually wasora's [3]) motto is "Simple problems ought to have simple inputs."

³Actually this line is handled by the wasora framework (https://www.seamplex.com/wasora on top of which fino is developed [3].



Figure 3: Resulting temperature distribution as solved by fino using thermal.fin and comparisons with respect to the analytical solution.

T(x,y,z) := -2*x-3*y-4*z+40

1

Listing 3: real.fin

The former can be read from the numerical output written by thermal.fin in a Gmsh file, altogether with the mesh topology:

MESH NAME temp FILE_PATH temperature.msh DIMENSIONS 3 READ_DATA T AS T

Listing 4: computed.fin

In this case, both the thermal and the mechanical problems are solved using the same second-order grid and as such, the temperature T and the Young's modulus E will have the same points of definition and interpolation functions so the performance will be better. However, fino can handle the case where the numerical T that E depends on is defined in a different grid, for example with a different number of nodes or with elements with a different order.

The mechanical problem can the be solved with the following fino input, which needs an extra command line argument to use either the analytical temperature distribution (real) or the numerical one (computed), and that

- 1. includes either real.fin or computed.fin to define T(x, y, z) (line 5),
- 2. sets E as an algebraically-defined function of space⁴ (through T(x, y, z)) and ν as a constant (lines 6–7),
- 3. defined the boundary conditions (lines 10–14),
- 4. solves the problem (line 16),
- 5. defines the analytical solutions (lines 19-27),
- 6. writes a VTK file with the three absolute errors and a vector with the computed displacements (line 29),
- 7. prints to the standard output the results the original benchmark problem asks for [1] (lines 31–37).

1 2 3	MESH FILE_PATH parallelepiped.msh DIMENSIONS 3 FINO_PROBLEM BREAK
4	# material problems
5	INCLUDE \$1.fin # either real or computed $T(x, y, z)$
6	E(x,y,z) := 1000/(800-T(x,y,z))
7	nu = 0.3
8	
9	# boundary conditions
10	PHYSICAL ENTITY NAME 0 BC TIXED
12	
13	PHYSICAL_INTIT NAME "x=1" BC n=1
14	PHYSICAL ENTITY NAME "x=0" BC p=1
15	
16	FINO_STEP
17	
18	# analytical solutions
19	n = 10
20	A = 0.002
22	B = 0.003
23	D = 0.76
24	
25	ureal $(x,y,z) := A/2*(x^2 + nu*(y^2+z^2)) + B*x*y + C*x*z + D*x - nu*A*h/4*(y+z)$
26	vreal(x,y,z) := -nu*(A*x*y + B/2*(y^2-z^2+x^2/nu) + C*y*z + D*y -A*h/4*x - C*h/4*z)
27	wreal (x,y,z) := -nu* $(A^*x^*z + B^*y^*z + C/2^*(z^2-y^2+x^2/nu) + D^*z + C^*h/4^*y - A^*h/4^*x)$
28	
29	MESH_POST FILE_PATH mechanical-\$1.vtk u(x,y,z)-ureal(x,y,z) v(x,y,z)-vreal(x,y,z) w(x,y,z)-wreal(x,y,z) VECTOR u v w
31	PRINT "U(A) = " % 6f U(20 0 0) TEXT "abs error =" % 2e U(20 0 0)-ureal(20 0 0)
32	FRIN $[v(A)_{-}]$ so $(2^{-})(2^{-$
33	FRINT "(v(A), =, "%.6f w(20,0,0) TEXT "abs.error.="%.2e w(20,0,0)-wreal(20,0,0)
34	
35	PRINT "u(D)_=_" %.6f u(20,5,5) TEXT "abs_error_=" %.2e u(20,5,5)-ureal(20,5,5)
36	PRINT "v(D)_=_" %.6f v(20,5,5) TEXT "abs_error_=" %.2e v(20,5,5)-vreal(20,5,5)
37	PRINT "w(D)_=_" %.6f w(20,5,5) TEXT "abs_error_=" %.2e w(20,5,5)-wreal(20,5,5)

Listing 5: mechanical.fin

[&]quot;The original reference [1] says "Il est nécessaire de prévoir un grand nombre de points de discrétisation de la courbe E(T) pour obtenir la précision souhaitée. Ici on a pris 250 points (E_i, T_i) ." With fino, line 6 is all that it takes. No need to worry about the definition points, it takes care by itself to sample the function wherever it is needed.

In this case, fino ought to be run with mechanical.fin as the main input file and an extra argument, either real or computed:

e-06				
e-06				
e-07				
le-05				
e-06				
e-06				
<pre>\$ fino mechanical.fin real</pre>				

u(A) =	15.600002	abs error =	2.43e-06
v(A) =	-0.569996	abs error =	4.50e-06
w(A) =	-0.769999	abs error =	1.02e-06
u(D) =	16.299989	abs error =	-1.13e-05
v(D) =	-1.784994	abs error =	5.60e-06
w(D) =	-2.007494	abs error =	5.93e-06
\$			

There is no appreciable difference between both cases. And again, pretty pictures like figure 4 can be built out of the post-processing files generated by Fino.

3.5 Coarser and finer grids

A quick grid sensitivity study can be performed from the commandline:⁵

```
$ gmsh -3 -clscale 1.5 parallelepiped.geo | grep vertices | grep elements
Info
      : 6160 vertices 5069 elements
$ fino thermal.fin
T(0) = 40.000000
                      abs error =
                                      1.35e-07
T(D) = -34.999991
                      abs error =
                                      9.45e-06
$ fino mechanical.fin computed
u(A) = 15.600002 abs error =
                                     2.32e-06
v(A) = -0.570009
w(A) = -0.770003
                      abs error =
                                      -9.09e-06
                      abs error =
                                      -2.82e-06
u(D) = 16.300015
                     abs error =
                                      1.51e-05
v(D) = -1.785009
w(D) = -2.007507
                      abs error =
                                      -8.63e-06
                      abs error =
                                      -6.73e-06
$ fino mechanical.fin real
u(A) = 15.600001 abs error =
                                      8.44e-07
v(A) = -0.570009
                      abs error =
                                      -8.69e-06
w(A) = -0.769999 abs error =
                                      1.26e-06
                    abs error =
u(D) = 16.300012
                                      1.16e-05
v(D) = -1.785010
                      abs error =
                                      -1.00e-05
w(D) = -2.007503
                      abs error =
                                      -2.51e-06
$ gmsh -3 -clscale 0.85 parallelepiped.geo | grep vertices | grep elements
Info : 23823 vertices 19030 elements
$ fino thermal.fin
T(0) = 39.999991
                      abs error =
                                      -9.34e-06
T(D) = -34.999982
                                      1.84e-05
                      abs error =
$ fino mechanical.fin computed
u(A) = 15.599999 abs error =
                                      -6.93e-07
v(A) = -0.570007
w(A) = -0.770004
                                      -7.50e-06
                      abs error =
                      abs error =
                                      -4.18e-06
u(D) = 16.299998
                     abs error =
                                      -1.92e-06
v(D) = -1.785012
                      abs error =
                                      -1.16e-05
w(D) = -2.007500
                      abs error =
                                      3.36e-07
$ fino mechanical.fin real
                                      -7.17e-07
u(A) = 15.599999
                      abs error =
v(A) = -0.570008
                      abs error =
                                      -7.62e-06
w(A) = -0.770004
                      abs error =
                                      -4.03e-06
u(D) = 16.299998
                      abs error =
                                      -1.92e-06
v(D) = -1.785012
                      abs error =
                                      -1.18e-05
w(D) = -2.007500
                      abs error =
                                      4.80e-07
$
```

⁵To see how to perform full parametric studies using fino's (actually, wasora's) functionality see reference [4].



(a) Original and strained geometry. The color range shows the modulus of the displacement vector |u(x, y, z), v(x, y, z), w(x, y, z)|.



(b) Absolute error in $\boldsymbol{u}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$



(c) Absolute error in $\boldsymbol{u}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$



(d) Absolute error in $\boldsymbol{u}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$



(e) Absolute error in $\boldsymbol{v}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$

(h) Absolute error in w(x, y, z)



(f) Absolute error in $\boldsymbol{v}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$



(i) Absolute error in w(x, y, z)



(g) Absolute error in v(x, y, z)





Figure 4: Results of the mechanical problem with the numerical temperature distribution computed in section 3.3

4 Conclusions

The benchmark problem from reference [1] was solved using the free and open source solver fino. An unstructured grid of approximate 15k nodes was generated using Gmsh in order to validate the usage of general tetrahedra with arbitrary jacobians and non-trivial transformations from real to canonical coordinate systems. Under the assumption of linear geometry (i.e. that the mechanical displacements do not distort the grid) and linear elasticity then the problem is not coupled in the mechanical-thermal direction and has analytical solutions for the temperature distribution T and the displacements u, v and w. The numerical solution obtained with fino, first by solving just the thermal problem and then by plugging the obtained temperature distribution into a mechanical problem where the Young's modulus E depended on T(x, y, z) had absolute errors of the order of 10^{-6} with respect to the analytical solutions. The same second-order grid was used for both problems, although similar results would have been obtained for the thermal problem if just first-order tetrahedra were used.

A quick grid sensitivity analysis shows that errors in the thermal problem are insensitive to the number of elements while in the mechanical problem errors slightly increase (decrease) with coarser (finer) grids. This is expected, as the thermal solution is purely linear so the errors do not come from spatial discretization issues. The mechanical problem is slightly more complex, but second-order elements are able to retain the physics and satisfactorily reproduce the analytical results.

It should be noted that this non-dimensional problem has a purely mathematical interest, and can be used to show that the same problem solved analytically on the one hand and numerical on the other hand give equivalent results. Clearly, the assumption that the geometry is linear does not hold as the domain expands almost at twice of its original length in the x direction. This fact renders the problem both non-linear and fully coupled, because the mechanical deformations change the areas of the surfaces that are subject to heat fluxes, not to mention that a material strained that way will for sure stop obeying Hooke's law. Nevertheless, the linear problem is a nice test case for computational finite-element codes and the results encourage us to keep on working on fino and its web-based front-end CAEplex.



Figure 5: Fino and Gmsh running as a back-ends on the cloud for the web-based front-end CAEplex at caeplex.com

References

- [1] Jacques Pellet. Parallélépipède dont le module d'Young est fonction de la température. Thermo-mécanique stationnaire linéaire des systèmes volumiques. French. Tech. rep. HPLV100. 2011.
- [2] Simscale GmbH. Parallelepiped whose Young modulus is function of the temperature. https://www.simscale.com/docs/content/validation. 2017.
- [3] Jeremy Theler. *Description of the computational tool wasora*. Technical Description SP-WA-15-TD-9E3D. Version B. Seamplex, 2016.
- [4] Jeremy Theler. *Thermo-elastic expansion of finite cylinders*. Benchmark problem SP-FI-17-BM-5460. Version A. Seamplex, Apr. 12, 2017.