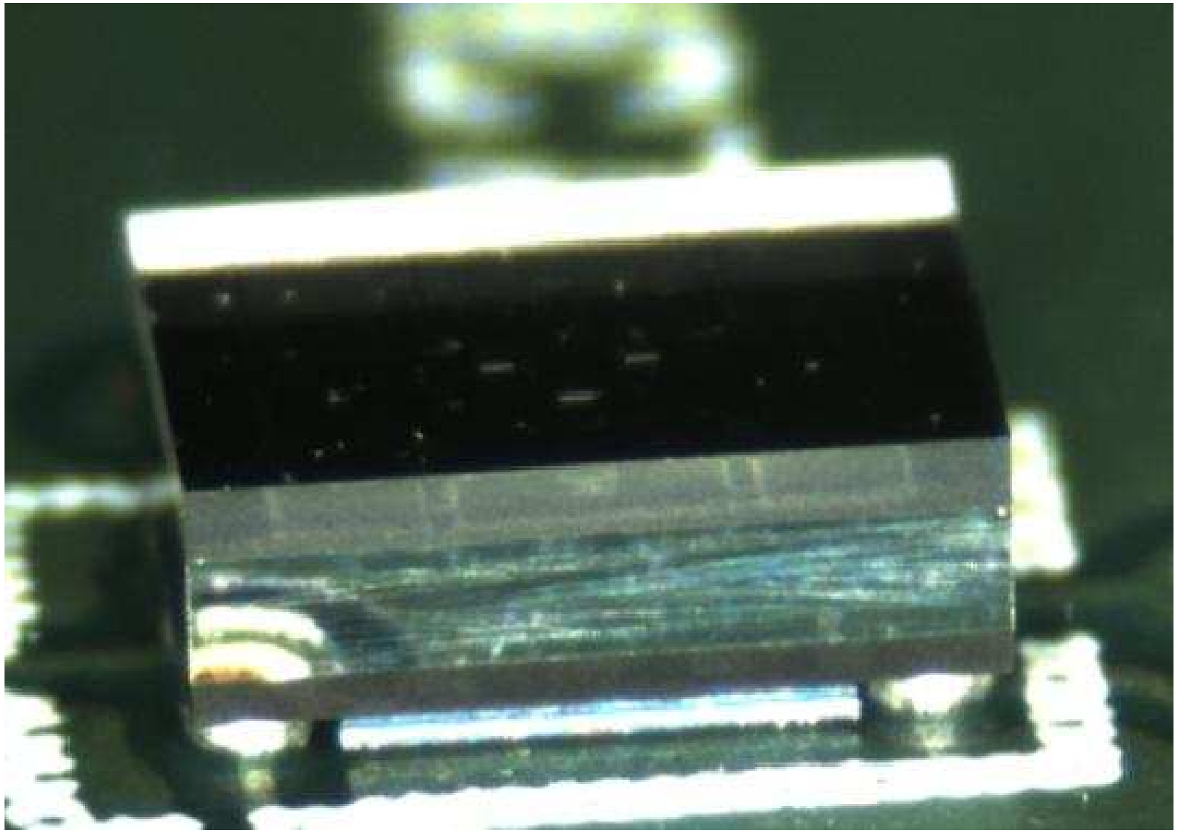
Presented at the COMSOL Conference 2009 Milan



Multi-Objective Optimization of a Ball Grid Array using modeFRONTIER & COMSOL Multiphysics

2009-10-16

Håkan Strandberg, ESTECO Nordic AB, Lund, Sweden
Tapani Makkonen & Joni Leinvuo, VTI Technologies, Vantaa, Finland

VTI Technologies Oy, Vantaa, Finland

- Accelerometers (1-, 2- and 3-axis), gyroscopes and pressure sensors
- Automotive, Instrument, Medical and Consumer applications
- Common automotive application: ESC – electronic stability control
- 3D MEMS technology:
  - accuracy in low-g ranges
  - low power consumption
  - small size
  - competitive price-to-quality ratio
- Net sales ~70 MEUR/year

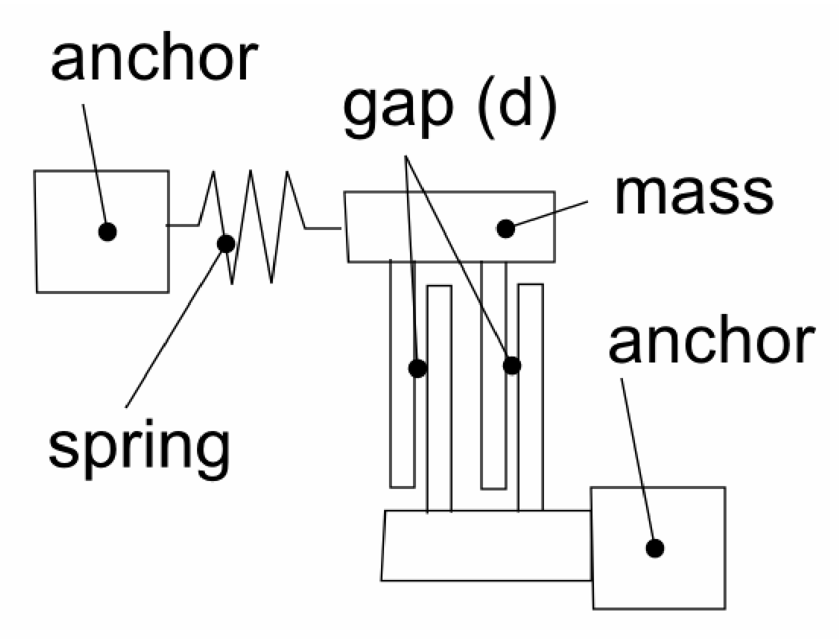
3-axis digital accelerometer (automotive)

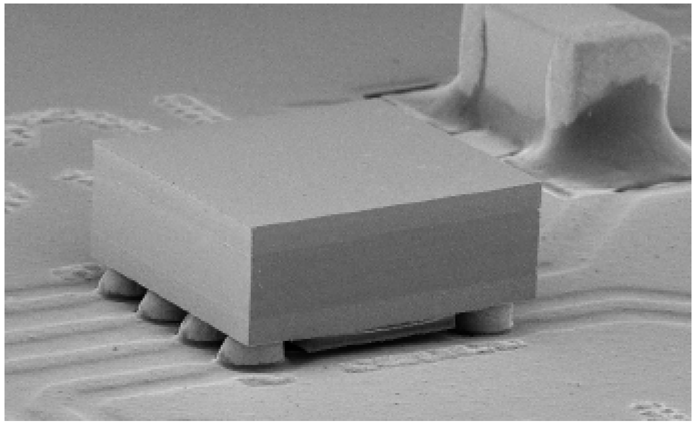
www.vti.fi

Absolute pressure sensor (altimeter applications)

How does a low-g accelerometer work?

- Silicon capacitive accelerometer
- Distance (d) between two surfaces changes under acceleration
- Capacitance C=εA/d, A is surface area, d is distance between the two surfaces
- Made from single-crystal silicon and glass, hermetically sealed

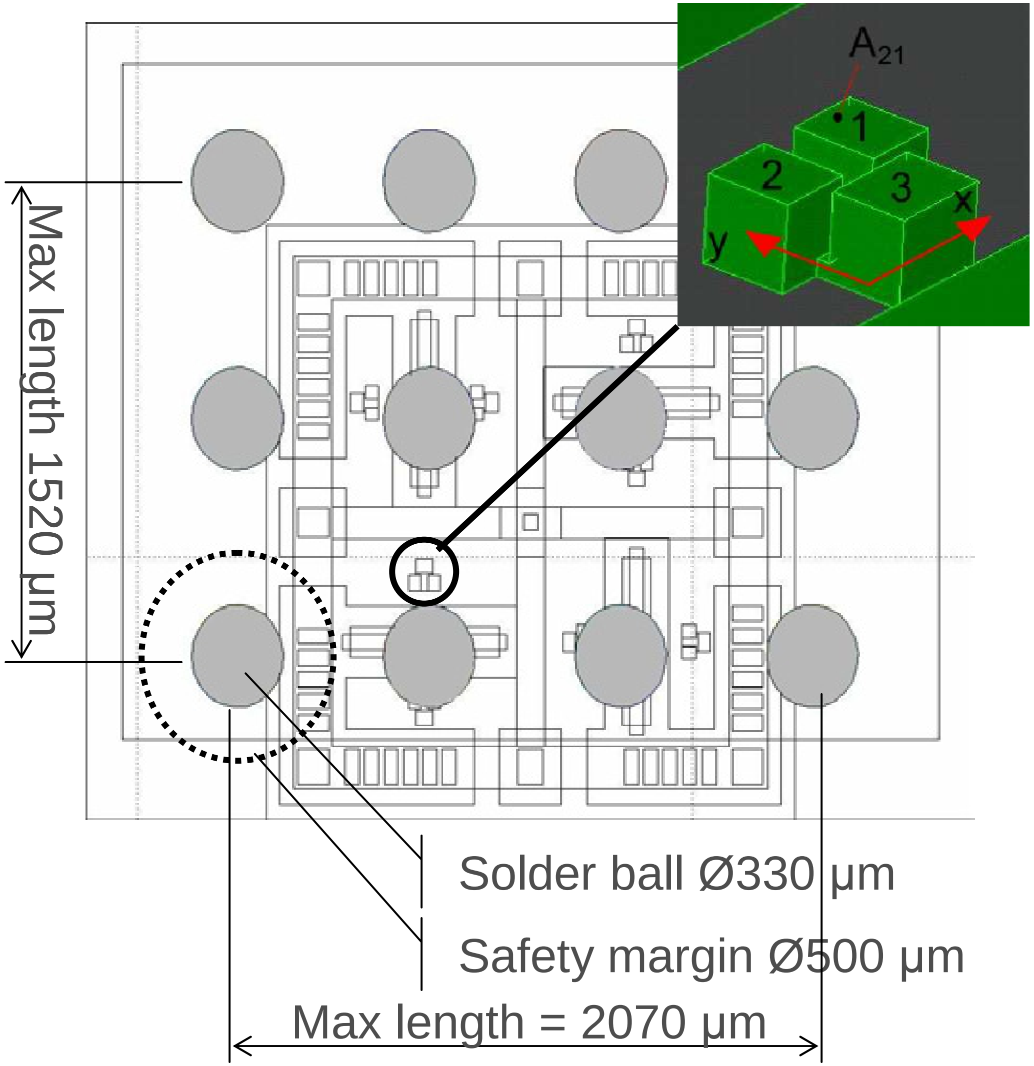




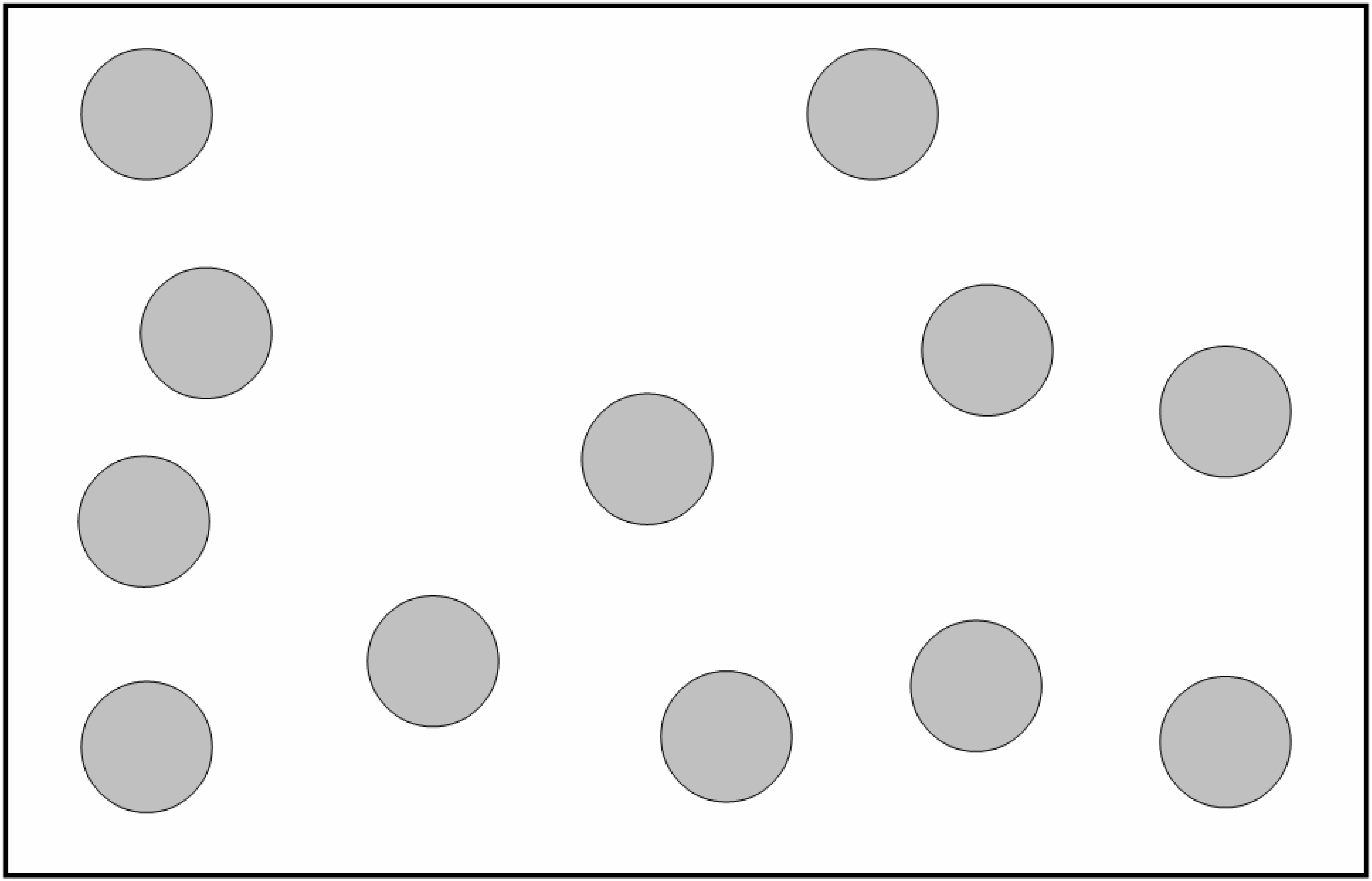
CMA3000 3-axis accelerometer for consumer applications flip-chipped on PCB (2mm x 2mm x 0.95mm)

Optimization of the BGA layout

- Minimize the effect of temperature changes by finding the best location of the 12 solder balls
- 24 input parameters
  - x & y of each solder ball
- Two conflicting objectives
  - Minimize the signal offset error, i.e. minimize the movement of the 24 anchors
  - Minimize peak stress in the solder balls to maximize service life
- Manufacturing constraint, minimum distance between solder balls = 500 µm

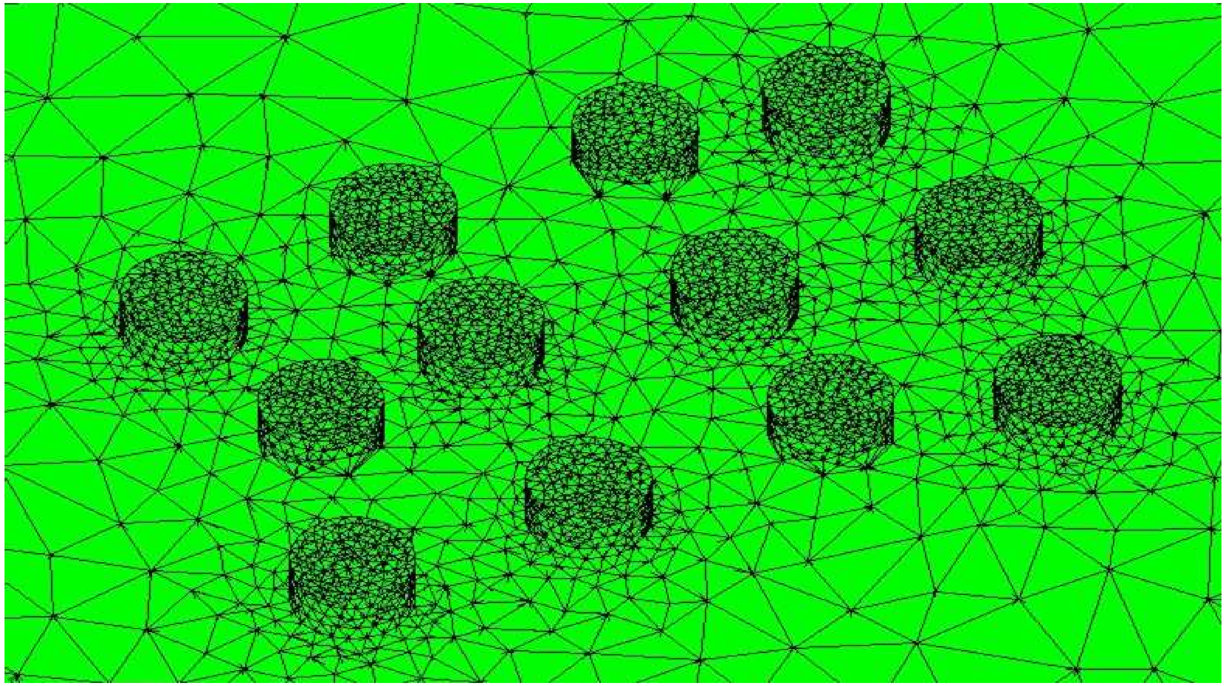


This design may be of interest



The Comsol analysis model

- Linear model, plasticity and creep of the solder is omitted
- Temperature changes from +85 C to -40 C
- ~400000 tetrahedral elements (10 noded), ~1.65 MDOF



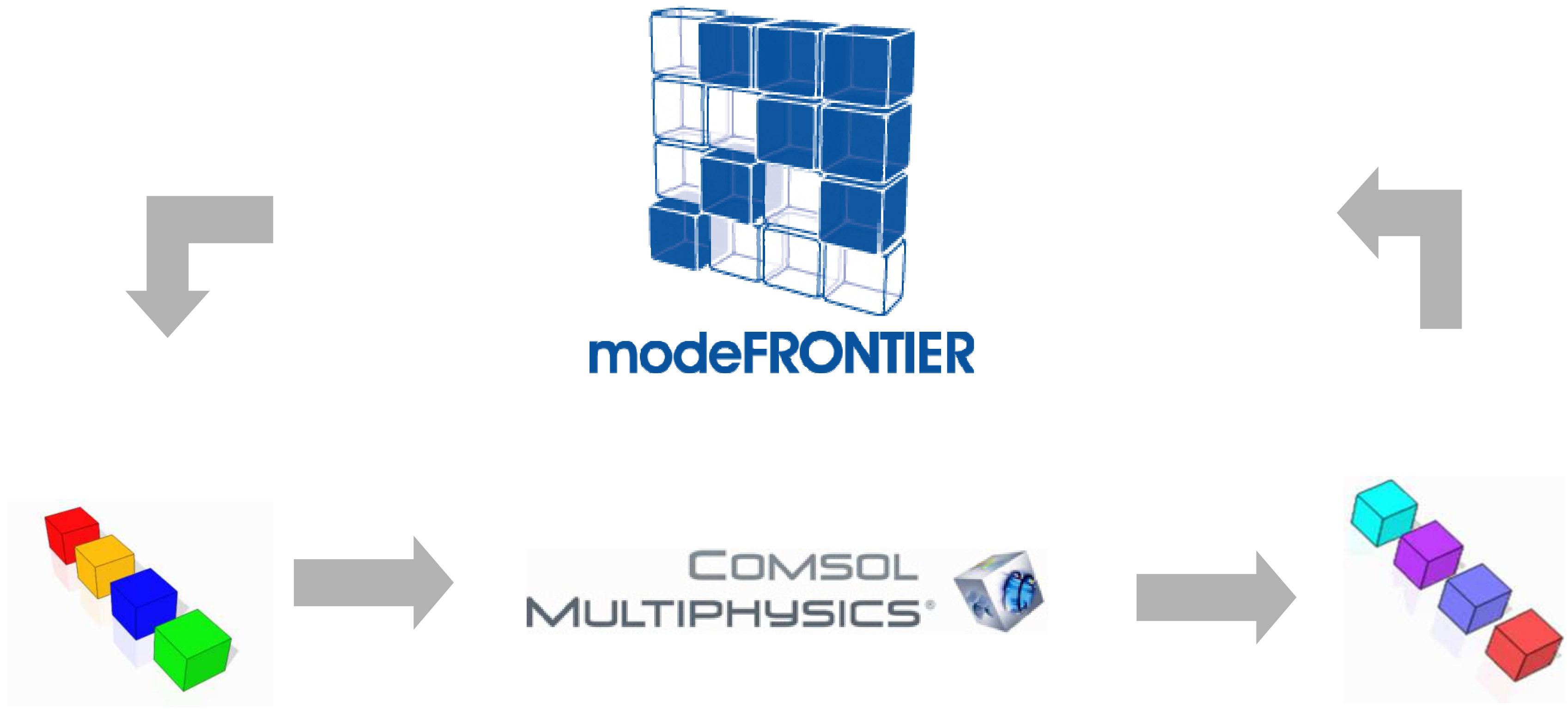
Mesh controls are set on lines and surfaces to ensure a dense mesh in critical parts of the model and to achieve consistent mesh between different geometries

modeFRONTIER

modeFRONTIER is an "artificial intelligence" software which fully exploits the available software and hardware resources to:

- automate the design process
- explore the design space
- independently search for optimal solutions of a multi-objective and multi-disciplinary problem

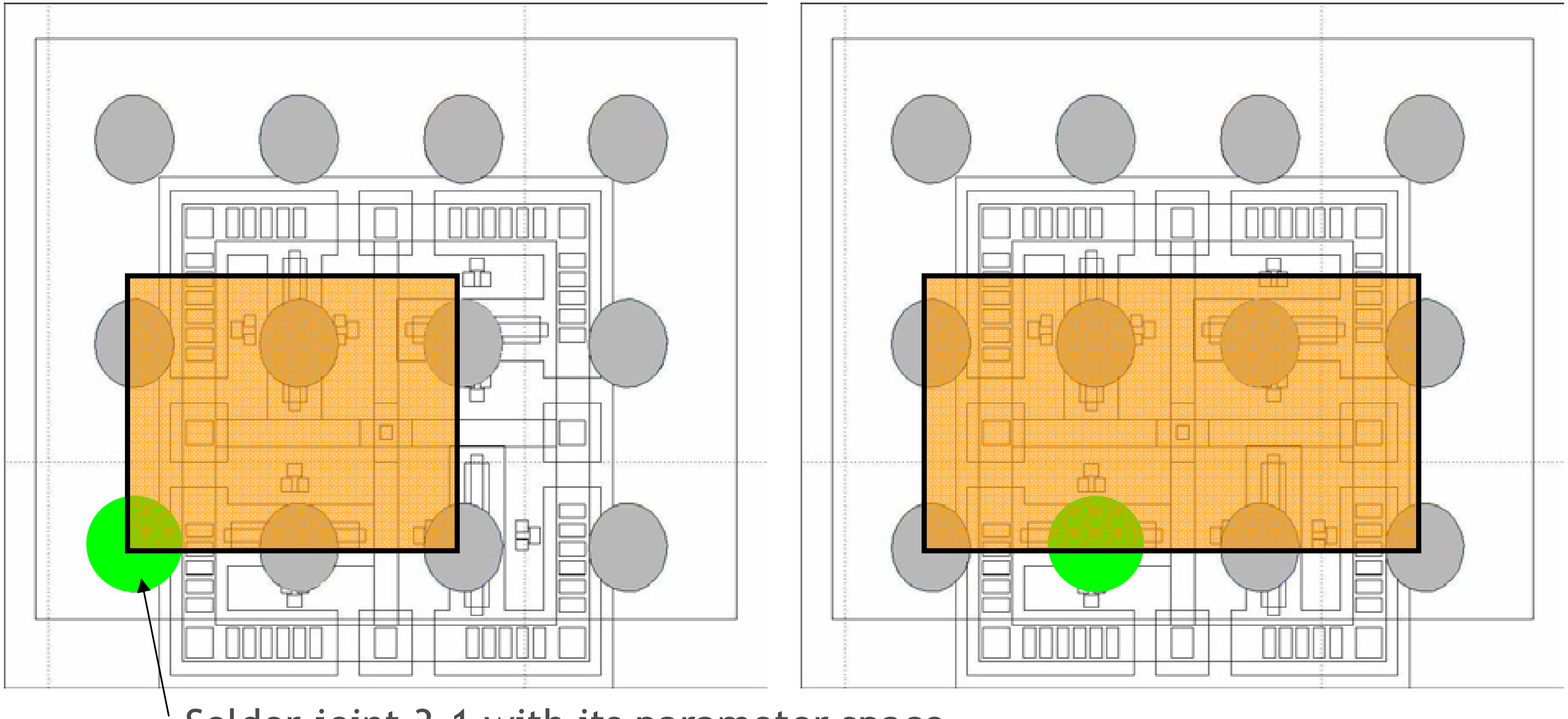
Process integration



Design Variables:
Entities defining the design space

Output Variables:
Measurements from the system

Parameterization, range & step



Solder joint 3,1 with its parameter space

- Allowing very different BGA layouts requires large parameter ranges
- Easy-to-implement parameterization leads to many collisions and a hard optimization problem

Parameterization, Process integration

```
% Constants
C2x=515e-6;
C2y=265e-6;

g2=move(g2,[C2x,C2y,0]);
...
```

```
% Constants
C2x=<VAR name="x9" format="#0"/>e-6;
C2y=<VAR name="y9" format="#0"/>e-6;

g2=move(g2,[C2x,C2y,0]);
...
```

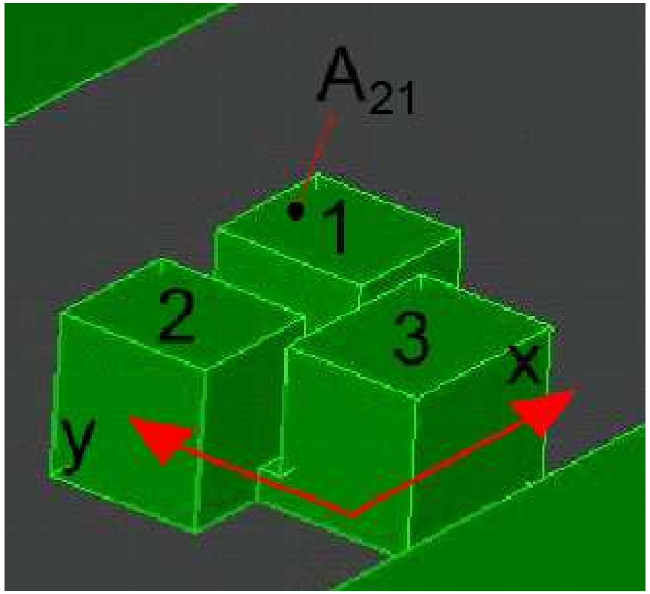
- Each solder ball is moved in Comsol Multiphysics from the baseline position to a new location
- The move command is recorded in the .m file and 24 input parameters are ready for use
- modeFRONTIER will replace the baseline parameter values with the values of the current design

Results, minimize offset error

Offset error = $|F(D_{11},D_{12},D_{13},D_{14},D_{15},D_{16}) + F(D_{31},D_{32},D_{33},D_{34},D_{35},D_{36})| + |F(D_{21},D_{22},D_{23},D_{24},D_{25},D_{26}) + F(D_{41},D_{42},D_{43},D_{44},D_{45},D_{46})|$, where average displacement on top surface of an anchor is defined as

$$D_{ij} = \frac{1}{A_{ij}} \int_{A_{ij}} u_{ij}(x,y)\,dy\,dx$$

i=1,2,3,4 identifies the sensor and j identifies the anchor within the sensor. i=1,3 are sensors measuring acceleration in x-direction, i=2,4 are sensors measuring acceleration in y-direction. *u* is either x- (i=1,3) or y-displacement (i=2,4)

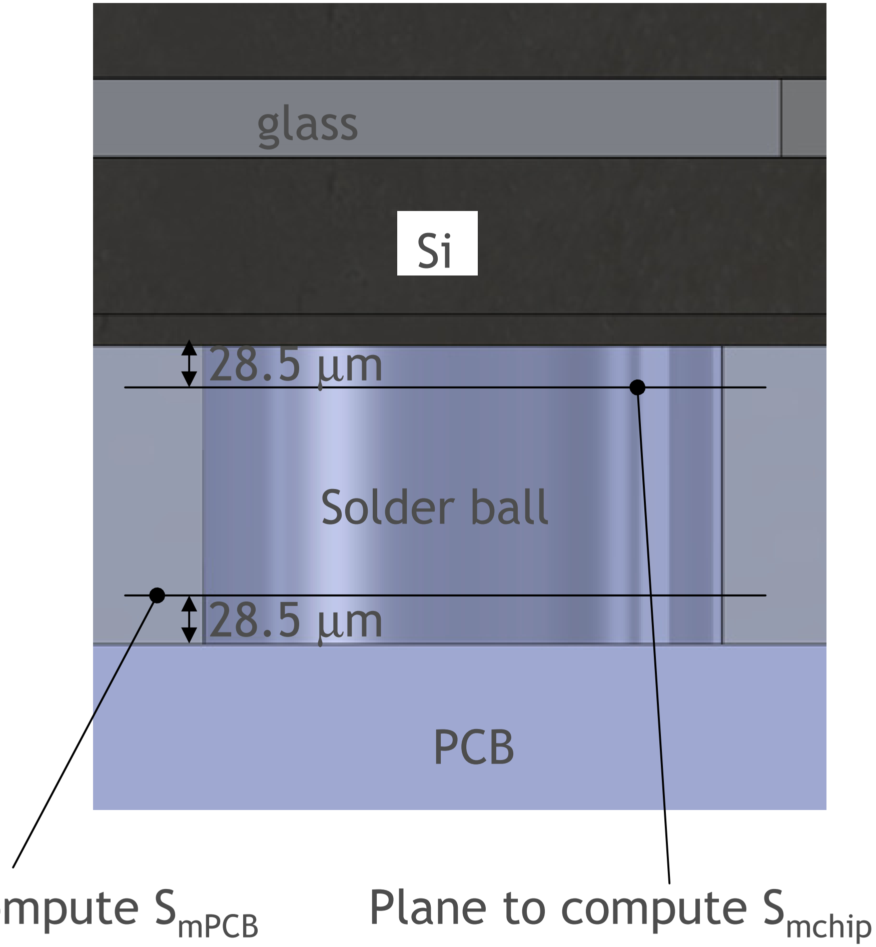


Results, maximize service life

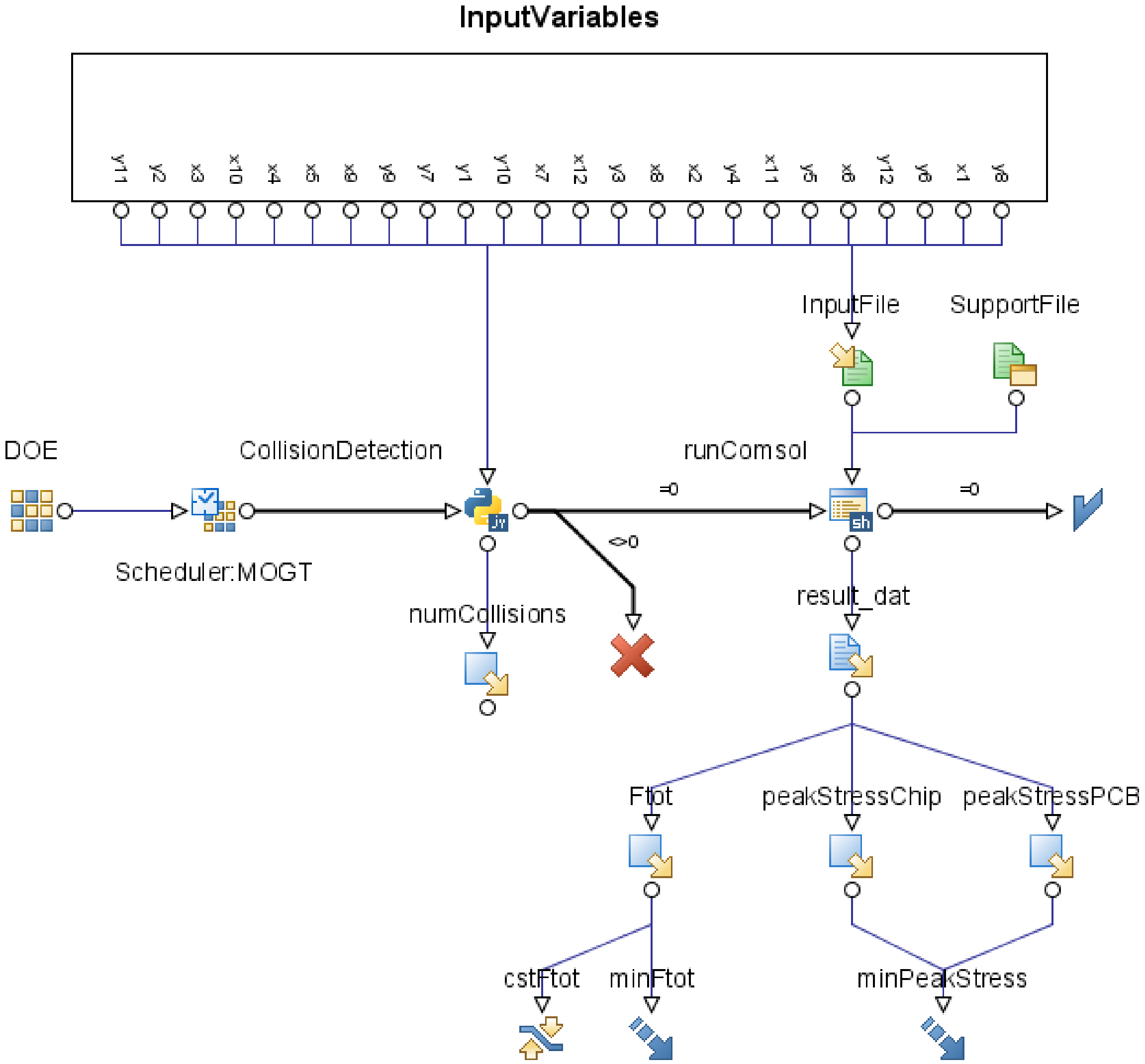
Service life may be maximized by minimizing the peak shear stress among the solder balls according to:

$$\sqrt{S_{\mathrm{mchip}}^{2} + S_{\mathrm{mPCB}}^{2}}, \text{ where } S_m = \frac{\sigma_1 - \sigma_3}{2}$$

where S_{mchip} is the maximum shear stress close to the sensor side of the solderballs, S_{mPCB} is the maximum shear stress close to the PCB side of the solderballs and σ_1 and σ_3 are tensile and compressive principal stresses, respectively.



Define the design logic in modeFRONTIER



24 input variables

Each design is tested for collisions, only zero-collision designs are run in Comsol

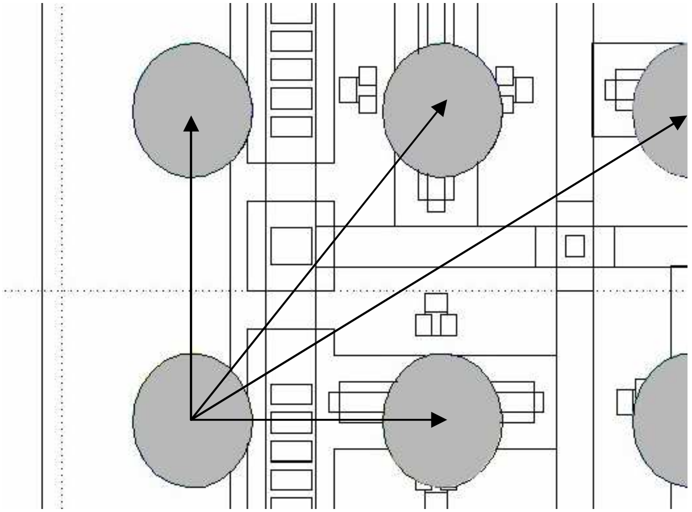
3 results

2 objectives

1 constraint

modeFRONTIER workflow

Collision detection

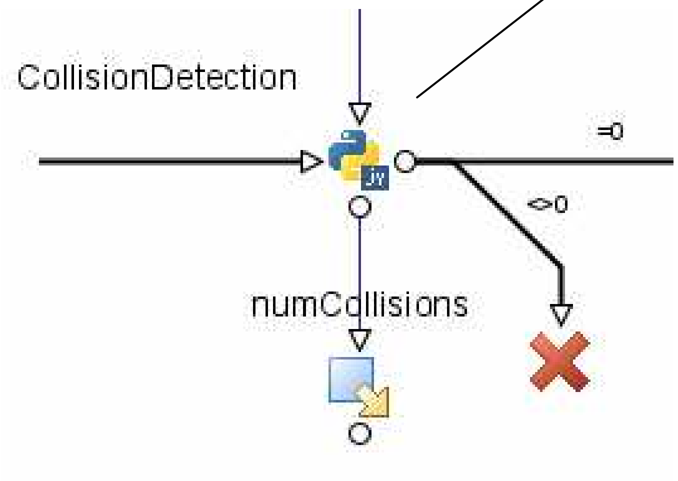


```
# Position for the solder joints, unit um
x = [ x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12 ]
y = [ y1, y2, y3, y4, y5, y6, y7, y8, y9, y10, y11, y12 ]

collisionLimit = 500
numCollisions = 0
for i in range(0, len(x)-1):
     for j in range(i+1, len(y)):
         dist =((x[i]-x[j])**2+(y[i]-y[j])**2)**0.5
         if ( dist < collisionLimit ):
             numCollisions += 1

print 'Number of collisions =', numCollisions

if ( numCollisions > 0 ):
    exitCode=1
else:
    exitCode=0   # Normal termination
```

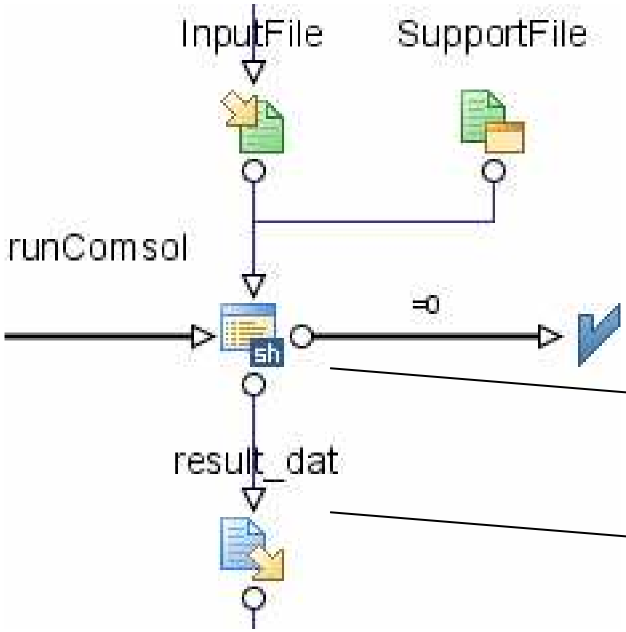


- Simple 2D collision detection implemented in the built-in script language jython (python in Java)
- Execution time: ~0.05 sec

Running Comsol Multiphysics in batch

- InputFile: The parametric .m file
- SupportFile: The parasolids geometry

- Comsol & modeFRONTIER was run on 64-bit Linux
- 9 – 15 min solution time

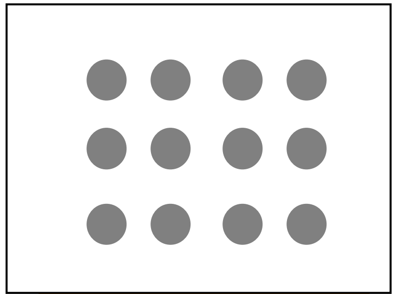


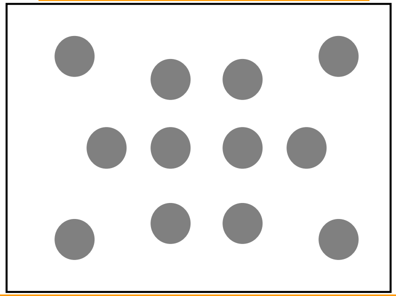

```
comsol -np 2 matlab path -ml -nodesktop
       -ml -nosplash -mlr chipscript
```

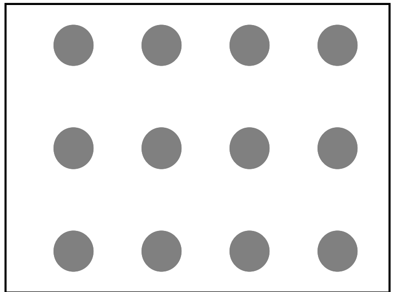
```
3.1096630e+00 7.7078782e-10 -4.9616820e-11 …
```

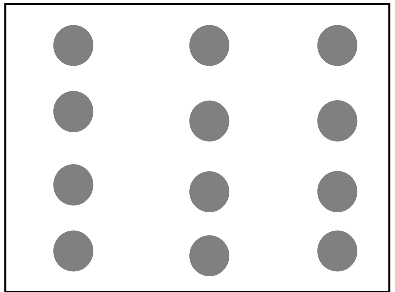
Optimization strategy

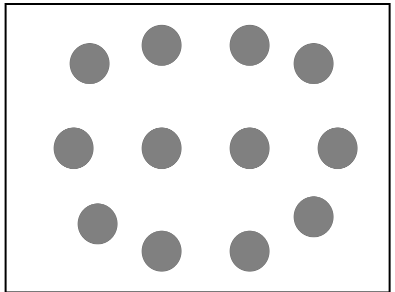
- A genetic algorithm our first choice here
  - Recommended generation size = 50
- Collisions are not allowed
- How to create the initial population
  - Design of Experiments: Sobol 256000
    - Runtime approx 1.5 hour
    - All had collisions
  - Manual editing
    - 48 values for each design x 6 = 288
    - Approx 1 hour later = 6 designs

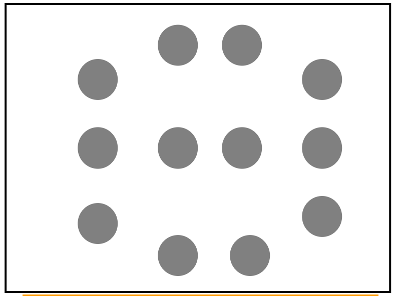

Meshing error


Too many elements


OK!


Too many elements


Failed to mesh


Failed to mesh

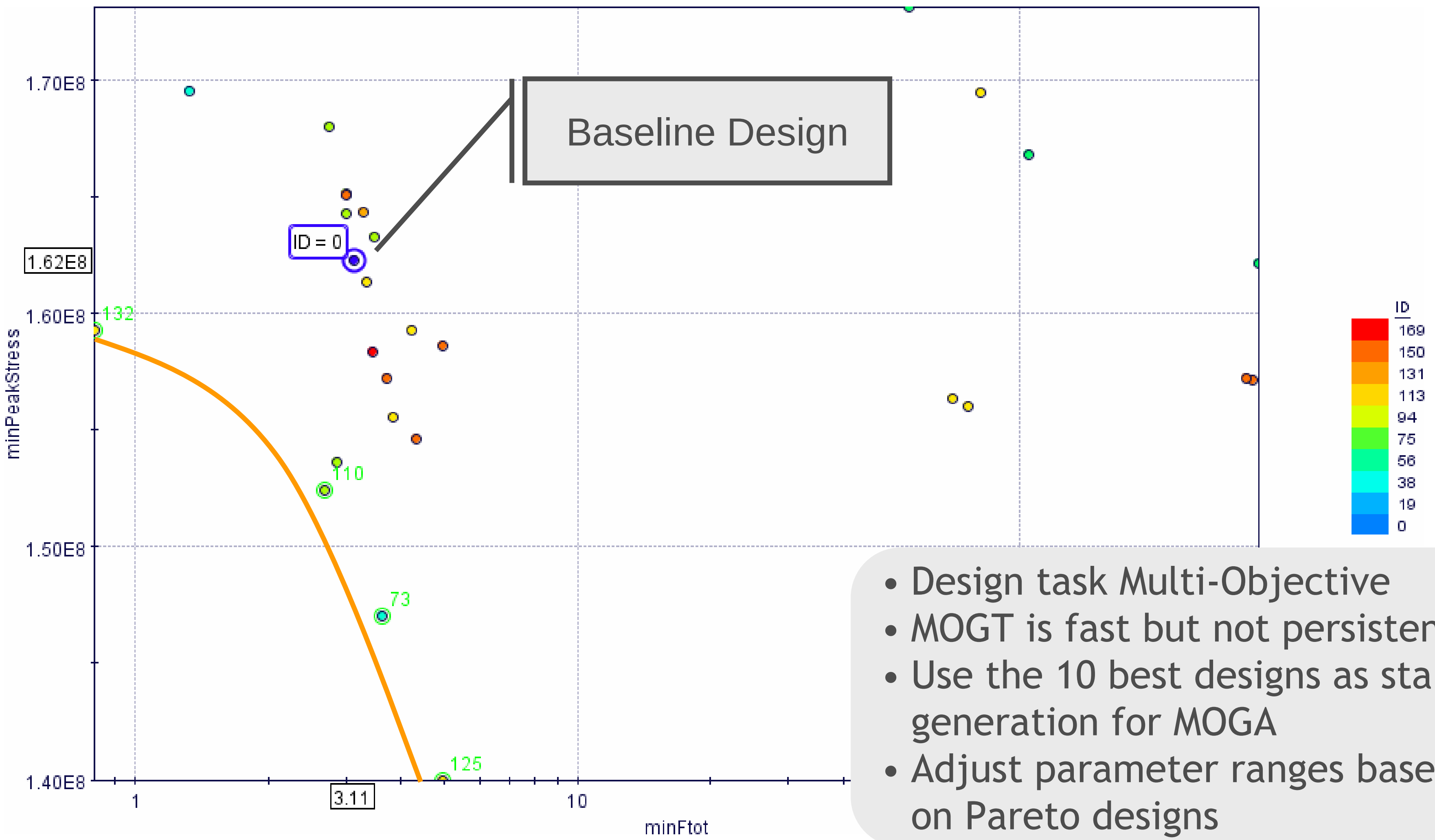
Multi-Objective Game Theory

Designs Table

ID	RID	y6	y7	y8	y9	Ftot	numCollisions	peakStressChip	peakStressPCB	minFtot	minPeakStress	cstFtot
71		1025	1025	1025	1785	□	5	□	□	□	□	□
73		1167	1206	1031	1746	3.606	0	1.0591E8	1.0152E8	3.606	1.467E8	3.606
74		946	997	1069	1550	□	1	□	□	□	□	□
75		1049	856	842	1777	□	4	□	□	□	□	□
76		924	1073	985	1629	□	2	□	□	□	□	□
77		854	1056	956	1598	1.324	0	1.1565E8	1.2355E8	1.324	1.692E8	1.324
78		1177	898	961	1751	□	1	□	□	□	□	□
79		1008	889	1166	1565	□	1	□	□	□	□	□
80		1099	1190	1125	1773	□	3	□	□	□	□	□
81		1088	1197	959	1539	□	4	□	□	□	□	□
82		942	885	854	1726	□	3	□				
83		1091	962	1100	1617	□	2					
84		895	1040	1103	1690	□	4					
85		789	1000	798	1730	□	6					
86		927	1308	685	1646	□	8					
87		1025	1025	1025	1785	□	3					
88		1059	1087	872	1785	□	4					
89		1025	1025	1025	1785	□	0					
90		974	1019	1020	1617	□	0					
92		1167	1206	1031	1746	□	0					
93		1167	1206	1031	1746	346.032	0	1.151				
94		1167	1206	1031	1746	55.851	0	1.212				
95		1167	1206	1031	1746	104.441	0	1.132				
96		1167	1206	1031	1746	□	1					
97		1167	1206	1031	1746	□	1					
98		1167	1206	1031	1746	□	1					
99		1167	1206	1031	1746	□	1					
100		1167	1206	1031	1746	□	3					
101		1167	1206	1031	1746	□	1					
102		1167	1206	1031	1746	□	6					
104		1096	1116	1028	1766	3.458	0	1.1839				
105		1057	1102	1050	1648	2.986	0	1.0787E				
106		1108	1031	937	1762	□	1	□	□	□	□	□
107		1046	1140	1008	1688	2.740	0	1.1493E8	1.2210E8	2.740	1.677E8	2.740

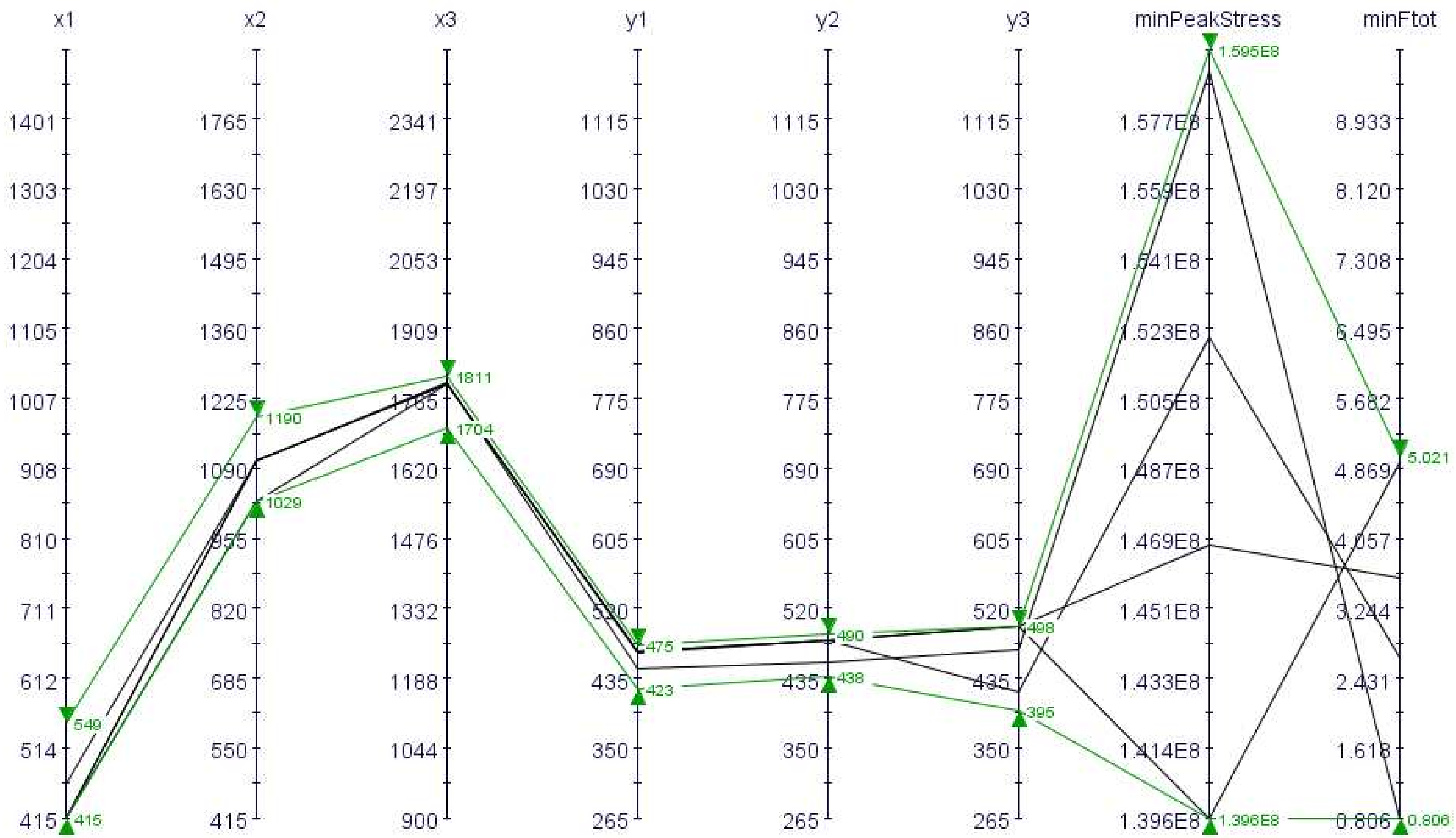
- Let the Game Theory algoritm search for feasible designs
- The highlighed column shows the number of collisions
- Color codes
  - Red = collisions
  - Yellow = violate constraint
  - White = feasible design
- Numerous collisions but after ~5 hours some good designs were found

Multi-Objective Game Theory, 5 h later



- Design task Multi-Objective
- MOGT is fast but not persistent
- Use the 10 best designs as start generation for MOGA
- Adjust parameter ranges based on Pareto designs

Reduction of parameter ranges

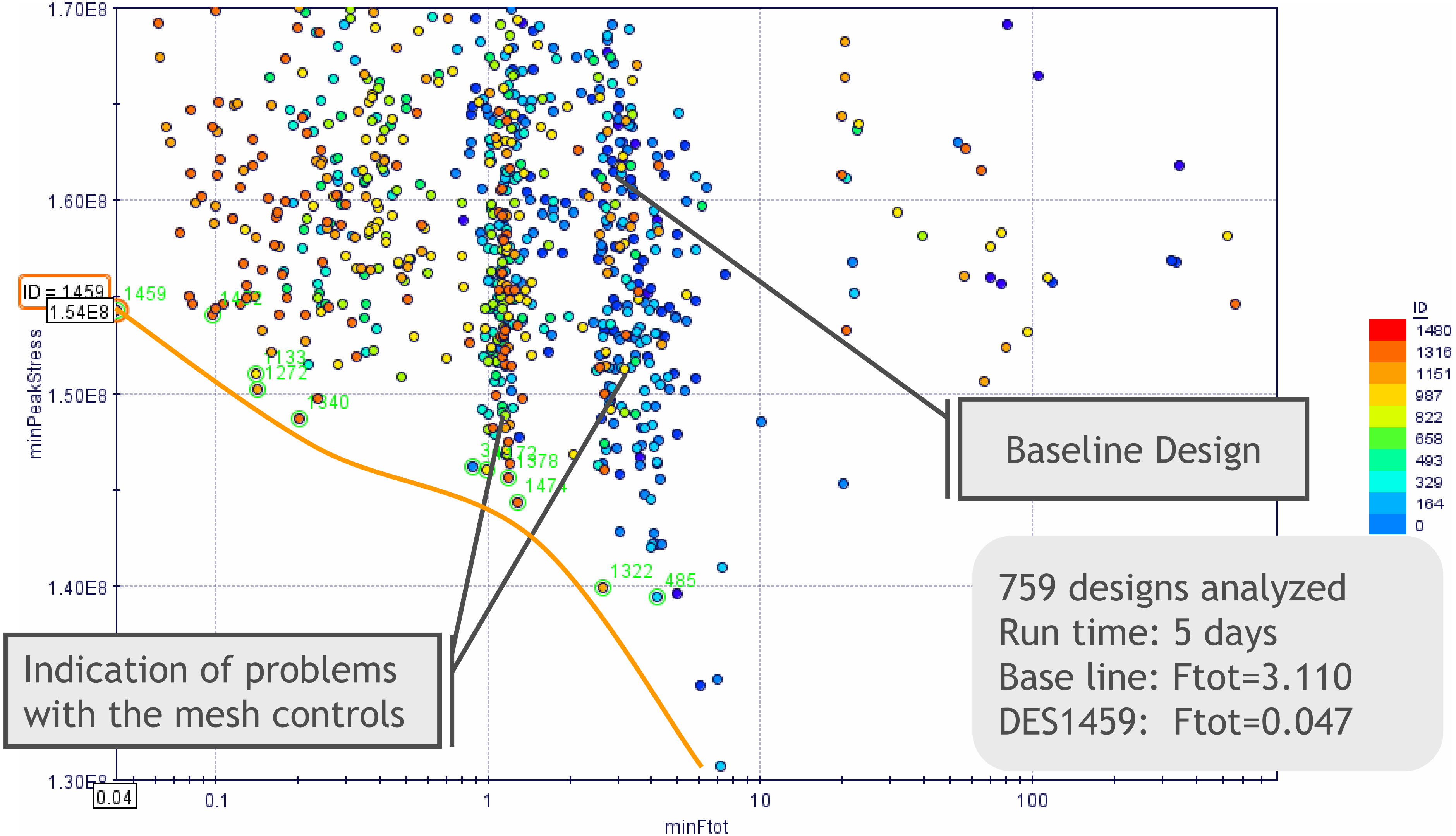


Parallel coordinates chart

- Shows in- and output parameters
- Axes show specified parameter range

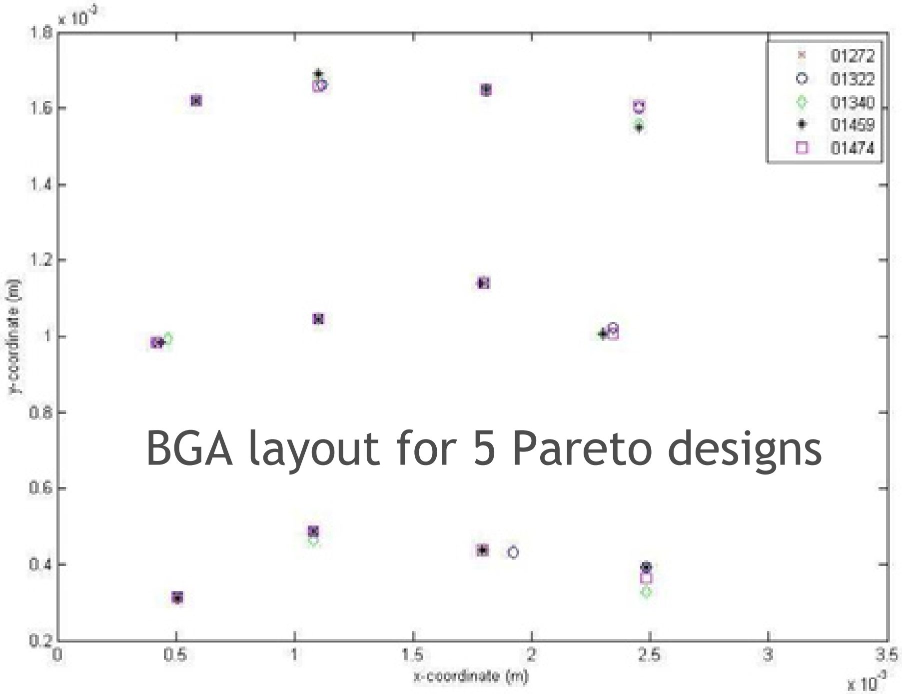
- Pareto designs focused to a small region
- Reduce input parameter ranges to
  - Simplify the learning process
  - Reduce crashes

Multi-Objective Genetic Algorithm

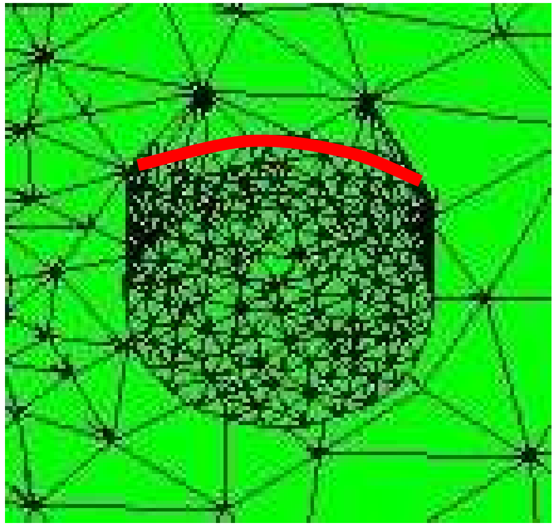


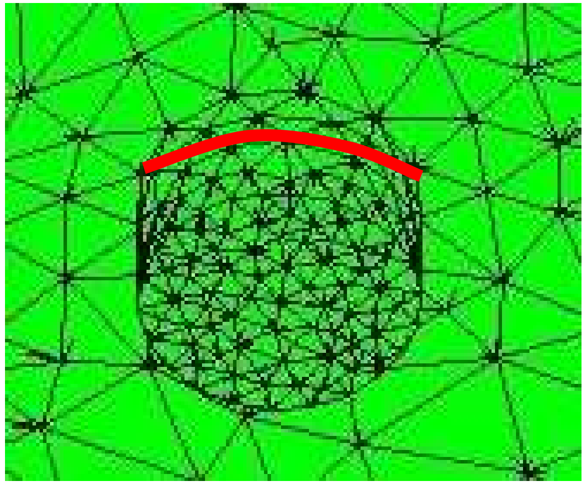
759 designs analyzed
Run time: 5 days
Base line: Ftot=3.110
DES1459: Ftot=0.047

Results



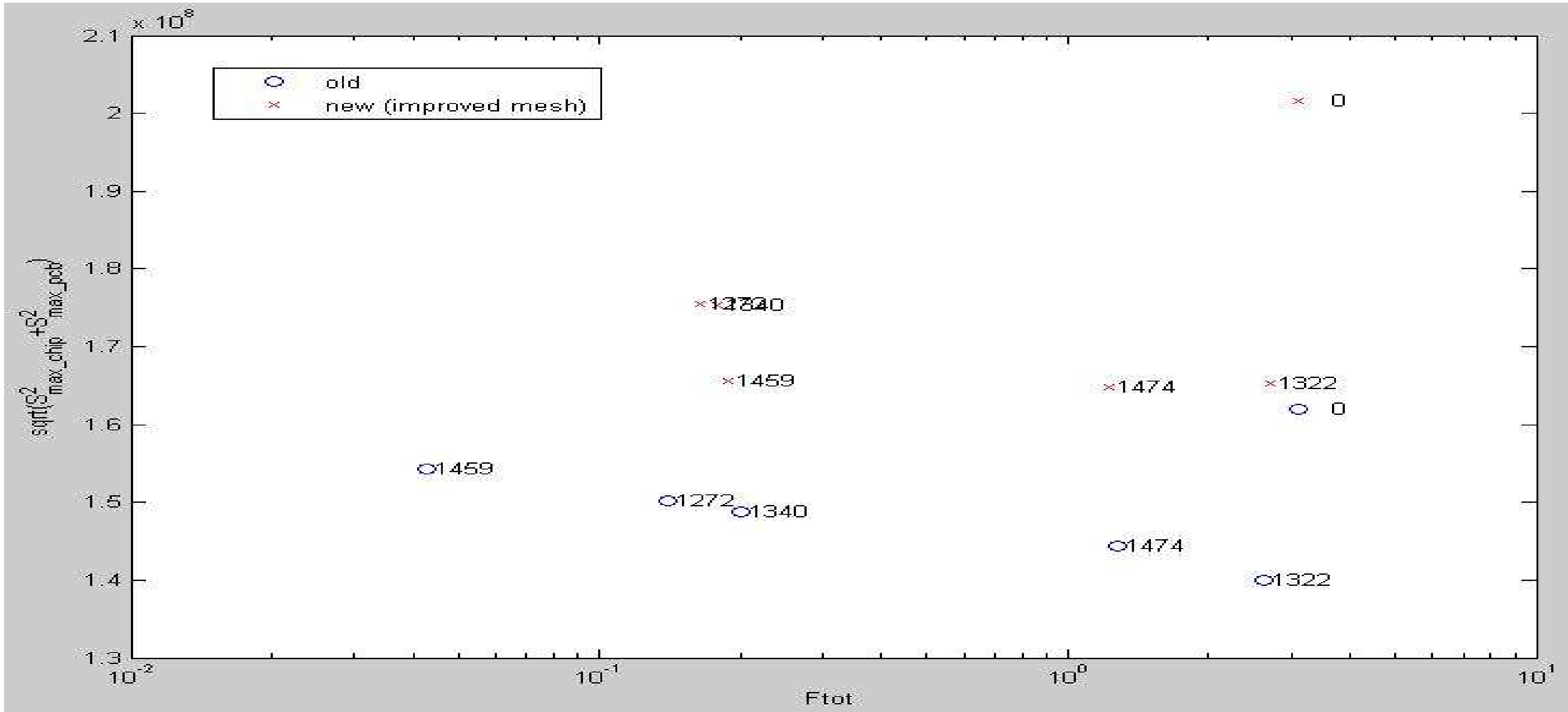
BGA layout for 5 Pareto designs





- Subtle, close-to-invisible changes have great effect on the results
- The numbering of the solder ball geometry was not kept constant in all design configurations
  - Lost mesh control settings
  - Coarse default mesh size

Verification with a fine mesh

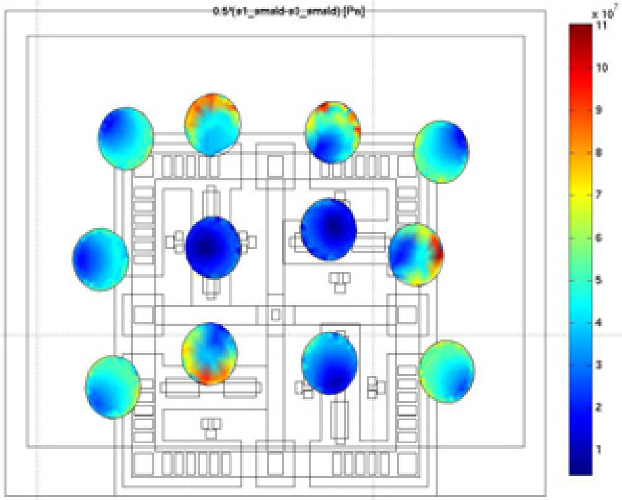


- The five Pareto designs were reanalyzed with a more accurate mesh

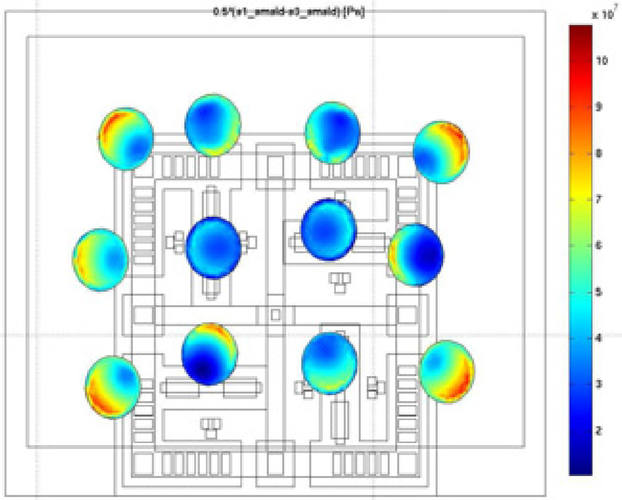
Base line:	Ftot = 3.1	PeakStress = 202
DES1459:	Ftot = 0.18	PeakStress = 165
	94%	18%

Conclusions

- Improved design found
  - Facilitates to meet stringent specifications
  - Lower return of components from the field
- Candidate designs provide better understanding of the relevant aspects in the behavior of the flip-chipped MEMS accelerometer under temperature load
- Methodology grows with the needs
  - Include more physics - capacitance
  - Include complex 3D geometry
  - Robustness studies



DES 1459, close to PCB



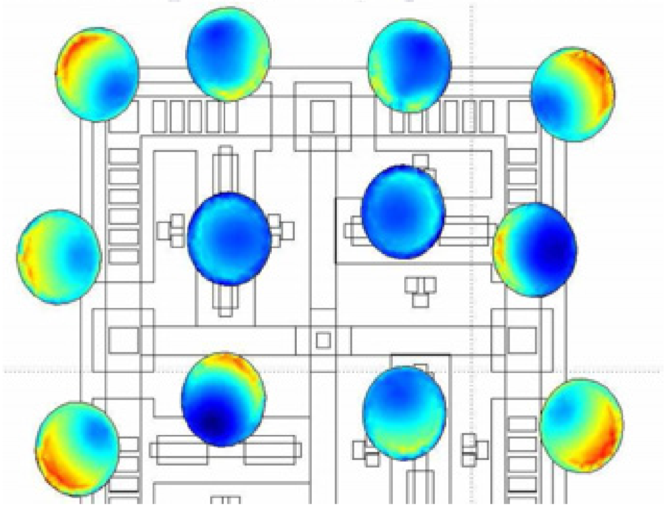
DES 1459, close to Si

Summary

- Significant design improvement
  - Ftot 94% reduction, 99% in opt
  - Stress 18% reduction, 5% in opt

- It does work
  - despite harsh environment with numerous crashes
  - 1000+ designs tested automatically at maximum throughput

- modeFRONTIER and Comsol Multiphysics help you drive your designs from **'good'** to **'GREAT'**



Thank you for your attention!

hakan.strandberg@esteconordic.se