Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

A Topology Optimization Interface for LS-Dyna

Nikola Aulig, Ingolf Lepenies

2012

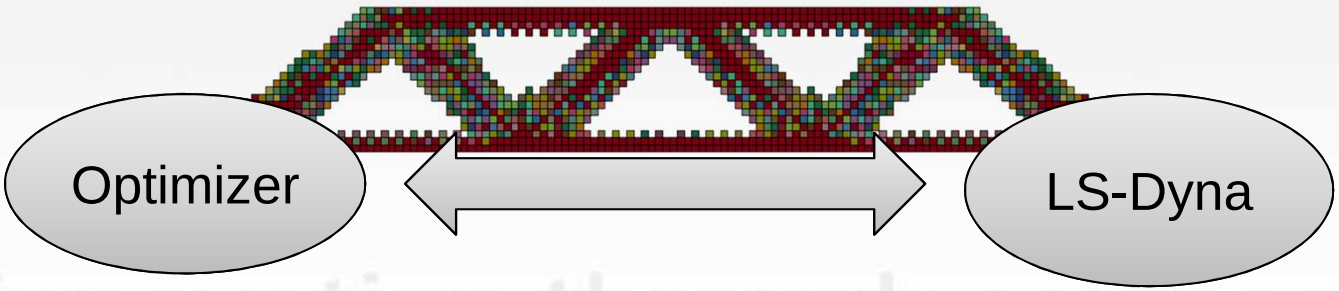
Preprint:

This is an accepted article published in LS-Dyna Forum 2012. The final authenticated version is available online at: https://doi.org/[DOI not available]

Honda Research Institute EU

A Topology Optimization Interface for LS-Dyna

Dipl.-Ing. Nikola Aulig[1], Dr.- Ing. Ingolf Lepenies[2]



innovation through science

[1]nikola.aulig@honda-ri.de
Honda Research Institute Europe GmbH,
Carl-Legien-Straße 30, 63073 Offenbach-Main, Germany

[2]ingolf.lepenies@dynamore.de
DYNAmore Gesellschaft fuer Ingenieurdienstleistungen mbH,
Pohlandstr. 19, 01309 Dresden, Germany

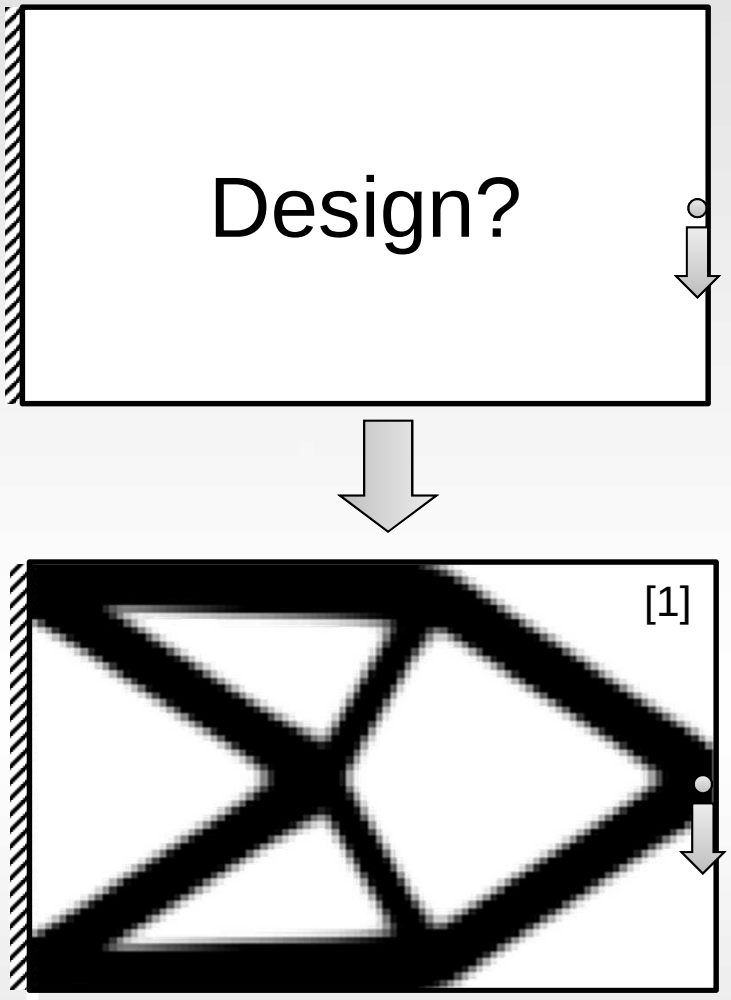
Outline

- **Introduction**
- **Implementation**
- **Application of Topology Optimization**
- **Summary**

Introduction Topology Optimization

Structural optimization

- Sizing / Thickness
- Shape
- **Topology**



Goal of topology optimization

- Make optimal use of material for a given design space
- Find a concept design

Benefits

- Save weight and material
- Shortening of design cycles

[1] O. Sigmund O. A 99 line topology optimization code written in Matlab. Structural and Multidisciplinary Optimization 21(2):120–127, 2001.

Motivation

Application of topology optimization:

- Available in commercial software
- Mostly limited to linear or slightly non-linear problems

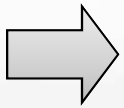
Topology optimization of crash structures

- Difficult to derive sensitivity information
- Approached by LS-TaSC [2,3,4]

Research directions:

- Multi-disciplinary problems
- More globally optimal solutions

Problem: How to couple topology optimization algorithm to LS-Dyna?



Solution: Develop interface between optimizer and LS-Dyna

[2] Roux, W.: “Topology Design using LS-TaSC™ Version 2 and LS-DYNA”, 8th European LS-DYNA Users Conference, 2011
[3] Goel T., Roux W., and Stander N.: “A Topology Optimization Tool for LS-DYNA users: LS-OPT/Topology”, 7th European LS-DYNA Users Conference, 2009.
[4] Patel, N. M.: “Crashworthiness Design using Topology Optimization”, PhD thesis, University of Notre Dame, 2007

SIMP based topology optimization

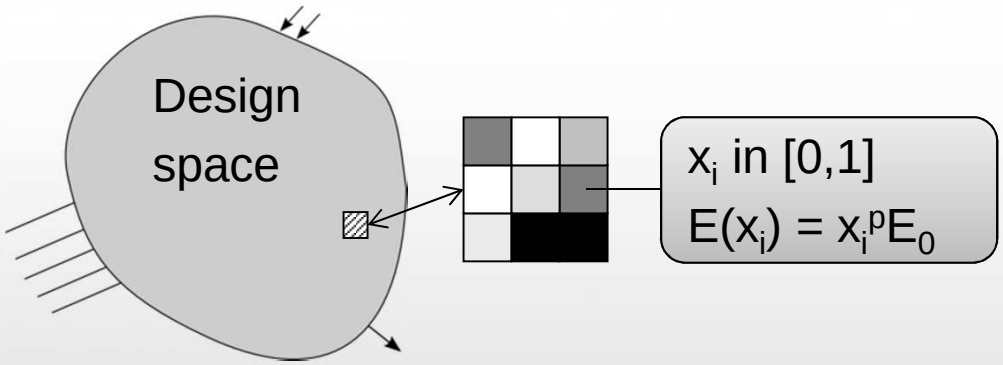
Interface requirements:

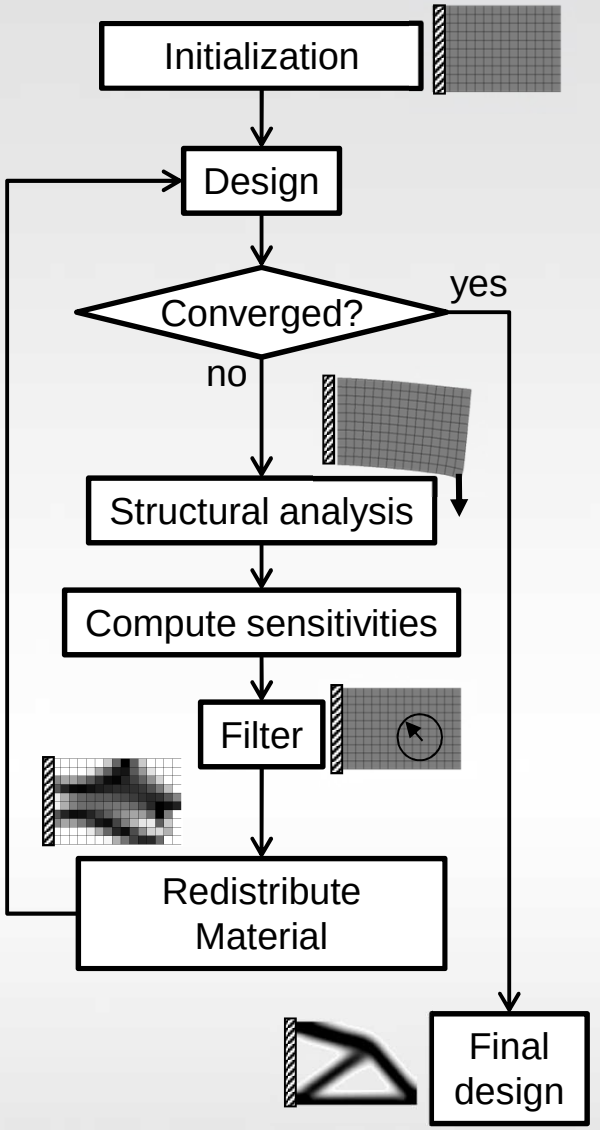
- Facilitate implementation of existing algorithms
- Flexibility to develop novel methods

Commonly approach to topology optimization

- Every finite element is assigned a density variable
- Material is redistributed iteratively, based on sensitivities
- Material properties are interpolated:

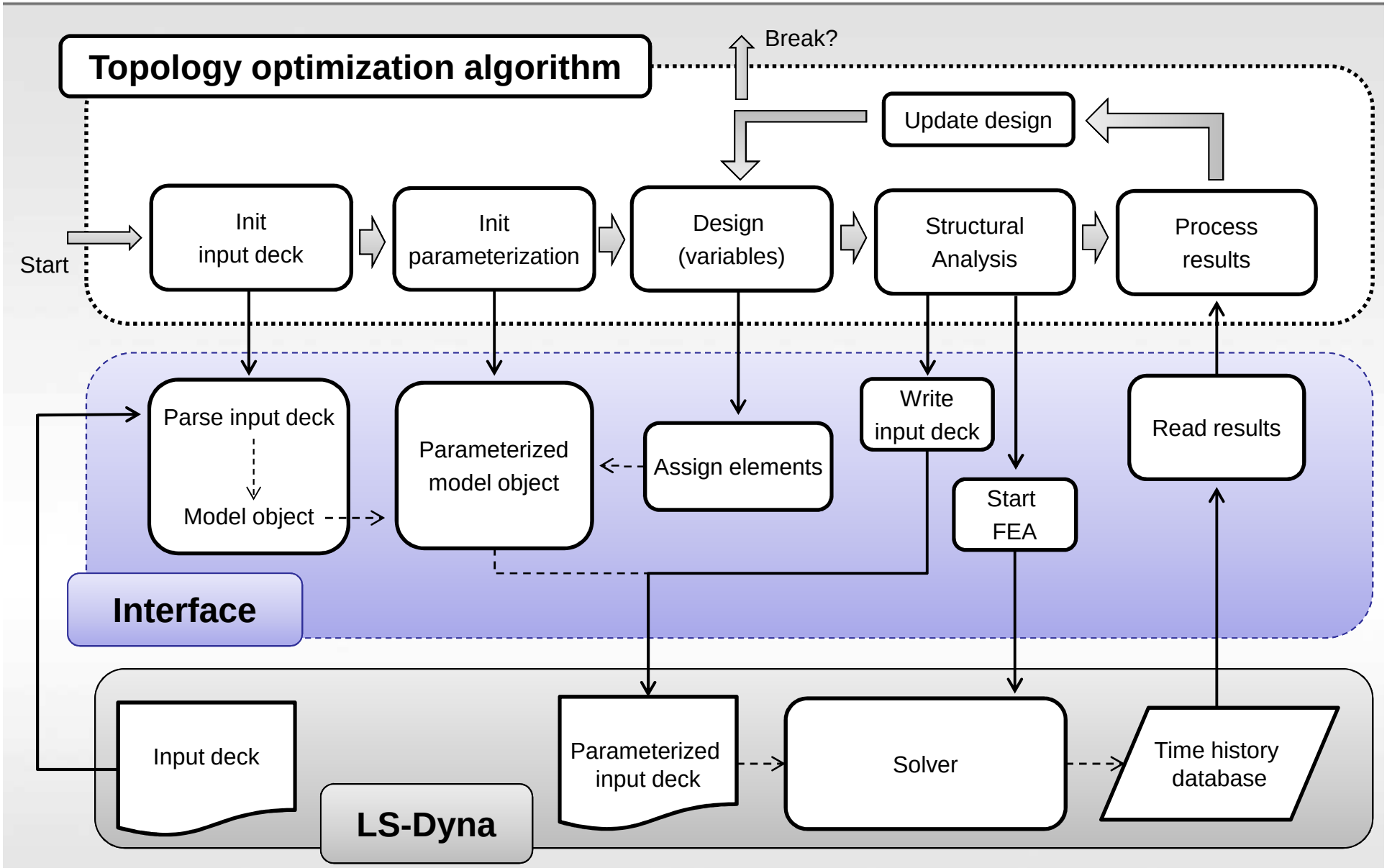
  Solid Isotropic Material with Penalization (SIMP) [5]



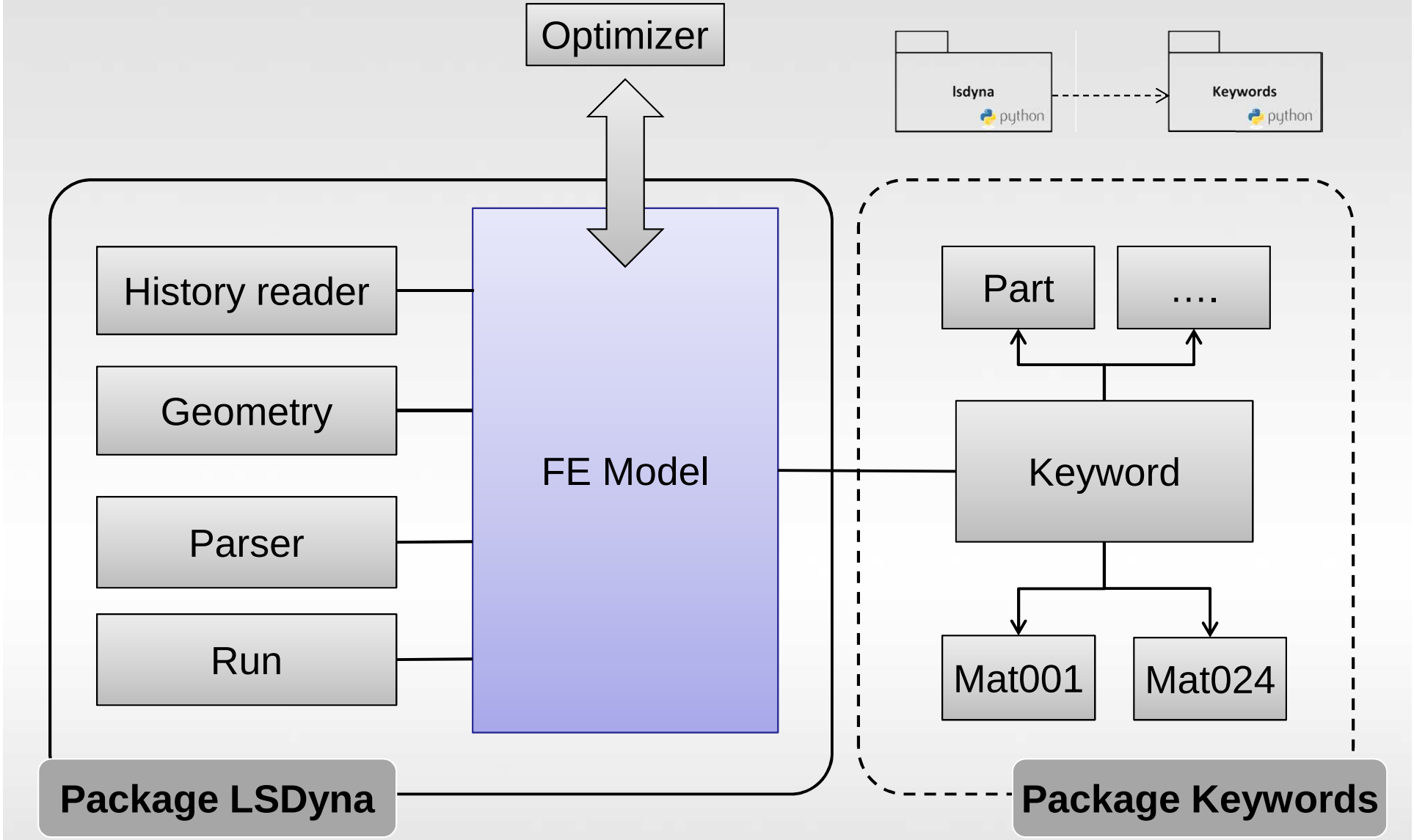


[5] Bendsoe, M.P., Sigmund, O., Topology Optimization, Theory, Methods and Applications, *Springer*, 2nd Ed., 2004.

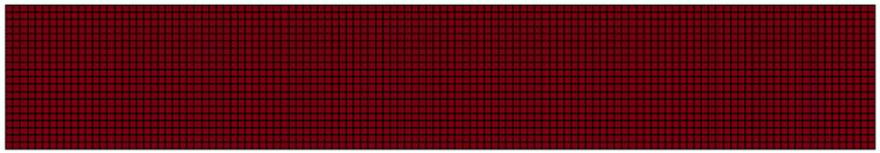
Overview Process Flow



Implementation Overview



```
import lsdyna

# declare FEModel object
femodel = lsdyna.FEmodel()
femodel.initInputDeck('model.key', [1])
```



Original input deck

Material Parameterization

```
*MAT_ELASTIC
      1 2.7000E-9  1.000000  0.330000     0.000     0.000
```

```
femodel.initInputDeck('model.key', [1])
```

```
# Parameterize the femodel
femodel.parameteriseMaterial('lin', 100)
```

```
*MAT_ELASTIC
$# copy of material 1, scaled by factor 0 for normalized density range [ 0.    0.01]
      2  2.700-09  1.000-10  3.300-01
*MAT_ELASTIC
$# copy of material 1, scaled by factor 0.01 for normalized density range [ 0.01  0.02]
      3  2.700-09  1.000-02  3.300-01
*MAT_ELASTIC
$# copy of material 1, scaled by factor 0.02 for normalized density range [ 0.02  0.03]
      4  2.700-09  2.000-02  3.300-01
                                                    :
```

Material Parameterization

```
import lsdyna

# declare FEModel objec
femodel = lsdyna.FEmode
femodel.initInputDeck('model.key', [1])
```

```
*ELEMENT_SHELL
    1    1    12    342    346    345    0    0    0    0
    2    1    13    347    350    343    0    0    0    0
    3    1    14    351    354    348    0    0    0    0
                                        :
```

```
# Parameterize the femodel
femodel.parameteriseMat
```

```
*ELEMENT_SHELL
    1    93    12    342    346    345    0    0    0    0
    2    87    13    347    350    343    0    0    0    0
    3    91    14    351    354    348    0    0    0    0
```

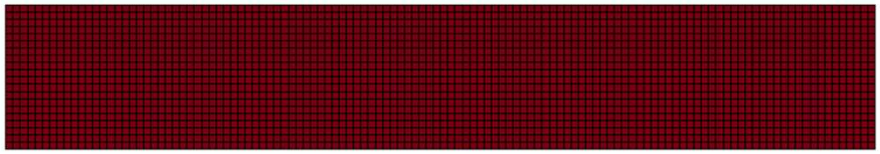
```
# Do some optimization

# Assign density values to the elements
femodel.writeDesignVariables(designVariables)

# Create an ls dyna Input deck
run = femodel.createRun([…])
femodel.createInputDeck(run)
```

Material Parameterization

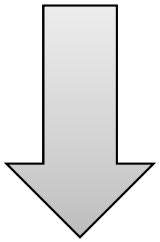
```
import lsdyna

# declare FEModel object
femodel = lsdyna.FEmodel()
femodel.initInputDeck('model.ke
```

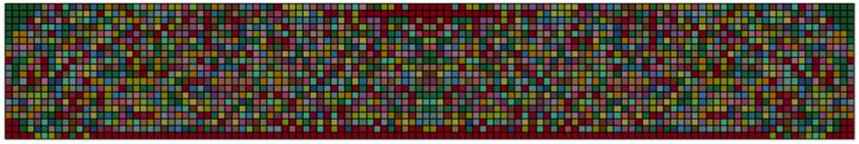
```
# Parameterize the femodel
femodel.parameteriseMaterial('
```

```
# Do some optimization […]

# Assign density values to the
femodel.writeDesignVariables(de

# Create an ls dyna Input deck
run = femodel.createRun([…])
femodel.createInputDeck(run)
```



Original input deck



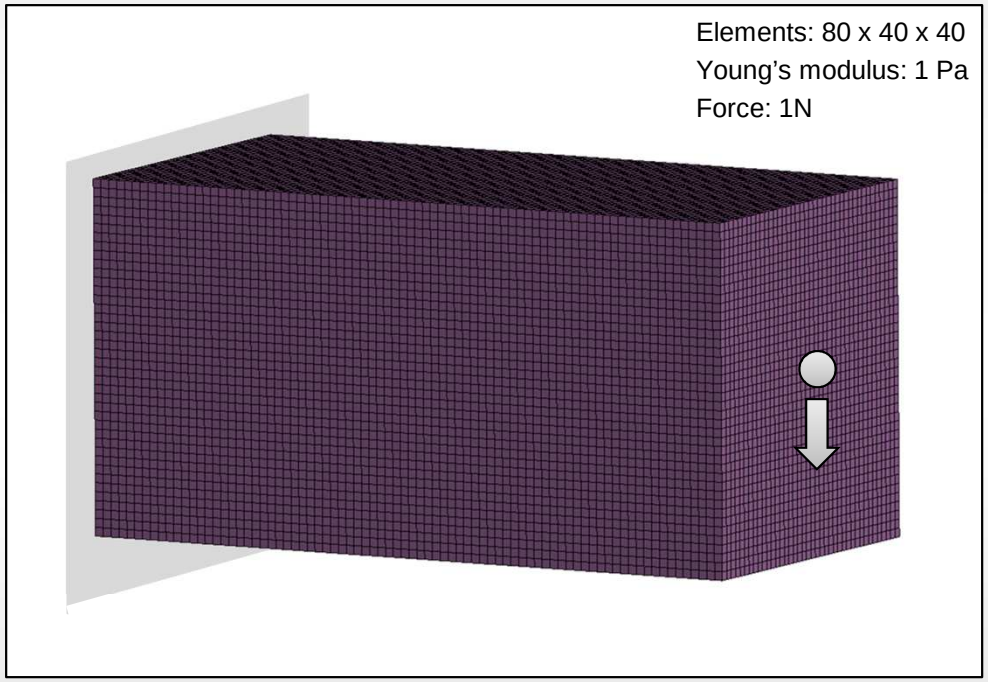


Parameterized input deck

First target:

- Implement standard algorithm [1]
- Use interpolation scheme

⇨ SIMP with optimality criteria update for stiffness optimization

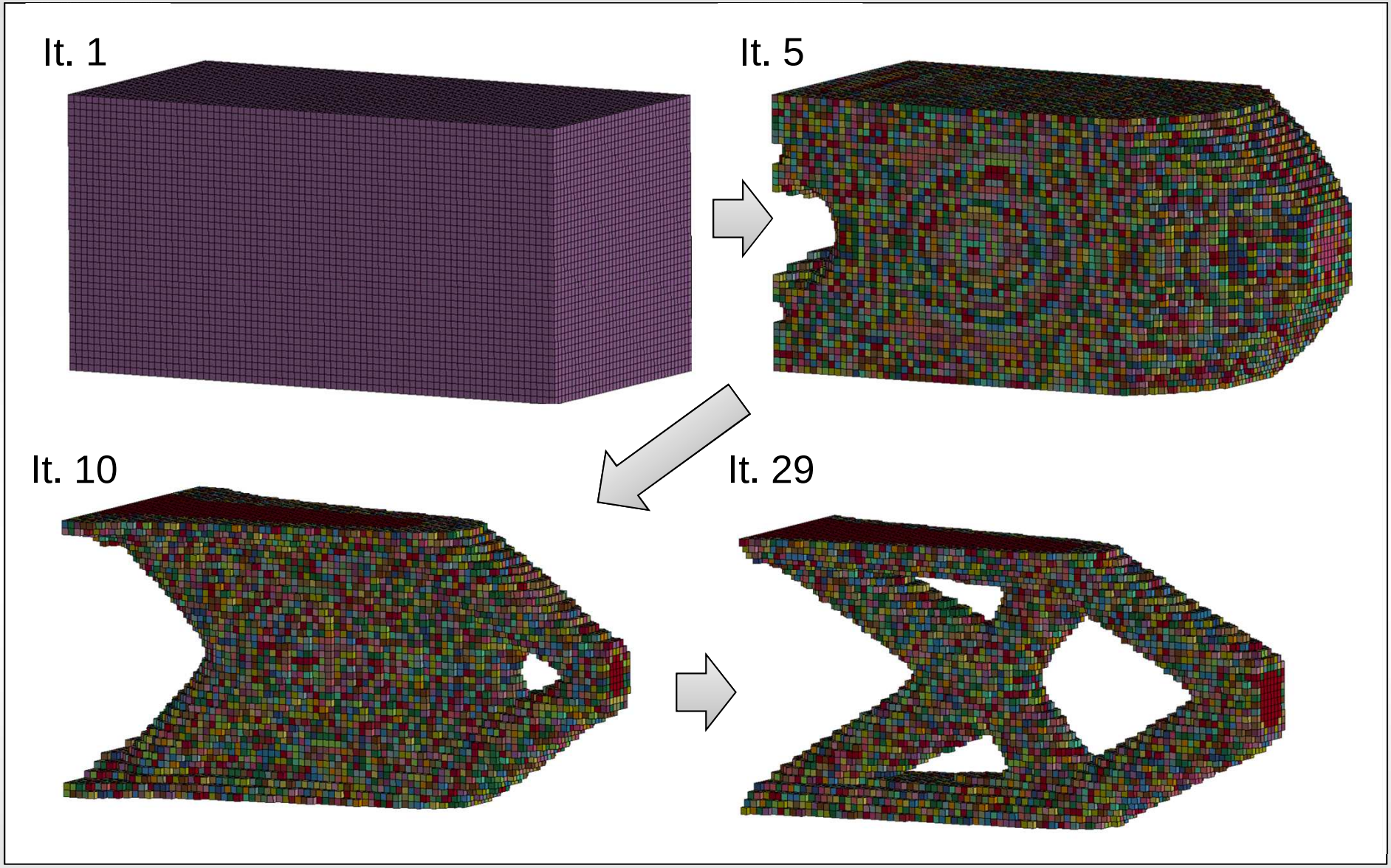


Test case:

- Short 3D cantilever
- 10% volume fraction
- One static load applied
- 128,000 solid elements
- Analysed with LS-Dyna implicit solver

[1] Sigmund, O. A 99 line topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization* 21(2):120-127, 2001

Example 1 Result

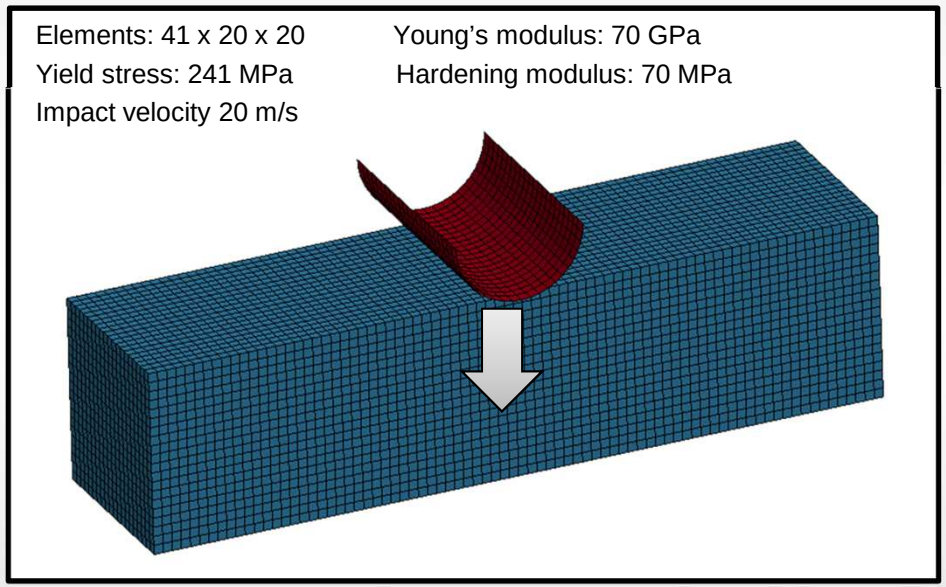


Second target:

- Tackle optimization of energy absorbing structures

Bi-directional evolutionary structural optimization (BESO) [5]

- Elements are assigned a heuristic “sensitivity number”
- No interpolation, complete elements are removed or added

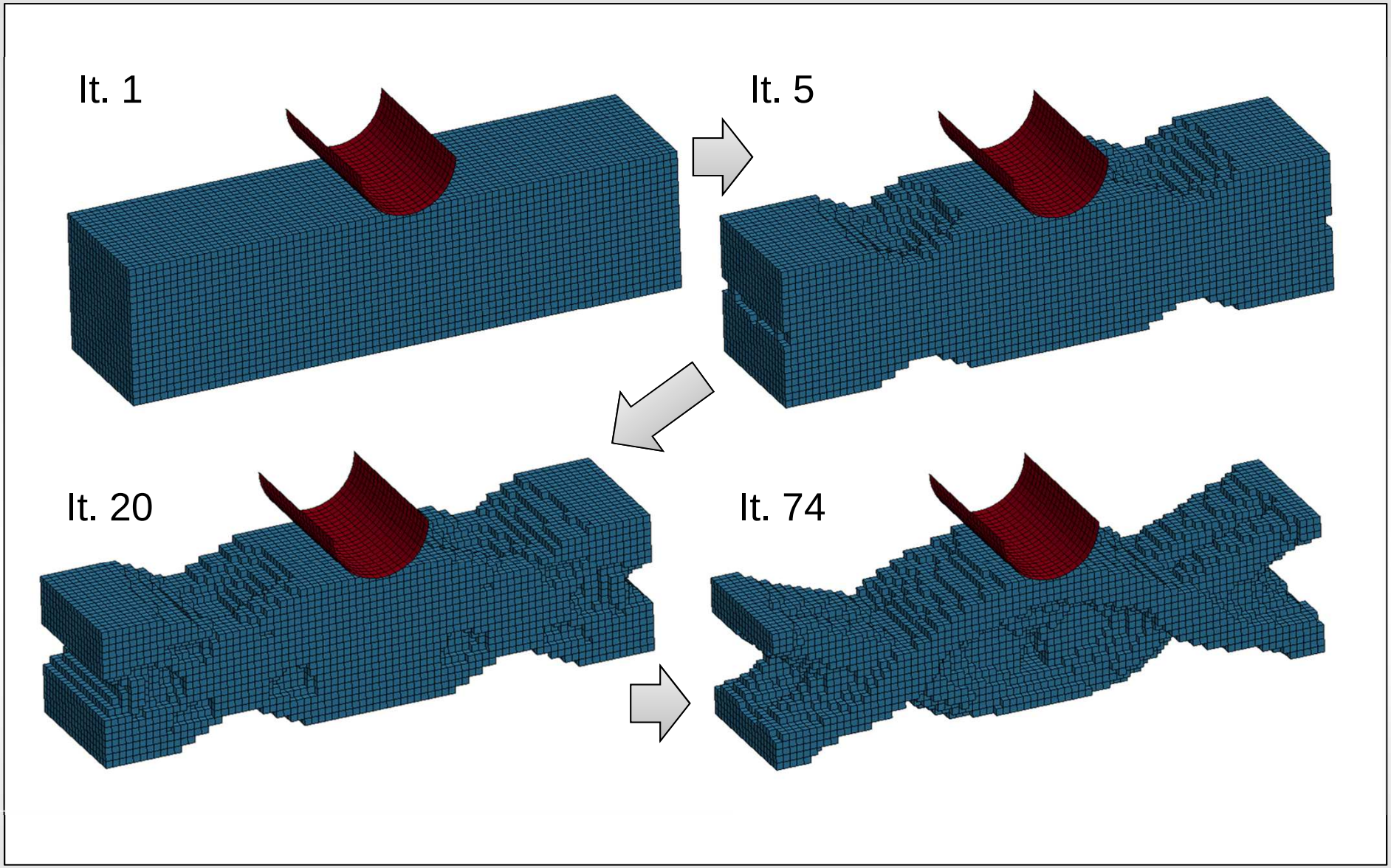


Test case:

- Pole impacting a beam, fixed on both sides
- Piecewise linear-elastic plastic material model
- 50% volume fraction

[5] Huang, X., Xie, Y.M., Lu, G. Topology optimization of energy absorbing structures. *International Journal of Crashworthiness* 12(6):663-75. 2007

Example 2 Result



Target: Research of topology optimization of non-linear problems

⇨ Necessity for coupling of LS-Dyna and optimizer

Interface Summary

- Parsing
- Parameterization
- Creation of input deck
- **Application:** Minimize compliance, maximize energy absorption

Future directions

- Extend for more material cards, ESL method (Interface)
- Tackle multi-disciplinary problems (Optimization)

Thank you for your attention!

Appendix: Class Structure Implementation

