Large Scale PDE Optimization with FAIPA,

the Feasible Arc Interior Point Algorithm

José Herskovits (*)
José Miguel Aroztegui (*)
Alfredo Canelas (**)

(*) OptimizE - Engineering Optimization Lab
Mechanical Eng. Program, COPPE
Federal University of Rio de Janeiro

(**) Instituto de Estructuras y Transporte
Facultad de Ingeniería - Uruguay

Introduction

- We consider the nonlinear constrained optimization program:

$$\left.\begin{array}{rl}\underset{x}{\text{minimize}} & f(x), \quad x \in R^n \\ \text{subject to} & g(x) \le 0; \quad g \in R^m \\ \text{and} & h(x) = 0; \quad h \in R^p\end{array}\right\} \quad (1)$$

where:

$x \equiv [x_1, x_2, \ldots, x_n]$ is the vector of unknowns,

$f(x)$ is the objective function,

$g(x)$ and $h(x)$ are the inequality and equality constraints.

- A Partial Differential Equation must be solved to compute the objective function and/or the constraints.

Introduction

- PDE Constrained Optimization appears in:
  - Optimal Design
  - Optimal Control
  - Parameters Estimation

of systems governed by Partial Differential Equations

- The size and complexity of the discretized PDE often pose significant challenges for optimization methods.

- Real applications generally require a large number of variables and constraints.

Introduction

- As an example we consider an engineering model for Topology Structural Optimization with Finite Elements:
  - Design variables:
    - Elements thickness
  - Minimize an objective:
    - Weight, cost,…
  - With mechanical constraints:
    - Local stress at each finite element
    - Nodal displacements
    - Natural frequencies, etc.

- We have a very large number of variables and constraints.

Introduction

- To solve the optimization problems we use **FAIPA**, the Feasible Arc Interior Point Algorithm for Nonlinear Constrained Optimization.
- We will show that including some new or existing numerical tools in FAIPA, we can solve efficiently a class of real life PDE Optimization problems.
- The computer memory requirement of the present technique is very small.

FAIPA

Feasible Arc Interior Point Algorithm

- FAIPA is a general technique to solve nonlinear constrained optimization problems.
- It requires an initial point at the interior of the inequality constraints and generates a sequence of interior points.
- When the problem has only inequality constraints the objective function is reduced at each iteration.
- Iterating in the primal variables (x) and in the dual variables (Lagrange multipliers), FAIPA finds a local minimum characterized by the Karush-Kuhn-Tucker conditions

FDIPA
Feasible Direction Interior Point Algorithm

- We consider first FDIPA and discuss the ideas involved in this approach in the framework of the inequality constrained problem:

$$\left.\begin{array}{rl} \text{minimize} & f(x) \\ \text{submitted to} & g(x) \le 0 \end{array}\right\} \qquad (2)$$

Let the feasible set be: $\Omega \equiv \{ x \in R^n / g(x) \le 0 \}$

Basic Ideas

- **Karush-Kuhn-Tucker** optimality conditions:
  - If x is a local minimum, then

$$\begin{aligned} \nabla f(x)+\nabla g(x)\lambda=0, \\ G(x)\lambda=0, \\ g(x)\leq 0 \text{ and} \\ \lambda\geq 0. \end{aligned} \qquad (3)$$

where $\lambda \in R^m$ are the dual variables and $G(x)$ a diagonal matrix with $G_{ii}(x) = g_i(x)$.

Basic Ideas

- In the present approach, we look for (x, λ) that satisfies the KKT conditions

- We propose a Newton - like iteration to solve the **equalities** in the KKT conditions:

$$\nabla f(x) + \nabla g(x)\lambda = 0$$
$$G(x)\lambda = 0$$

in such a way that each iterate satisfies the **inequalities**:

$$g(x) \le 0$$
$$\lambda \ge 0.$$

Basic Ideas

- The Newton Iteration in (x,λ) for the KKT equality conditions is:

$$\begin{bmatrix} B & \nabla g(x) \\ \Lambda \nabla g^t(x) & G(x) \end{bmatrix} \begin{bmatrix} x_0 - x \\ \lambda_0 - \lambda \end{bmatrix} = - \begin{bmatrix} \nabla f(x) + \nabla g(x)\lambda \\ G(x)\lambda \end{bmatrix} \quad (4)$$

where (x,λ) is the present point, (x_0,λ_0) is the new estimate

Λ is a diagonal matrix such that $\Lambda_{ii} = \lambda_i$.

Basic Ideas

- We can take:

$B = \nabla^2 f(x) + \sum_{i=1}^{m} \lambda_i \nabla^2 g(x)$: **Newton's Method**

$B =$ quase-Newton approx.: **Quasi-Newton**

$B = I$: **First order method**

Basic Ideas

- We define now the vector d_0 in the primal space, as

$$d_0 = x_0 - x \qquad (5)$$

- Then, we have:

$$\begin{aligned} Bd_0 + \nabla g(x)\lambda_0 &= -\nabla f(x) \\ \Lambda \nabla \mathrm{g}^{\mathrm{t}}(x) d_0 + G(x)\lambda_0 &= 0 \end{aligned} \qquad (6)$$

Basic Ideas

- We prove that, if;

- B is **Positive Definite,**
- $\lambda > 0$

and

- $g(x) \leq 0,$

then:

- **The linear system has an unique solution**
- d_0 is a **descent direction** for $f(x)$

However, d_0 is not always a **feasible direction.**

Basic Ideas

In fact, $\Lambda \nabla g^{t}(x) d_0 + G(x)\lambda_0 = 0$

is equivalent to:

$$\lambda_i \nabla g_i^{t}(x) d_0 + g_i(x)\lambda_{0i} = 0; \quad i = 1,\ldots,m \qquad (6)$$

Thus, d_0 is not always feasible since it is tangent to the active constraints.

Basic Ideas

Then, to obtain a feasible direction, a negative number is added to the right hand side:

$$\lambda_i \nabla g_i^t(x)d + g_i(x)\overline{\lambda}_i = \boxed{-\rho\lambda_i\omega_i} \qquad i = 1, \ldots, m,$$

and we get a new perturbed system:

$$\begin{aligned} Bd + \nabla g(x)\overline{\lambda} &= -\nabla f(x) \\ \Lambda \nabla g^t(x)d + G(x)\overline{\lambda} &= -\rho\lambda \end{aligned} \qquad (7)$$

where $\rho > 0$

Basic Ideas

- The negative number in the right hand side produces the effect of bending d_0 to the interior of the feasible region, being the deflection relative to each constraint proportional to ρ.

Basic Ideas

As the deflection is proportional to ρ and d_0 is descent, by establishing upper bounds on ρ, it is possible to ensure that d is also a descent direction.

Since $d_0^t \nabla f(x) < 0,$ (7)

we can obtain these bounds by imposing

$$d^t \nabla f(x) \leq \alpha d_0^t \nabla f(x), \qquad (8)$$

which implies $d^t \nabla f(x) < 0.$

Basic Ideas

Let us consider

$$Bd_0 + \nabla g(x)\lambda_0 = -\nabla f(x)$$
$$\Lambda \nabla g^t(x)d_0 + G(x)\lambda_0 = 0 \qquad (9)$$

And the auxiliary system of linear equations

$$Bd_1 + \nabla g(x)\lambda_1 = 0$$
$$\Lambda \nabla g^t(x)d_1 + G(x)\lambda_1 = -\lambda \qquad (10)$$

Basic Ideas

We have that the solution of

$$Bd + \nabla g(x)\overline{\lambda} = -\nabla f(x)$$
$$\Lambda \nabla g^{t}(x)d + G(x)\overline{\lambda} = -\rho\lambda,$$

is

$$d = d_0 + \rho d_1 \qquad (11)$$

and

$$\overline{\lambda} = \lambda_0 + \rho\lambda_1 \qquad (12)$$

Basic Ideas

By substitution of $d = d_0 + \rho d_1$

In $$d^t \nabla f(x) \le \alpha d_0^t \nabla f(x), \tag{13}$$

we get
$$\rho \le (\alpha - 1) d_0^1 \nabla f(x) / d_1^t \nabla f(x), \tag{14}$$

in the case when $d_1^t \nabla f(x) > 0.$

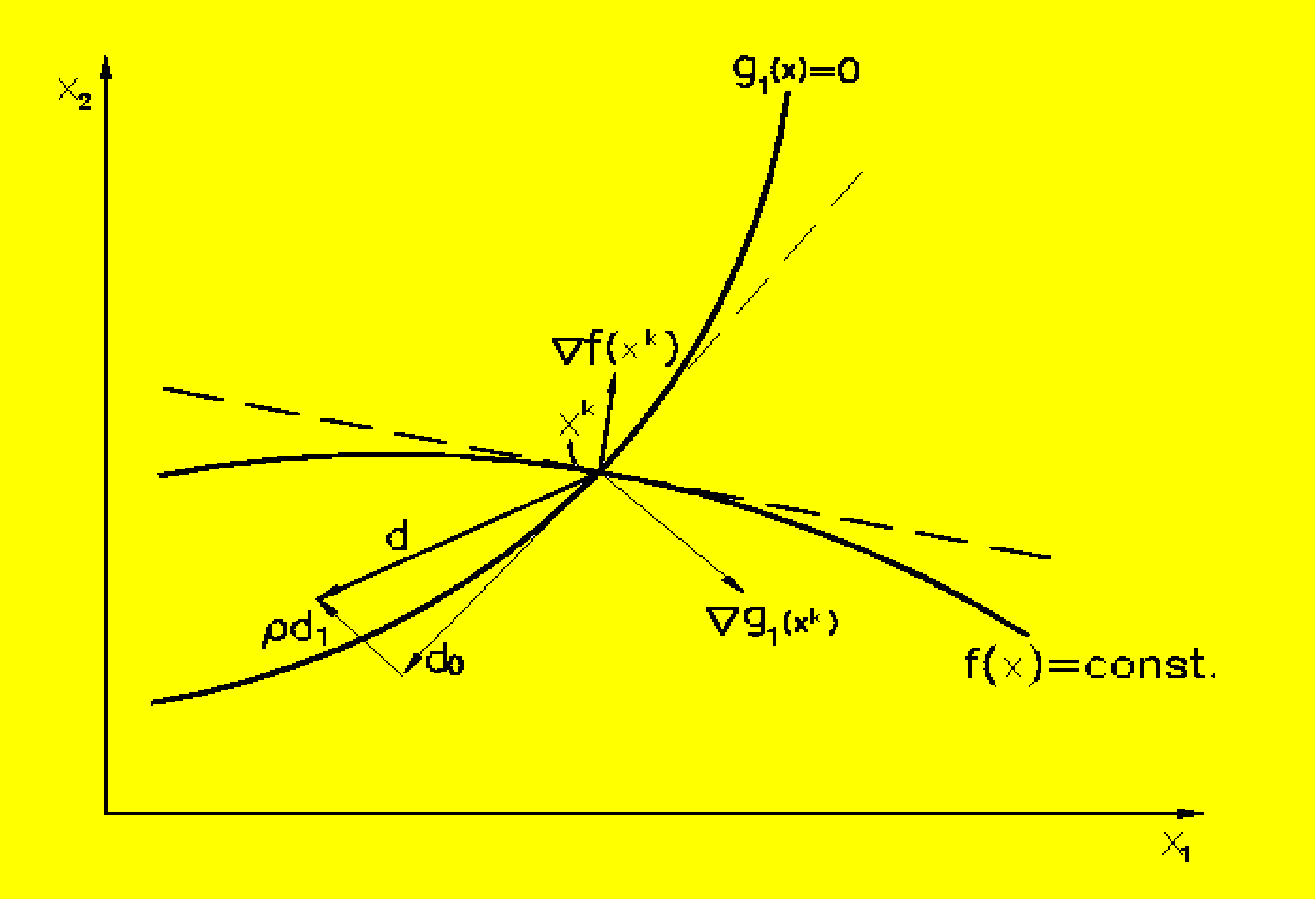
Otherwise, any $\rho > 0$ holds.

Basic Ideas

- To find a new primal point, an inaccurate line search is done in the direction of d .
- We look for a new interior point with a satisfactory decrease of the objective function.
- Different updating rules can be employed to define a new positive λ.

FDIPA
Feasible Direction Interior Point Algorithm

Search Direction



FAIPA

Feasible Arc Interior Point Algorithm

- Several practical applications and test problems were solved very efficiently with FDIPA.
- However for some problems with highly nonlinear constraints the unitary step length is not obtained and the rate of convergence is worst than superlinear.
- This effect is similar to the Maratos' effect and occurs when the feasible direction supports a too short feasible segment.
- **The Feasible Arc technique avoids this effect.**

FAIPA
Feasible Arc Interior Point Algorithm

The basic idea is to adjust better the constraints.

We compute the search direction d of **FDIPA,** and:

$$\tilde{\omega}^i = g_i(x+d) - g_i(x) - \nabla g_i(x) d; \qquad (15)$$

Then: $$\tilde{\omega}^i \approx \frac{1}{2} d^t \nabla^2 g_i(x) d;$$

is a 2nd order approximation of the constrains along d.

FAIPA

Feasible Arc Interior Point Algorithm

To obtain a feasible arc, we:

i) Compute the search direction d of FDIPA:

ii) Solve:

$$B\tilde{d} + \nabla g(x)\tilde{\lambda} = 0;$$
$$\lambda_i \nabla g^t(x)\tilde{d} + g_i(x)\tilde{\lambda}_i = -\lambda_i \tilde{\omega}^i, \qquad i = 1, \ldots, m. \qquad (16)$$

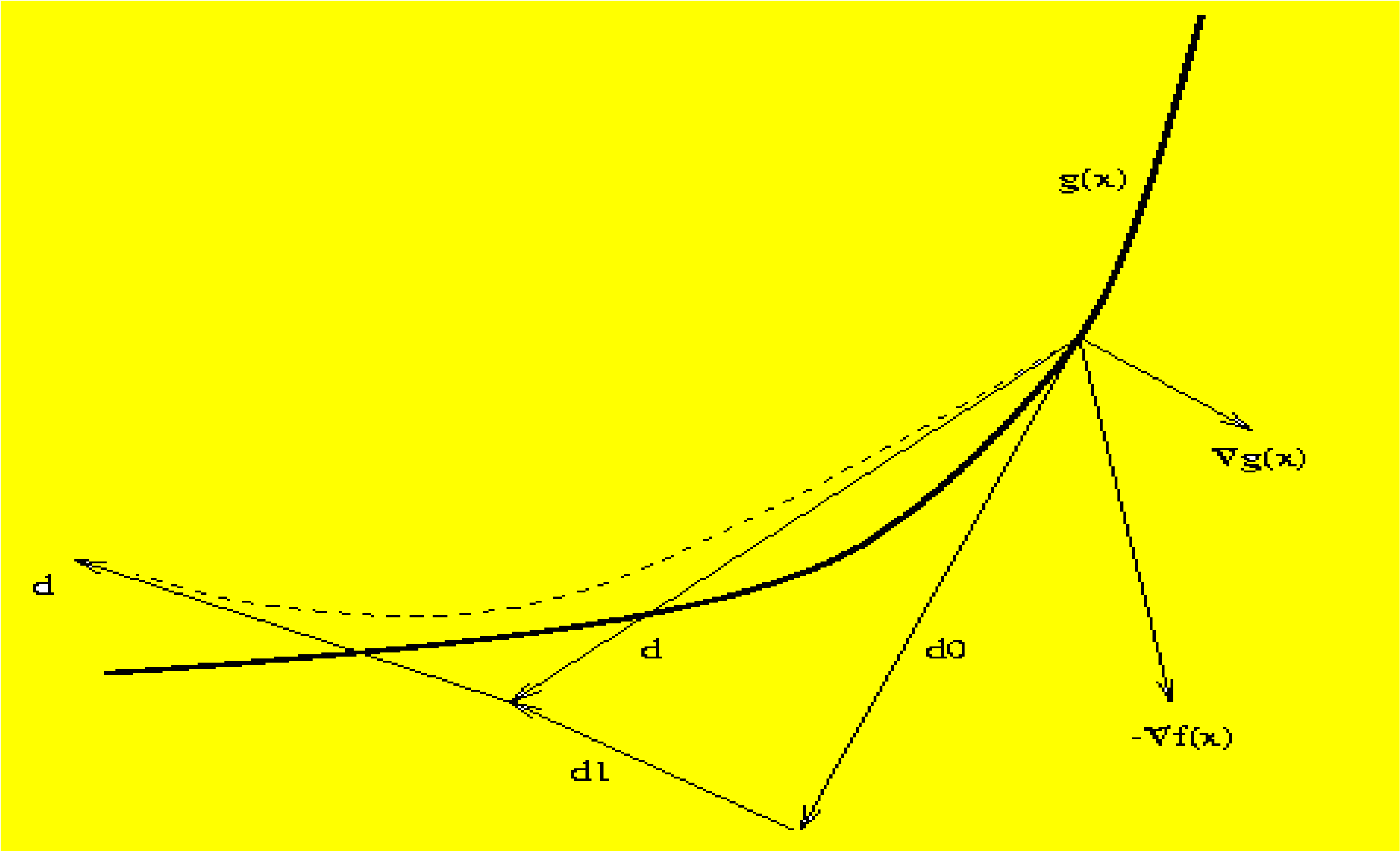
iii) Define the feasible arc as:

$$x^{k+1} = x^k + td + t^2\tilde{d} \qquad (17)$$

FAIPA

Feasible Arc Interior Point Algorithm

Feasible Descent Arc:



FAIPA

Feasible Arc Interior Point Algorithm

- When there are inequality constraints only, we solve:

  - The Primal-Dual System

$$\begin{bmatrix} B & \nabla g(x) \\ \Lambda \nabla g(x) & G(x) \end{bmatrix} \begin{bmatrix} d_0 & d_1 & \tilde{d} \\ \lambda_0 & \lambda_1 & \tilde{\lambda} \end{bmatrix} = - \begin{bmatrix} \nabla f(x) & 0 & 0 \\ 0 & -\lambda & -\lambda \tilde{w}^I \end{bmatrix} \quad (18)$$

  - Or the Dual System

$$\left[\nabla g^t(x) B^{-1} \nabla g(x) - \Lambda^{-1} G(x) \right] \left[\lambda_0 \quad \lambda_1 \quad \tilde{\lambda} \right] = \left[-\nabla g^t(x) B^{-1} \nabla f(x) \right] \quad (19)$$

which is symmetric and positive definite

FAIPA

Feasible Arc Interior Point Algorithm

To solve the Dual System:

$$\left[\nabla g^{t}(x)B^{-1}\nabla g(x)-\Lambda^{-1}G(x)\right]\left[\lambda_0 \quad \lambda_1 \quad \tilde{\lambda}\right]=\left[-\nabla g^{t}(x)B^{-1}\nabla f(x)\right]$$

We have to compute and store:

- The constraints derivative matrix $\nabla g(x)$
- The quasi-Newton matrix B
- The dual system matrix $\nabla g^{t}(x)B^{-1}\nabla g(x)-\Lambda^{-1}G(x)$

In structural optimization, these matrices are generally dense.

Numerical Techniques

To solve the Dual System with low memory requirements we use:

- Limited-Memory Quasi-Newton Method

  (storing s few vectors to represent the quasi-Newton matrix)

- Gradient Conjugate Method

  (avoiding system matrix storage)

- Product of the constraint gradient matrix times a vector

  (avoiding the storage of constraint gradient matrix)

BFGS QUASI-NEWTON UPDATING RULE

- Let be:

$$l(x,\lambda,\mu) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{i=1}^{p} \mu_i h_i(x) \qquad (16)$$

$$s_k = x_{k+1} - x_k$$
$$y_k = \nabla l(x_{k+1},\lambda_k,\mu_k) - \nabla l(x_k,\lambda_k,\mu_k) \qquad (17)$$

- BFGS updating rule:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^t B_k}{s_k^t B_k s_k} + \frac{y_k y_k^t}{y_k^t s_k} \qquad (18)$$

LIMITED-MEMORY QUASI-NEWTON METHOD

The BFGS updating rule can be written as follows:

$$B_k = B_0 - \begin{bmatrix} B_0 S_k & Y_k \end{bmatrix} \begin{bmatrix} S_k^t B_0 S_k & L_k \\ L_k^t & -D_k \end{bmatrix}^{-1} \begin{bmatrix} S_k^t B_0 \\ Y_k^t \end{bmatrix} \quad (18)$$

Where:

$S_k = [s_0, ..., s_k]$ & $Y_k = [y_0, ..., y_k]$ are (*n x k)* matrices

$$(L_k)_{ij} = \begin{cases} s_{i-1}^t y_{j-1} & \text{for } i > j \\ 0 \text{ for } i \le j \end{cases}$$

is a triangular $(k \text{ x } k)$ matrix

$$D_k = \text{diag}[s_0^t y_0, ..., s_{k-1}^t y_{k-1}]$$

is a diagonal $(k \text{ x } k)$ matrix

LIMITED-MEMORY QUASI-NEWTON METHOD

Instead of considering the k pairs of vectors $\{s, y\}$, B is updated taking only the last q pairs. Assuming that $B_{k-q}=I$:

$$B_k = I - \begin{bmatrix} S_k & Y_k \end{bmatrix} \begin{bmatrix} S_k^t S_k & L_k \\ L_k^t & -D_k \end{bmatrix}^{-1} \begin{bmatrix} \varepsilon_k S_k^t \\ Y_k^t \end{bmatrix} \qquad (19)$$

Now:

$S_k = [s_{k-q}, \ldots, s_{k-1}]$ & $Y_k = [y_{k-q}, \ldots, y_{k-1}]$ are $(n \text{ x } q)$ matrices,

$$(L_k)_{ij} = \begin{cases} s_{k-q-1+i}^t \, y_{k-q-1+j} \text{ for } i > j \\ 0 \text{ for } i \le j \end{cases}$$

is a triangular $(q \text{ x } q)$ matrix

$D_k = \text{diag}\,[s_{k-q}^t y_{k-q}, \ldots, s_{k-1}^t y_{k-1}]$ is a diagonal $(q \text{ x } q)$ matrix.

LIMITED-MEMORY QUASI-NEWTON METHOD

In practice, we always take *q < 10.*

For $v \in R^n$ *given, it is very easy to compute*

$$B_k v = v - \begin{bmatrix} S_k & Y_k \end{bmatrix} \begin{bmatrix} S_k^t S_k & L_k \\ L_k^t & -D_k \end{bmatrix}^{-1} \begin{bmatrix} \varepsilon_k S_k^t \\ Y_k^t \end{bmatrix} v \qquad (20)$$

without need of computing and storing the quasi-Newton matrix.

A similar expression is also obtained for $\boldsymbol{H} = \boldsymbol{B}^{-1}$

SOLVING LINEAR SYSTEMS OF EQUATIONS

Consider the linear system of equations:

$$Ax = b$$

The Conjugate Gradient algorithm is a well known iterative method to solve linear systems with a positive definite coefficient matrix.

CONJUGATE GRADIENT METHOD PRECONDITIONED BY THE LIMITED-MEMORY MATRIX

The proposed algorithm solves this problem:

$$HL^{-1}AL^{-t}y = HL^{-1}b \qquad (21)$$

where: $x = L^{-t}y$

- *L* is a triangular preconditioning matrix.

- H is the Limited-Memory Quasi-Newton matrix, to solve the unconstrained minimization problem:

$$\min \frac{1}{2} y^t L^{-1} A L^{-t} y - y^t L^{-1} b \qquad (22)$$

CONJUGATE GRADIENT ALGORITHM PRECONDITIONED BY THE LIMITED-MEMORY MATRIX

Given $x_{0,}$

Compute:

$$r_0 = L^{-1}(b - Ax_0), \quad z_0 = Hr_0, \quad p_0 = L^{-t} z_0$$

For i = 1 until convergence do:

$$\alpha_i = r_i^t z_i \big/ (Ap_i)^t p_i$$

$$x_{i+1} = x_i + \alpha_i p_i$$

$$r_{i+1} = r_i - \alpha_i L^{-1} A p_i$$

$$z_{i+1} = H r_{i+1}$$

$$\beta_i = r_{i+1}^t z_{i+1} \big/ r_i^t z_i$$

$$p_{i+1} = L^{-1} z_{i+1} + \beta_i p_i$$

end.

Note that:

-The system matrix ***A*** only appears multiplied by vectors

- When multiplying ***H*** by a vector, limited memory formulation is employed

Solving the Dual System of FAIPA

To solve the Dual System at each iteration of the PCG method, we must compute:

$$\left(\nabla g^t(x) B^{-1} \nabla g(x) - \Lambda^{-1} G(x)\right) z \; ; \quad z \in R^m$$

Where:

$v = \nabla g(x)\, z$ is the gradient of an auxiliary constraint $g^t(x)\, z$

$w = B^{-1} v$ is obtained with limited memory formulation

$\nabla g^t(x)\, w$ is a directional derivative of the constraints

Instead of storing the whole derivative matrix,

we just compute and store the products $\nabla g(x)\, z$

and $\nabla g^t(x)\, w$**.**

Computing $\nabla g(x)\, z$ and $\nabla g^t(x)\, w$

$\nabla g(x)\, z$: Can be computed with the adjoint variables method.

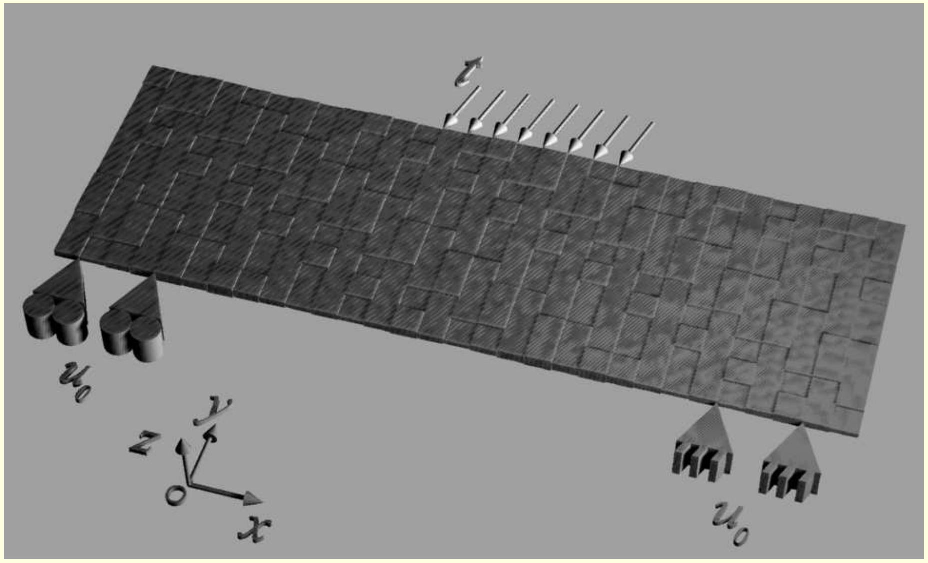
For linear elastic structures, one system with the stiffness matrix must be solved.

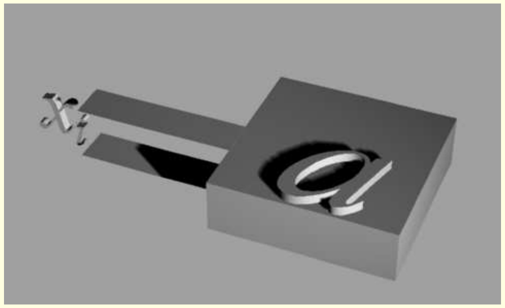
$\nabla g^t(x)\, w$: Directional derivatives of displacements in linear elastic structures follows from directional derivation of the equilibrium equation.

THEN: two linear systems with the stiffness matrix are solved at each iteration of the CG.

A structural optimization example:

- The optimization problem we are dealing with is the structural volume minimization with Von-Misses stress constraints.
- The structures are rectangular plates submitted to in-plane distributed loadings and supports.
- Structure responses are computed by FEA simulations using a mesh of quadrilateral bilinear plane stress elements.
- The thickness of an element is a design variable.

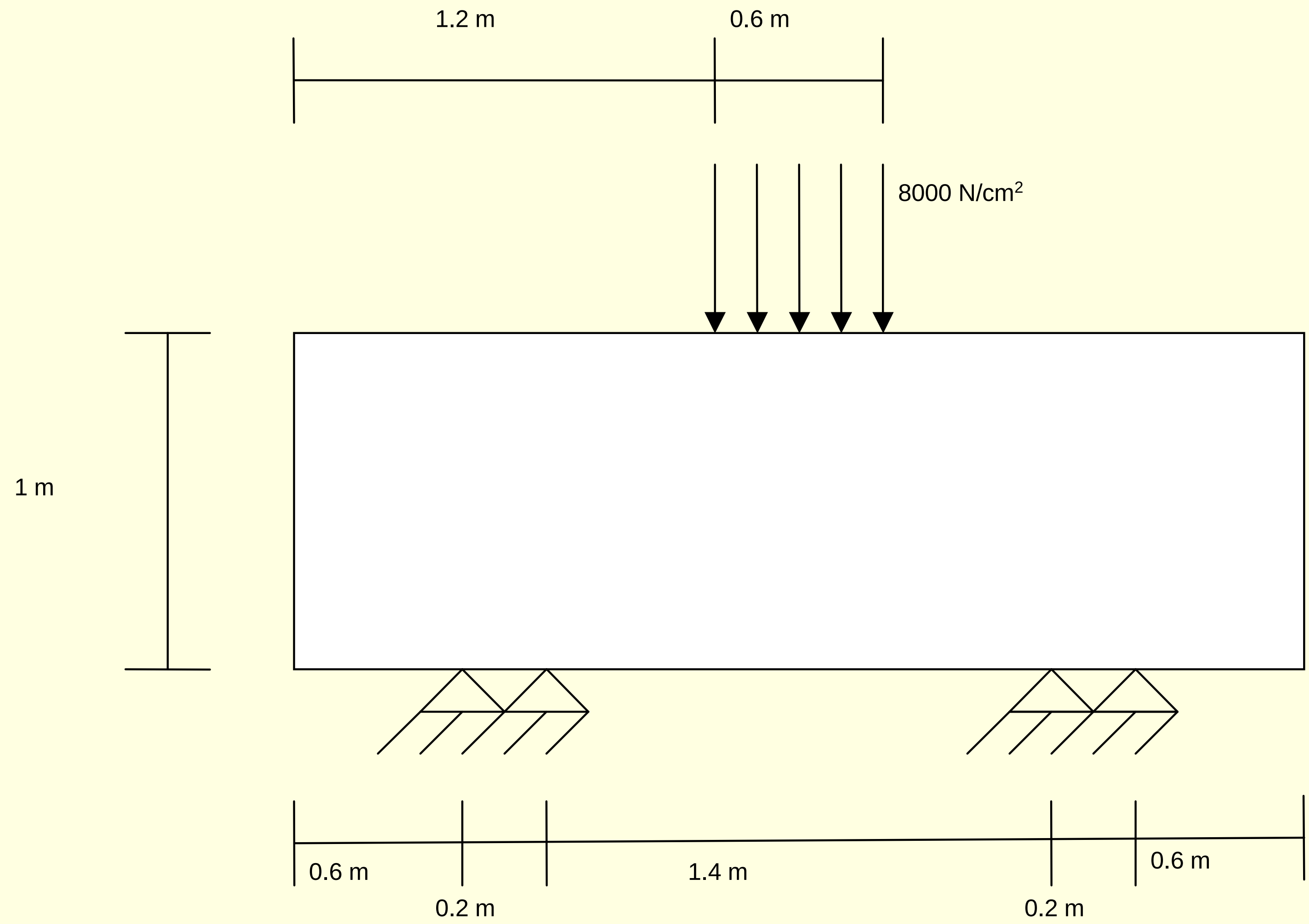




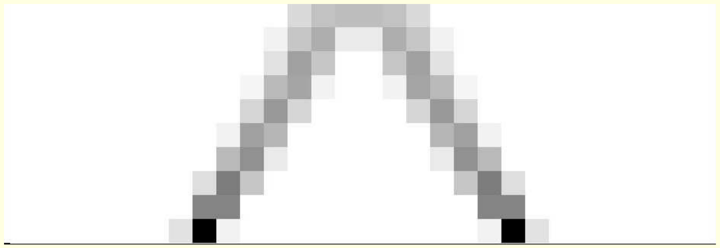
A structural optimization example:

- Structural Optimization Problem
  - 2D plates submitted to in-plane distributed loadings.
  - Quadrilateral bilinear plane stress elements with Young's module of 210GPa and Poisson's coefficient of 0.3 are assumed for each element.
  - Design Variables: Thickness of each element
  - Objective Function: Structural volume
  - Constraints, for each element:
    - Von-Mises stress less than σ_{adm}=250MPa.
    - Thickness between 0.1 and 1.0 centimeter.
  - Number of LM pairs:
    - q_A=8 (preconditioner of dual system, *A*)
    - q_B=10 (quasi-Newton matrix, *B*)
  - Resources:
    - Memory: 1 Gb
    - Processor: AMD Athlon with 64 bits and 1.8 GHz

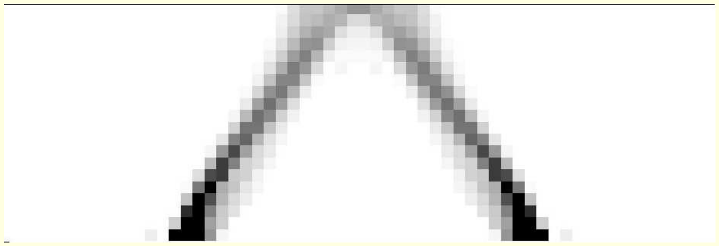
Example 1



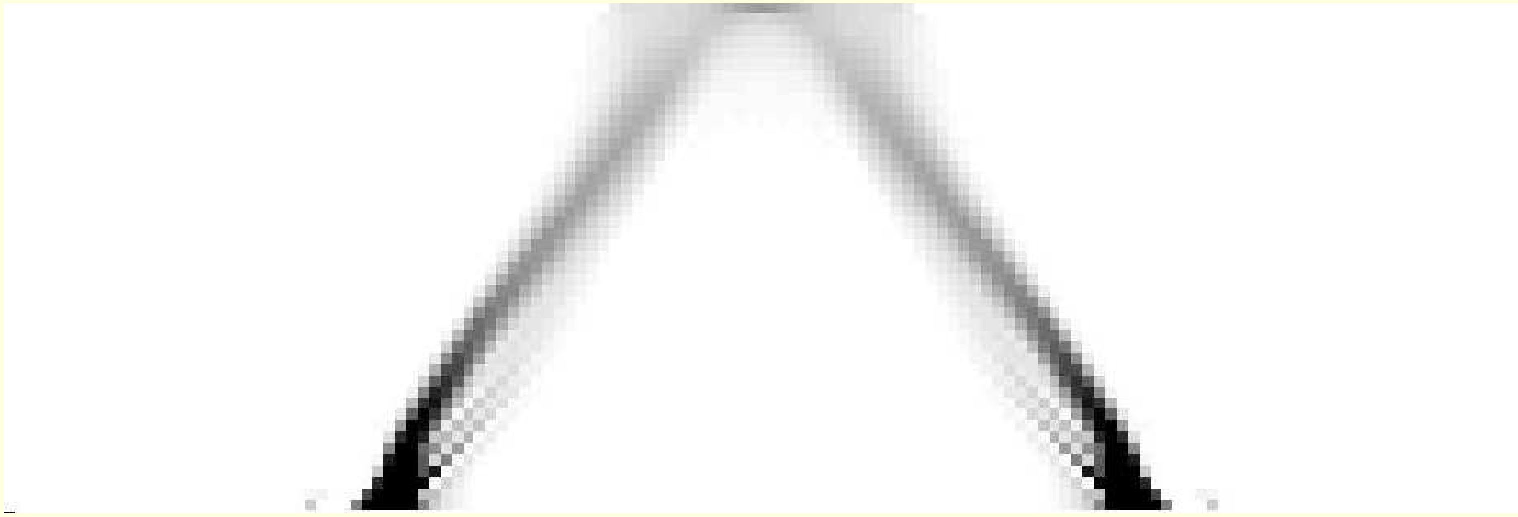
Example 1



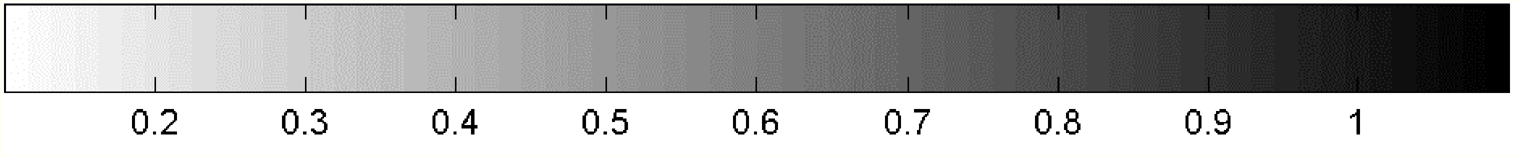
300 elements



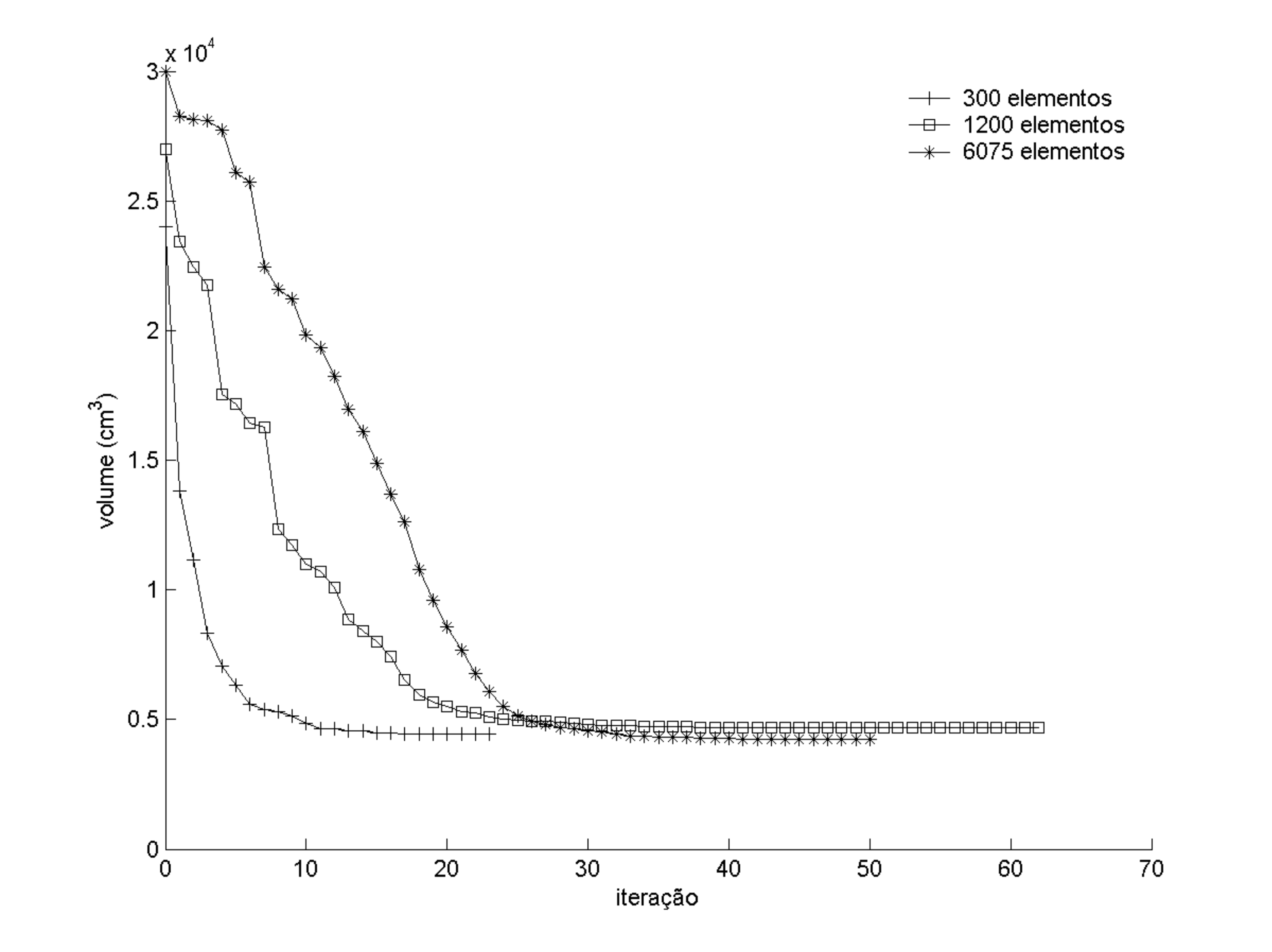
1200 elements



6075 elements

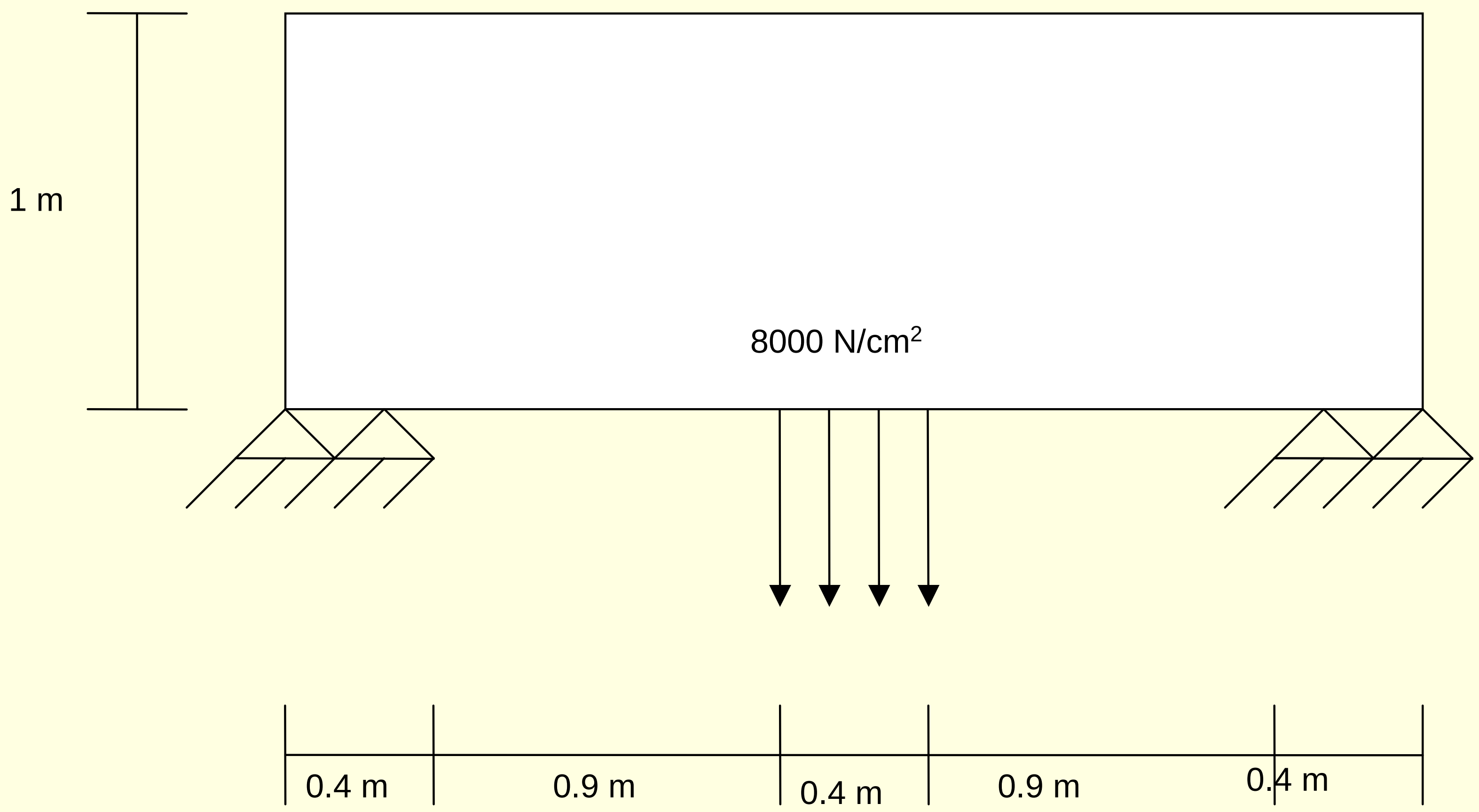


Example 1

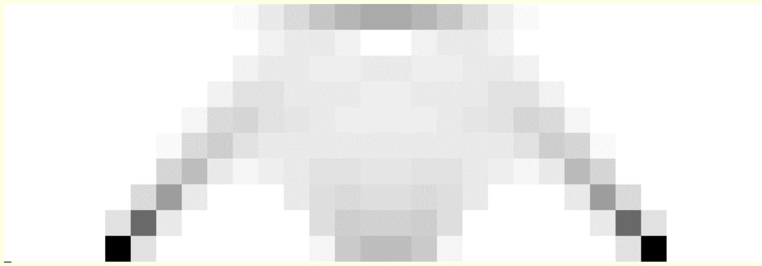


Iteration history

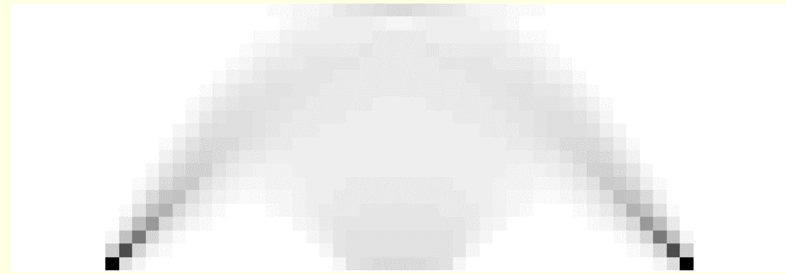
Example 2



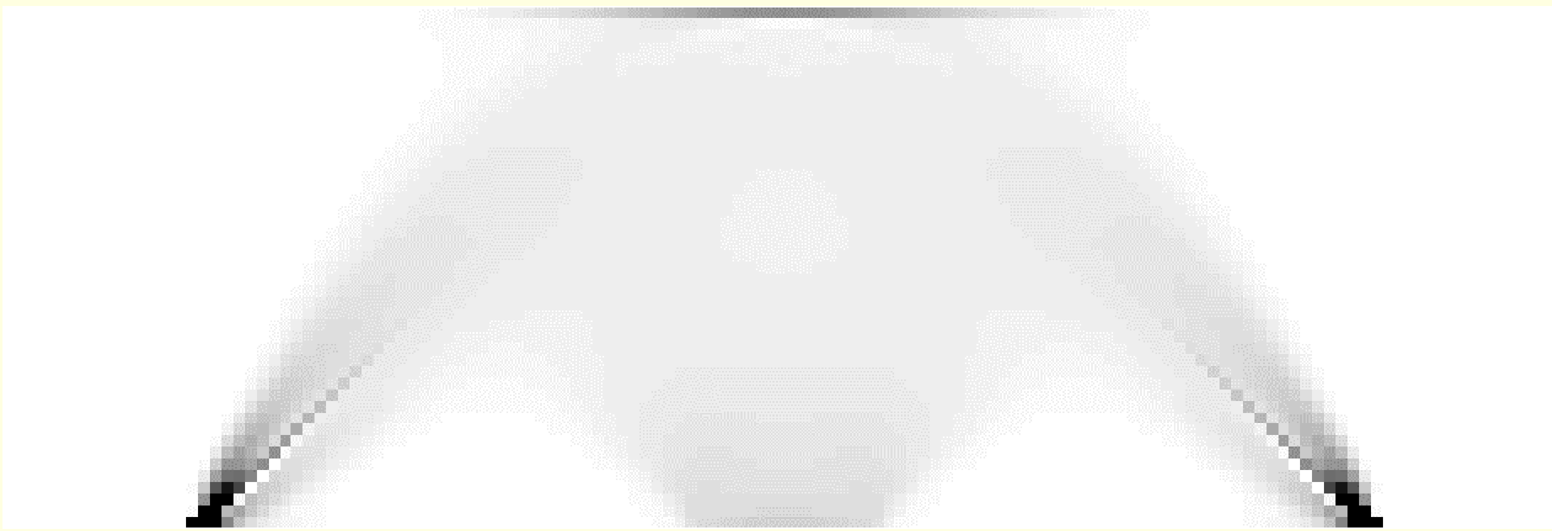
Example 2



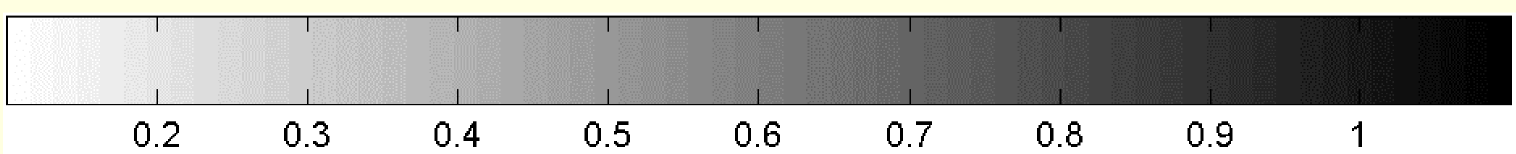
300 elements



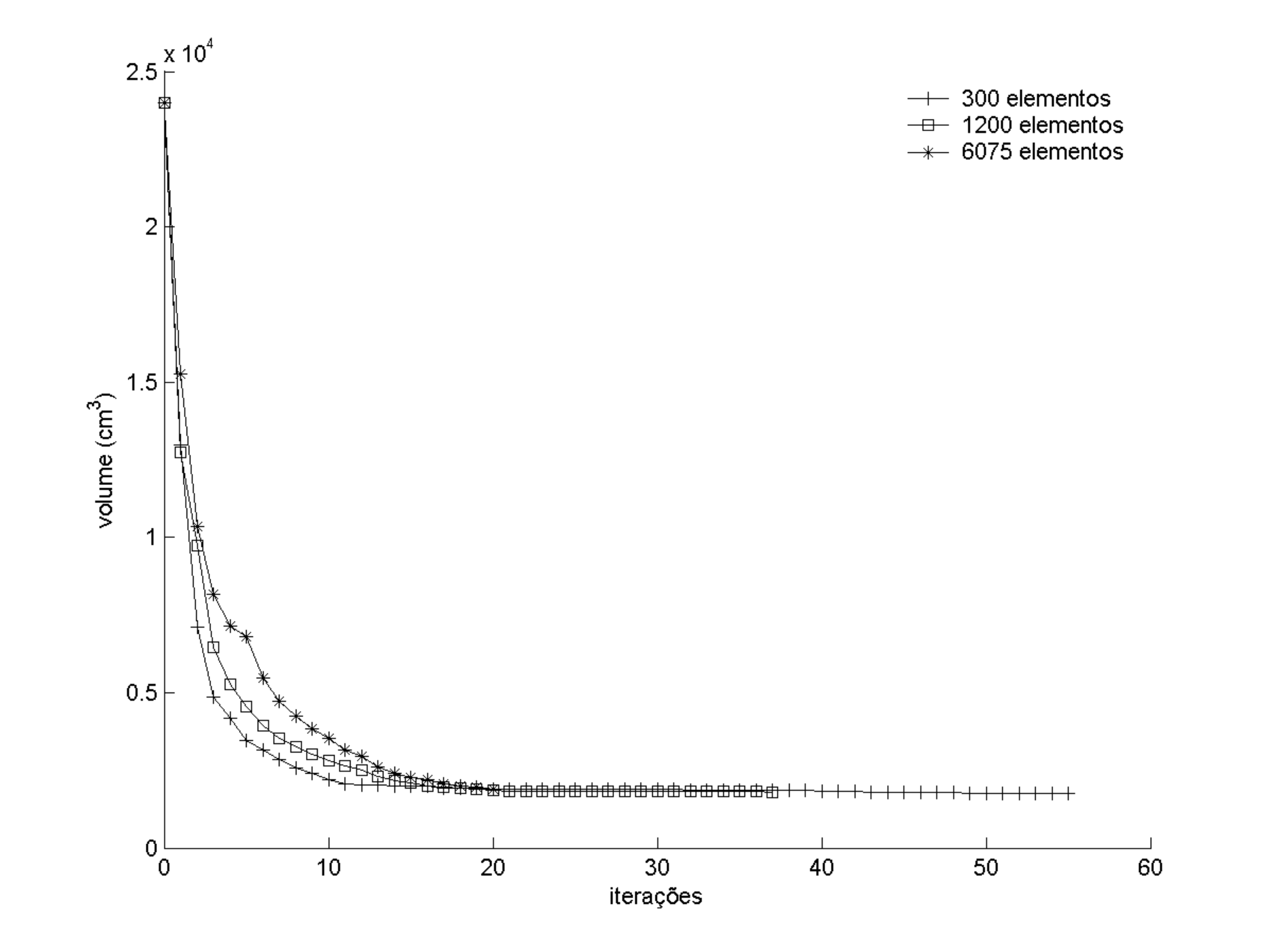
1200 elements



6075 elements



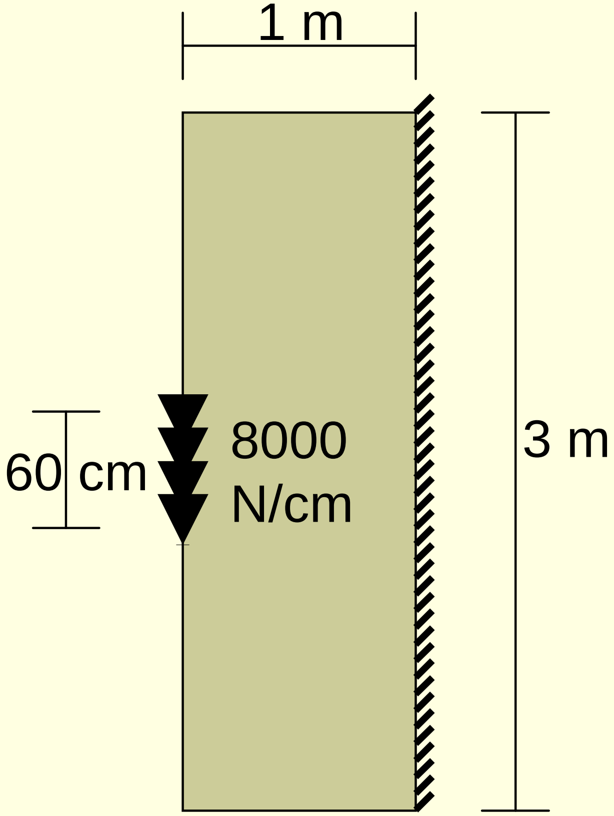
Example 2



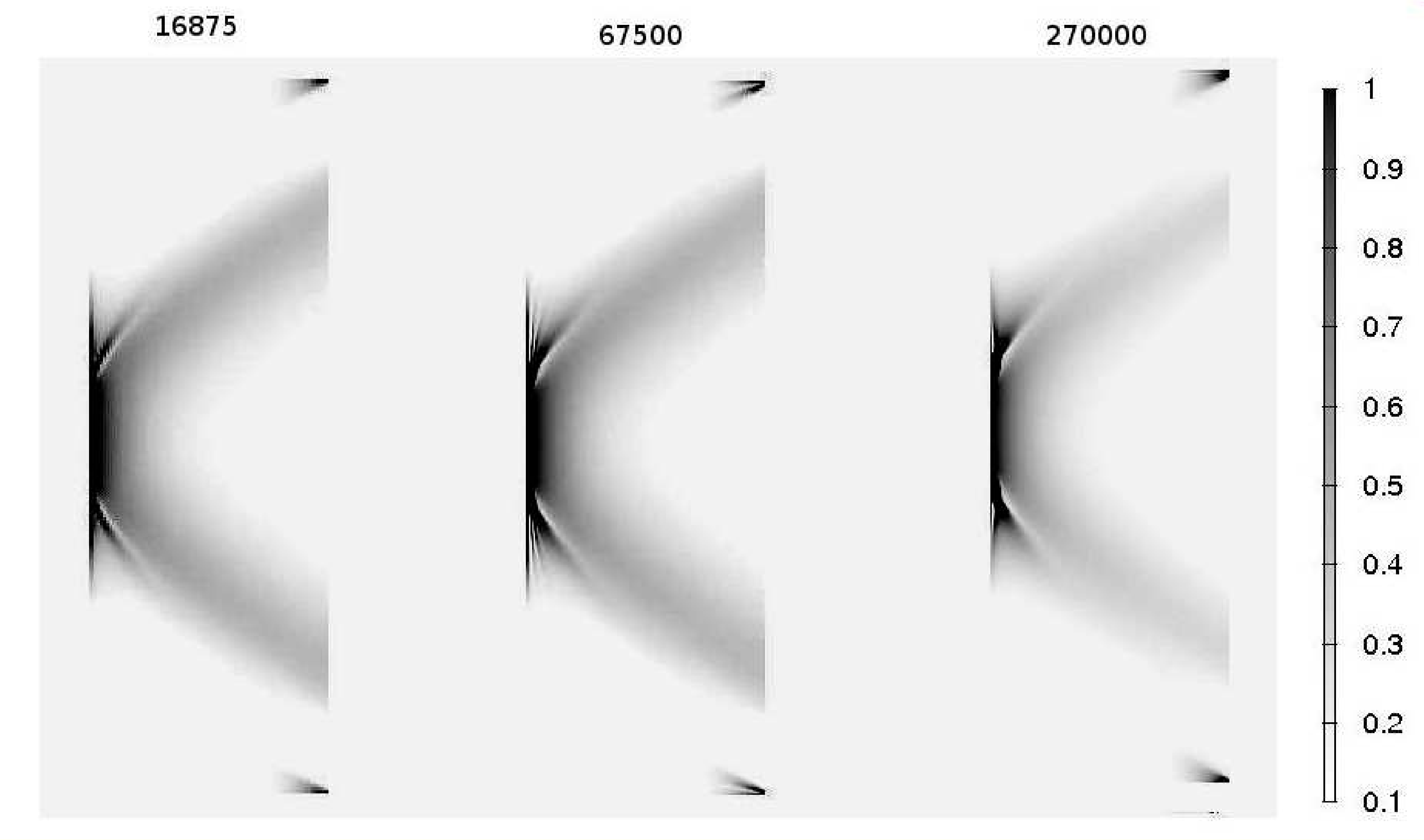
Iteration history

Example 3

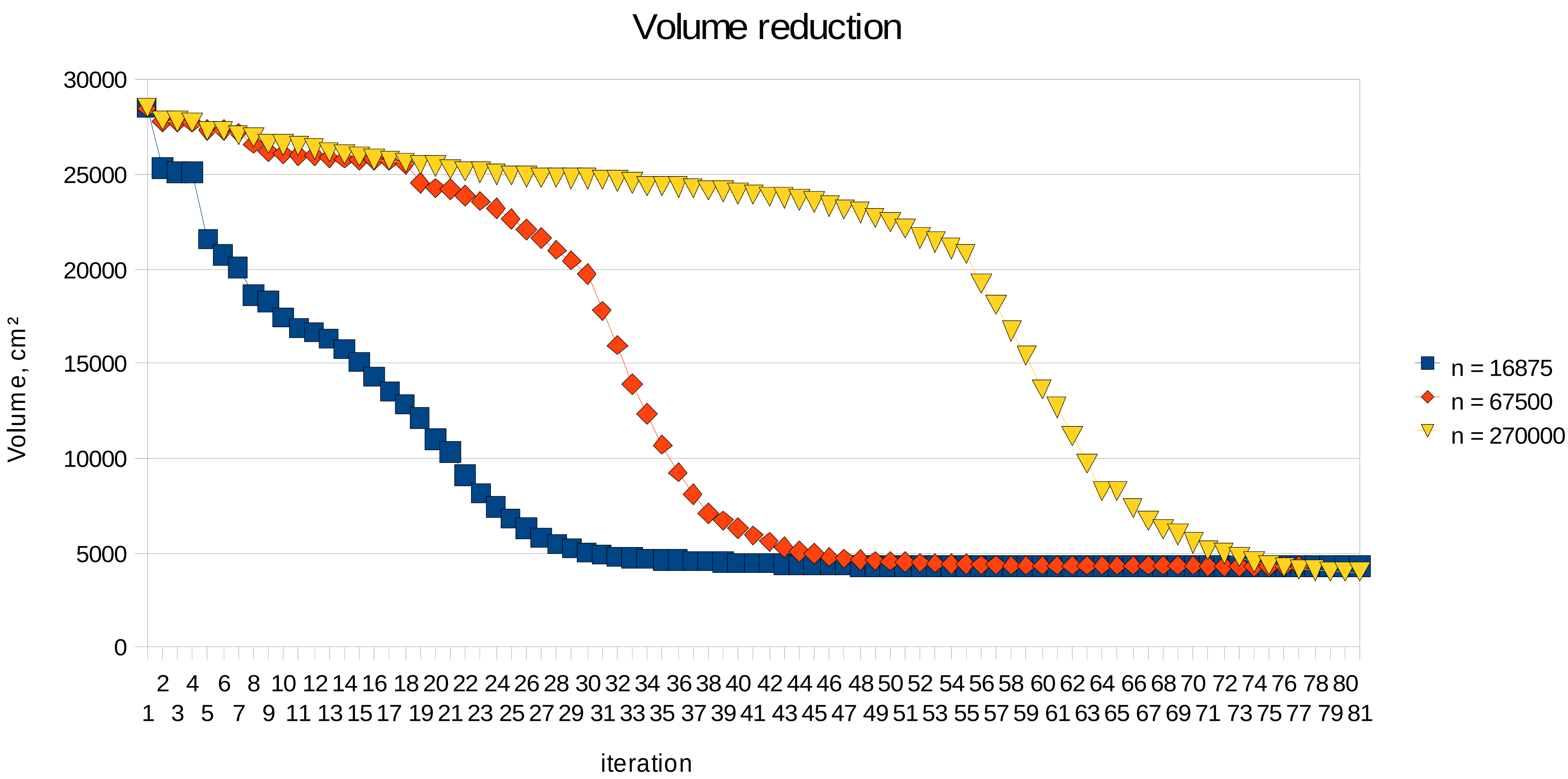
- 16875, 67500 and 270000 elements
- Initial thickness 0.95 cm
- Lower bound of thickness t_{min}=1 mm
- Upper bound of thickness t_{max}=1 cm
- all elements with isotropic material
  - Young module: 210 GPa
  - Poisson: 0.3
- Stress constraint
  - Mises stress in center of element
  - less than $2.5x10^4$ Pa



Example 3

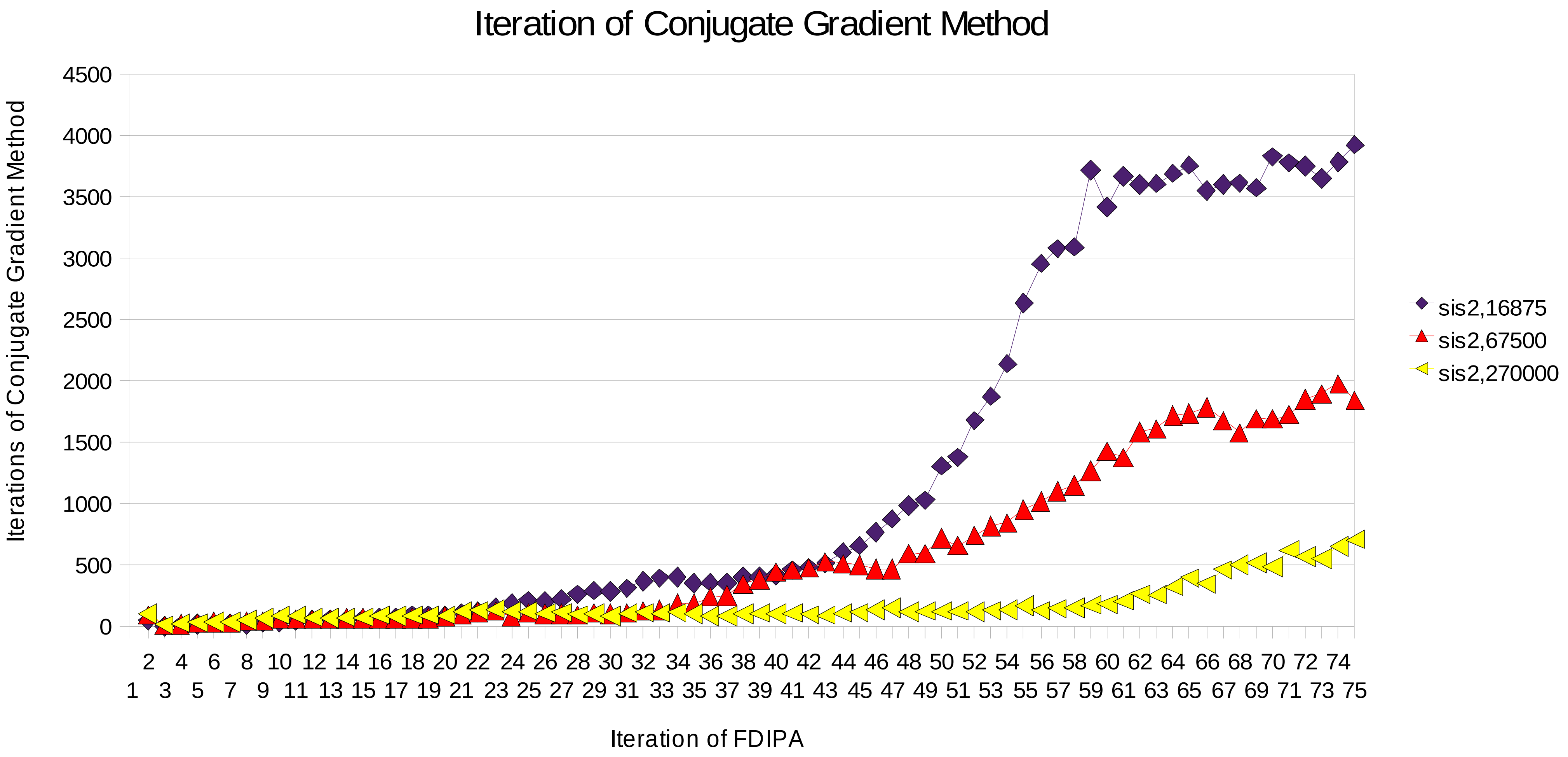


Example 3



Iterations history

Example 3



Iteration of Conjugate Gradient Method
Iterations of Conjugate Gradient Method
Iteration of FDIPA
0
500
1000
1500
2000
2500
3000
3500
4000
4500
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
sis2,16875
sis2,67500
sis2,270000

Conclusions

- The present technique requires modest computational resources due to Limited Memory and Conjugate Gradient Methods:
  - Storage of quasi-Newton and pre-conditioner matrices are not needed. Those matrices are represented using a few LM pairs;
  - Constrained derivatives matrices are not stored. When CG iterations number is small, less derivatives are computed.
  - Dual system matrix is not allocated and sensitivity matrix is not computed, reducing the number of structural analysis per iteration of FAIPA.