Department of Applied Mathematics
Institute of Geonics AS CR
Ostrava, Czech Republic

Application of parallel computing to elasticity and thermo-elasticity problems

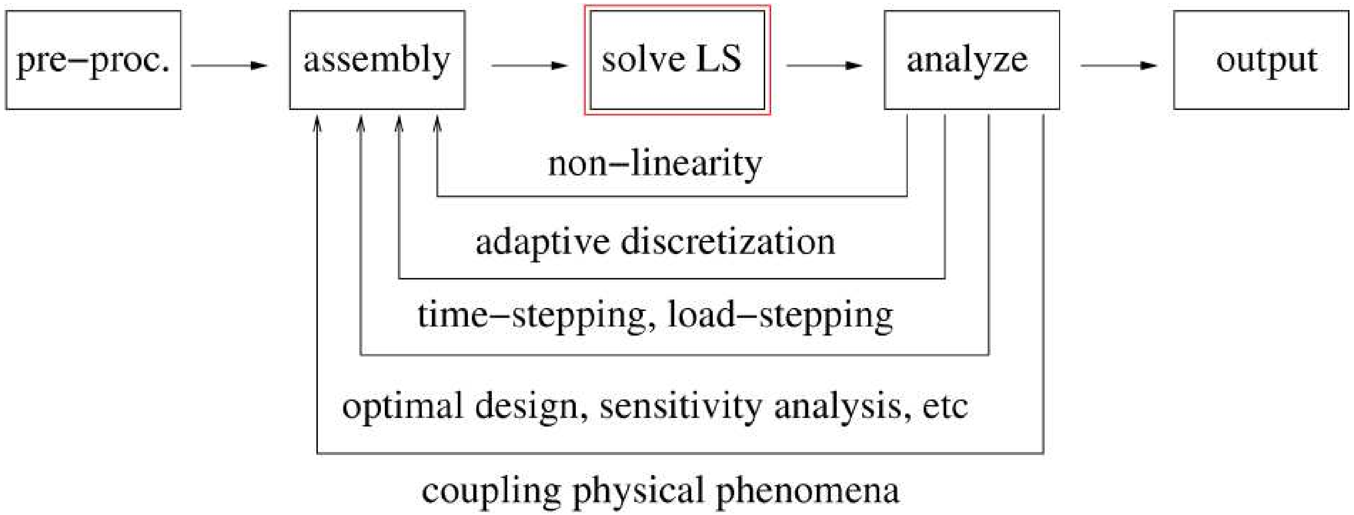
R. Blaheta, O. Jakl, K. Krečmer, J. Starý

E-mail: {blaheta|jakl|krecmer|stary}@ugn.cas.cz

Outline

- General introduction
- Iterative solvers based on the PCG method
- Space decomposition, DID and DD approach
- Parallel implementation of the solvers
- Computers: *Lomond*, *Thea*, *Natan*, *Termit*, *Ngorongoro*
- Benchmarks: *FOOT*, *DR (Dolní Rožínka)*, *KBS*
- Results and conclusion

Introduction



We shall consider the case, when the investigated physical phenomena are described by (*thermo-*) *elasticity problems* on domain $\Omega \subset R^3$. We shall also assume that the FE discretization of these boundary value problems leads to the solution of large-scale systems of the type

$$Au = f \; , \qquad u, f \in R^n \; ,$$

where A is a symmetric positive definite $n \times n$ matrix.

Iterative solvers based on the PCG method

$$r_0 = f - Au_0$$
$$v_0 = g_0 = G(r_0)$$
$$\sigma_0 = \langle g_0, r_0 \rangle$$

for $i = 0, 1, \ldots$ **until** $\|r_i\| \leq \varepsilon \, \|f\|$:

$$w = Av_i$$
$$\alpha = \sigma_i / \langle v_i, w \rangle$$
$$u_{i+1} = u_i + \alpha v_i$$
$$r_{i+1} = r_i - \alpha w$$
$$g_{i+1} = G(r_{i+1})$$
$$\sigma_{i+1} = \langle g_{i+1}, r_{i+1} \rangle$$
$$\beta = \sigma_{i+1} / \sigma_i$$
$$v_{i+1} = g_{i+1} + \beta v_i$$

end

In our computations: $\varepsilon = 10^{-4}$

GPCG[m] method:

$$v_{i+1} = g_{i+1}$$

for $k = min\{i+1, m\}, \ldots, 1$:

$$w = r_{i+2-k} - r_{i+1-k}$$
$$\beta^k_{i+1} = \langle g_{i+1}, w \rangle / \sigma_{i+1-k}$$
$$v_{i+1} = v_{i+1} + \beta^k_{i+1} v_{i+1-k}$$

end

Solvers:

- sequential
- *parallel*
  - data decomposition & parallel operations (vector updates, matrix by vector multiplication, . . .)
  - construction of the preconditioner & its efficient implementation

Space decomposition

$$V = V_1 + \cdots + V_p \,, \qquad V \sim R^n \,,\; V_k \sim R^{n_k}$$

$V_k, V_l \,\ldots$ do not need to be linearly independent

Restrictions:

$R_k : \; R^n \to R^{n_k}$

$$R_k \begin{bmatrix} v_1 \\ \vdots \\ v_p \end{bmatrix} = v_k$$

Prolongations:

$I_k : \; R^{n_k} \to R^n$

$$I_k v_k = \begin{bmatrix} 0 \\ v_k \\ 0 \end{bmatrix}$$

Decomposition of A:

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1p} \\ \vdots & \ddots & \vdots \\ A_{p1} & \cdots & A_{pp} \end{bmatrix}$$

Algorithm $g = G(r)$:

$g^0 = 0$

for $k = 1, \ldots, p$:

$$g^k = g^{k-1} + I_k A_{kk}^{-1} R_k z^k$$

end

Type of preconditioning:

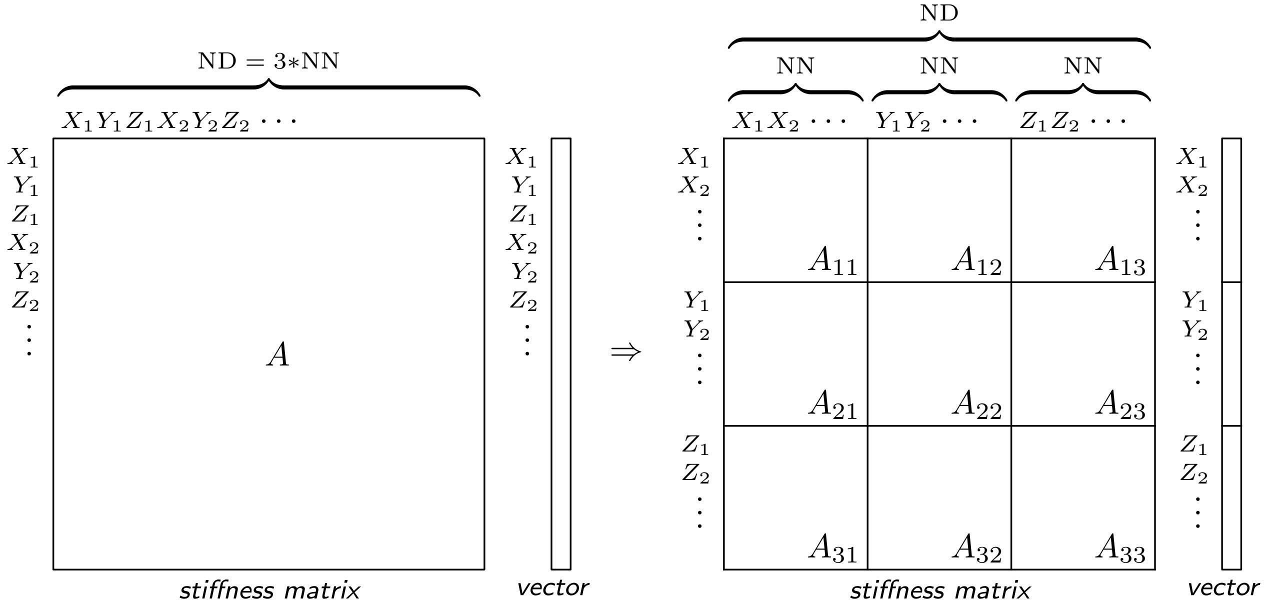
- *additive*: $z^k = r$
- multiplicative: $z^k = (r - A g^{k-1})$

Solution of A_{kk}^{-1}:

- exactly
- *inexactly*
  - by an incomplete factorization
  - by inner (PCG) iterations (up to a lower accuracy, $\varepsilon^* = 10^{-1}$)

Displacement decomposition

$$u = (u_1, \ldots, u_N)\,, \qquad u_i = (u_{ix}, u_{iy}, u_{iz})\,, \qquad V = V_x + V_y + V_z$$

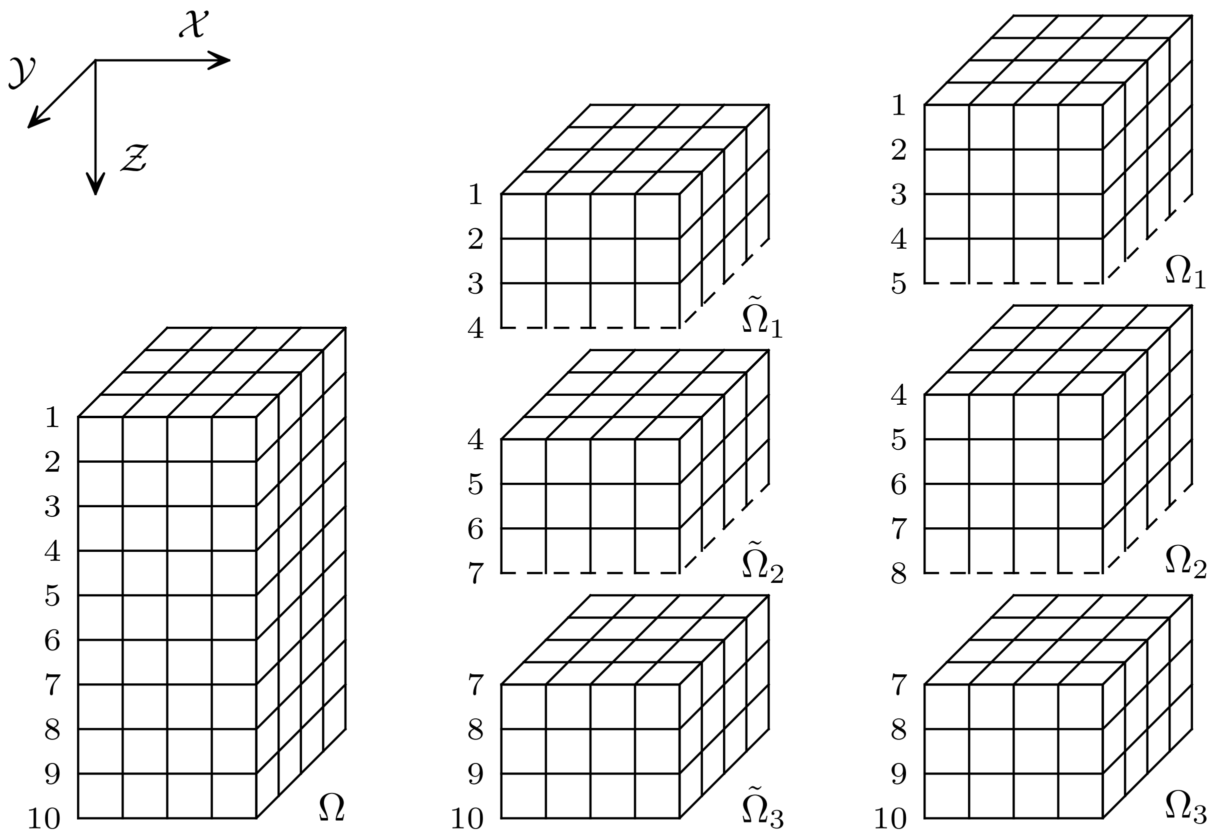


$$v = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \qquad A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \qquad B = \begin{bmatrix} A_{11} & & \\ & A_{22} & \\ & & A_{33} \end{bmatrix}$$

- splitting of computations among 3 concurrent processes
- blocks in B are completely independent - *no communication* of concurrent tasks *during the preconditioning* is needed

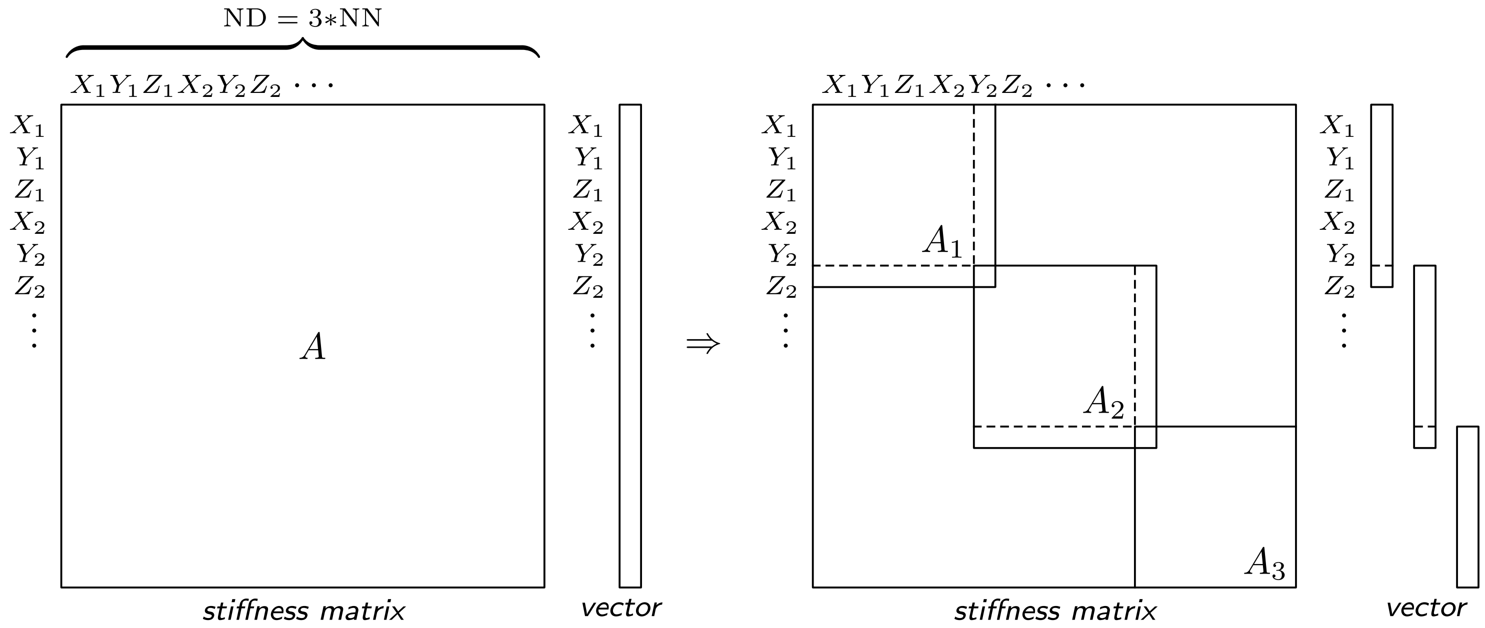
Domain decomposition (1)

$$\Omega = \bigcup_{i=1}^{p} \Omega_i \,, \qquad V = \sum_{i=1}^{p} V_i \,, \qquad V_i = \{v \in V : v = 0 \text{ in } \Omega \backslash \Omega_i\}$$



- DD in only dimension - each of concurrent tasks has to communicate with at most two others corresponding to neighbouring subdomains

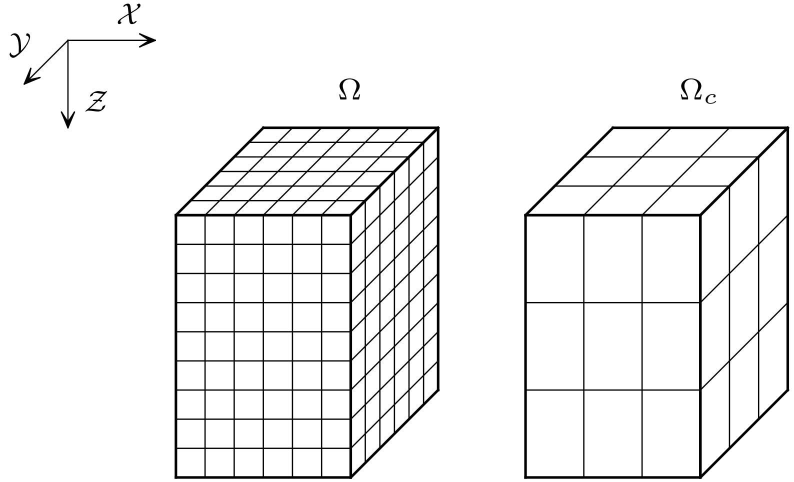
Domain decomposition (2)



nonoverlapping subdomains	overlapping subdomains
block Jacobi	*additive Schwarz*

$$g = G_1(r) = \sum_{i=1}^{p} I_i \tilde{A}_i^{-1} R_i r \qquad g = G_2(r) = \sum_{i=1}^{p} I_i A_i^{-1} R_i r$$

Two-level domain decomposition

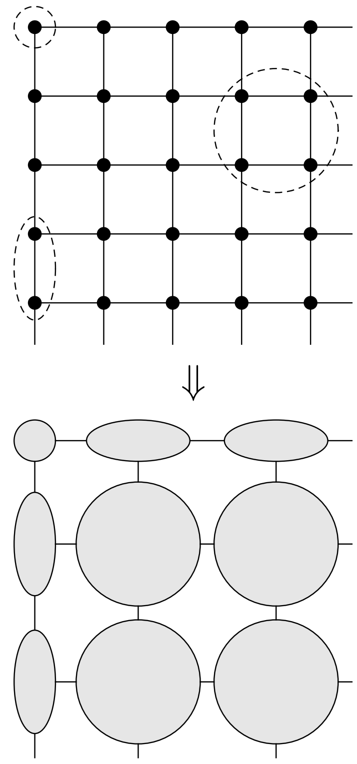


- $V = V_c + \sum_{i=1}^{p} V_i$
- need of A_c, R_c, I_c

- the construction of R_c, I_c depends on fact, if the coarse grid is nested or not
- *two-level additive Schwarz*

$$g = G_3(r) = G_2(r) + I_c A_c^{-1} R_c r$$

- if it is impossible to make the explicit coarse grid or too much time demanding to make or use R_c, I_c - *aggregations* allow the simple construction of all data



Example of regular aggregations

Parallel implementation

step	$k^{\mathbf{th}}$ worker process	communication	$l^{\mathbf{th}}$ worker process
	data in memory: $u_k, v_k, r_k, g_k, w_k \equiv w_{kk}, A_{kk}$ $w_{ki}, A_{ik}, i = 1, \ldots, m, i \neq k$		data in memory: $u_l, v_l, r_l, g_l, w_l \equiv w_{ll}, A_{ll}$ $w_{li}, A_{il}, i = 1, \ldots, m, i \neq l$
S_1	for $i = 1, \ldots, m$: $w_{ki} = A_{ik} v_k$		for $i = 1, \ldots, m$: $w_{li} = A_{il} v_l$
C_1		$= w_{kl} \Rightarrow$	
		$\Leftarrow w_{lk} =$	
S_2	$w_k = \sum_{i=1}^m w_{ik}$		$w_l = \sum_{i=1}^m w_{il}$
S_3	$s_k = \langle w_k, v_k \rangle$		$s_l = \langle w_l, v_l \rangle$
C_2		$\Leftarrow s = \sum_{i=1}^m s_i \Rightarrow$	
S_4	$\alpha = s_0 / s$		$\alpha = s_0 / s$
	$u_k = u_k + \alpha v_k$		$u_l = u_l + \alpha v_l$
	$r_k = r_k - \alpha w_k$		$r_l = r_l - \alpha w_l$
S_5, C_3	$g = G(r)$	$\Leftarrow r_k, r_l \Rightarrow$	$g = G(r)$
S_6	$s_k = \langle g_k, r_k \rangle$		$s_l = \langle g_l, r_l \rangle$
C_4		$\Leftarrow s_1 = \sum_{i=1}^m s_i \Rightarrow$	
S_7	$\beta = s_1 / s_0$		$\beta = s_1 / s_0$
	$s_0 = s_1$		$s_0 = s_1$
	$v_k = g_k + \beta v_k$		$v_l = g_l + \beta v_l$

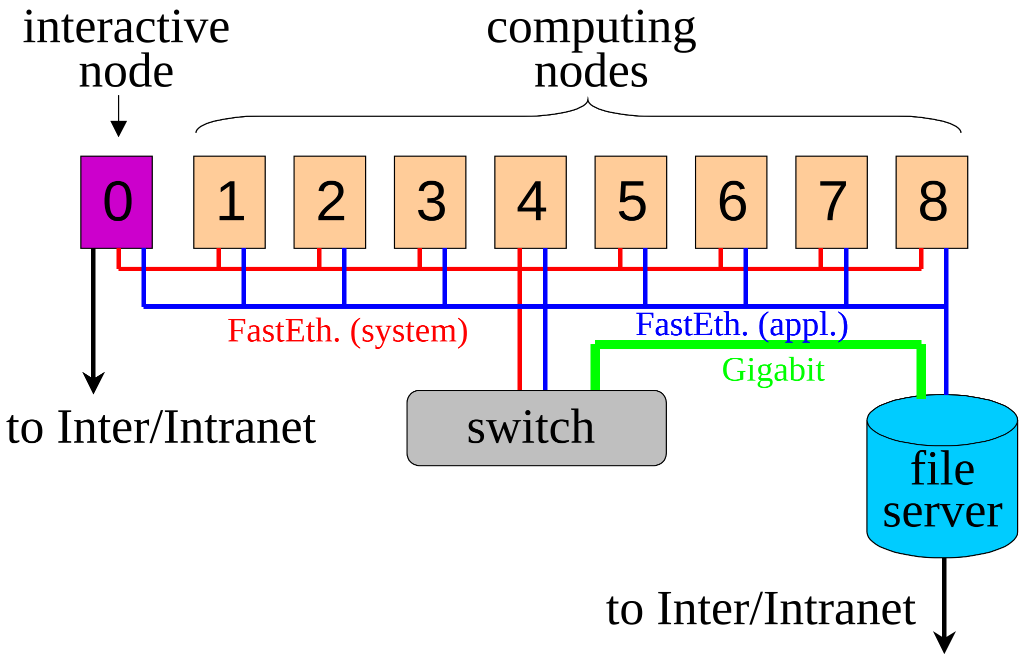
The worker-to-worker implementation of one iteration of the parallel CG algorithm with a Schwarz-type preconditioner

Used computers: Lomond

- Sun HPC cluster installed at EPCC, Edinburgh
- front-end: SMP system HPC 3500 with 8 UltraSPARC-II/400 and 8 GB of shared memory
- back-end: $2\times$ SMP system Sunfire 6800 with 24 UltraSPARC-III/750 and 48 GB of shared memory
- operating system: Solaris 2.8
- software: Sun Grid Engine, MPI (Sun)

Used computers: Thea

- Beowulf cluster installed at IGAS, Ostrava
- consists of 1 interactive and 8 computing nodes + file server



- each node is equipped by AMD Athlon/1400 processor, 1.5 GB of memory, 2 FastEthernet interfaces
- operating system: Debian Linux
- software: MPI (MPICH 1.2.6 and LAM 6.5.6), PVM (3.4.2), PETSc (2.1.3 and 2.2.0), PGI compilators

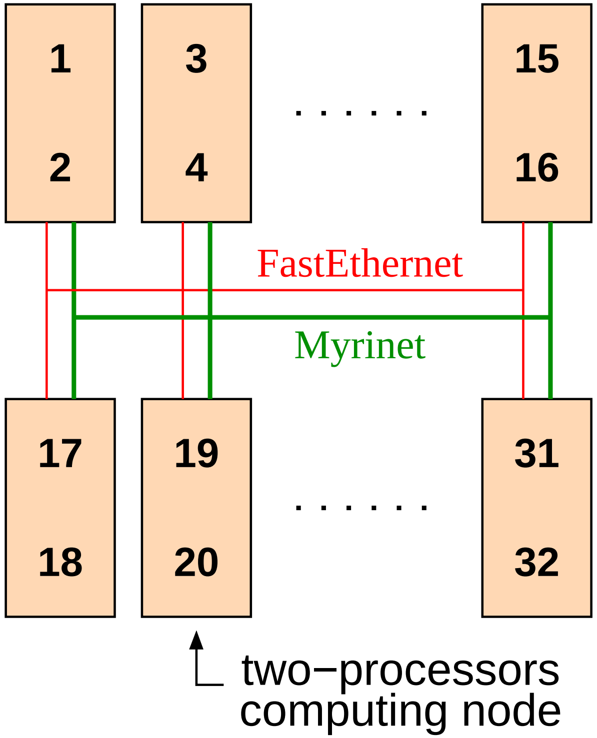
Used computers: Thea (photo)

Used computers: Natan

- IBM e-server xSeries 455 installed at IGAS, Ostrava
- SMP consists of 8 64-bit Intel Itanium 2/1300 processors, 16 GB of shared memory, 3 MB of L2 cache, RAID controller with 2× 72 GB disk storage
- operating system: SUSE Linux Enterprise Server 8
- software: MPI (MPICH 1.2.6), Intel compilators 8.1

Used computers: Termit

- Beowulf cluster installed at VŠB-TU, Ostrava
- consists of 16 computing nodes connected by Myrinet L9 2Gb/s and FastEthernet networks, one of nodes controls the others
- each node is equipped by 2 AMD Athlon MP/2600 processors and 3 GB of memory
- operating system: RedHat Linux
- software: MPI (MPICH 1.2.5 and LAM 6.5.6), PVM (3.4.4)



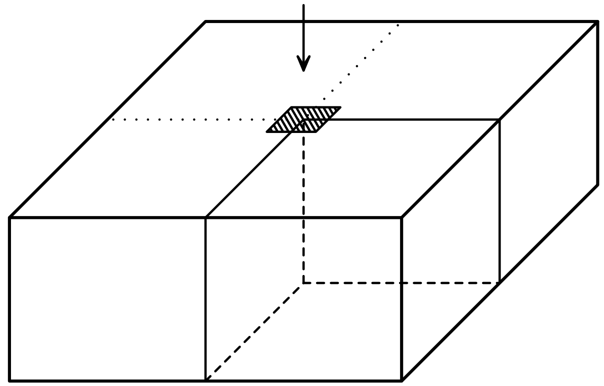
Used computers: Termit (photo)

Used computers: Simba (Ngorongoro)

- Sun Fire 15k (the theoretical top performance 86 GFlops/s) installed at UPPMAX, Uppsala
- SMP consists of 48 UltraSPARC III/900, 48 GB of shared memory, 8 MB of L2 cache, 12 $\times$ 18 GB drives + Fibrenetix RAID controller with 3.4 TB disk storage
- Sun Fireplane system interconnect has the peak data bandwith of 9.6 GB/s
- 4 parts: Simba (36 CPU's, 36 GB of memory) + Mbogo, Tembo, Duma (each has 4 CPU's, 4 GB of memory)
- operating system: Solaris 5.9
- software: Sun Grid Engine, MPI (Sun)

The *FOOT* benchmark

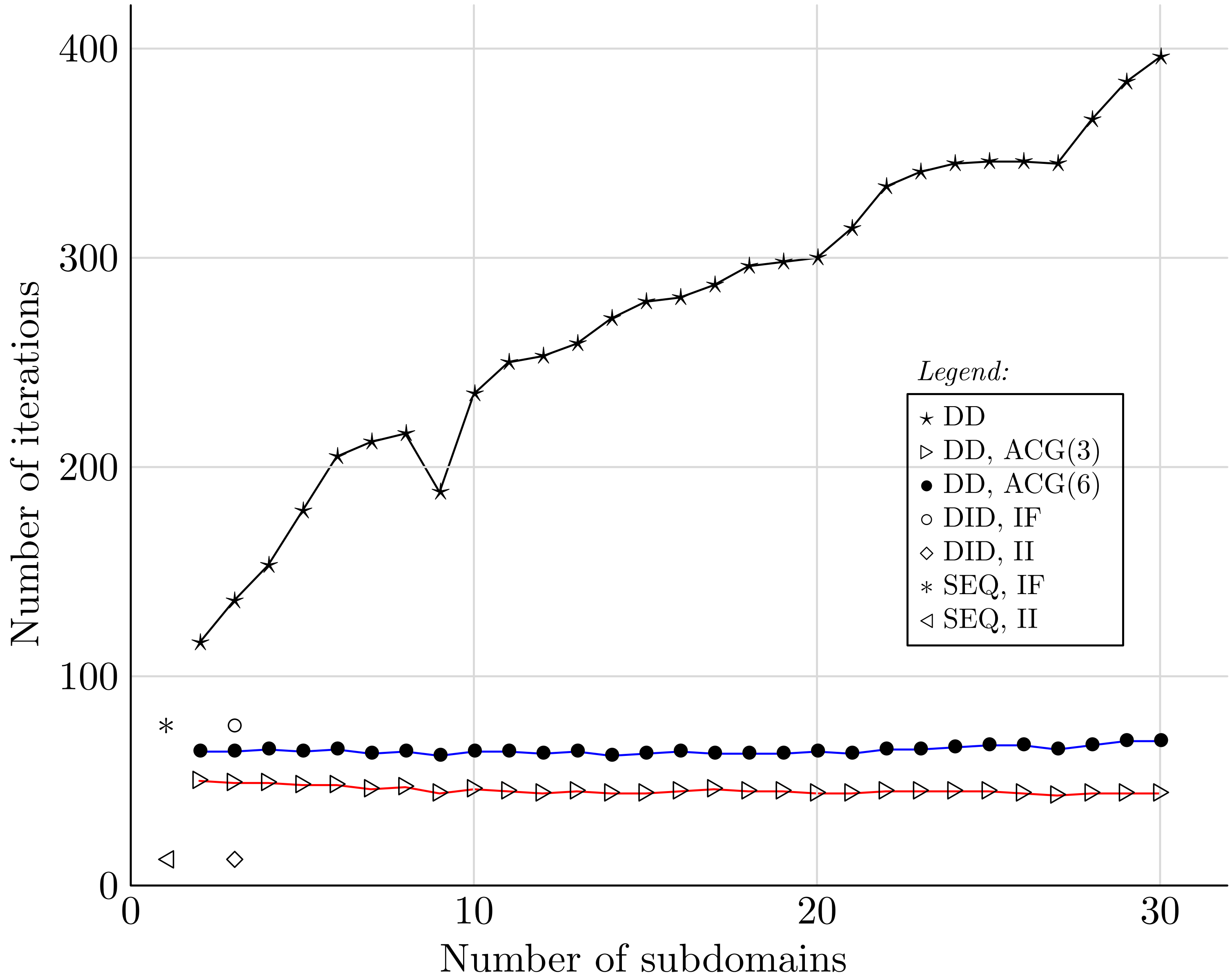
- the square footing is a 3D elasticity problem of soil mechanics
- the square area 10×10 m on the top side of the domain 100×100×40 m is loaded by the uniform pressure 2.4 MPa



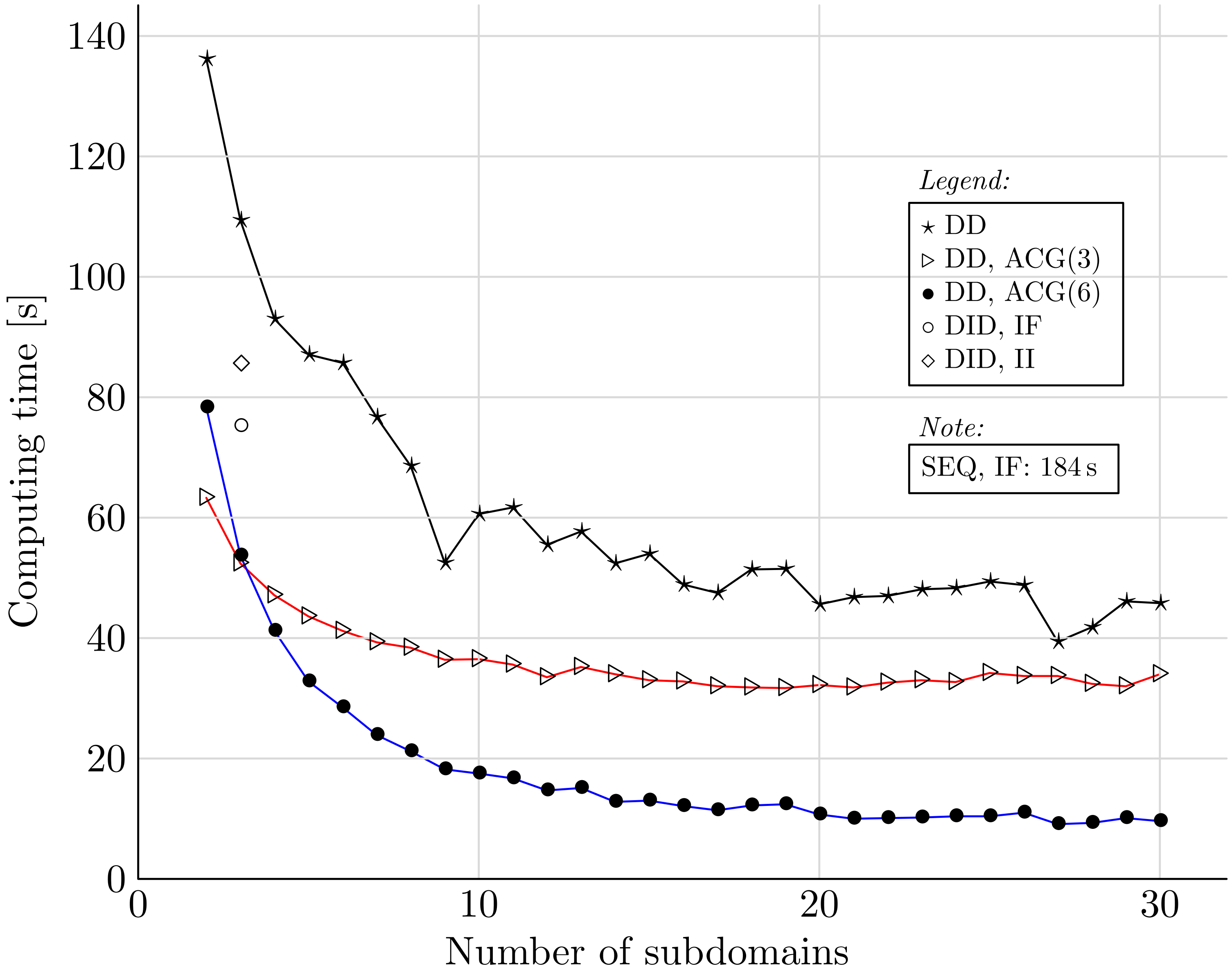
- due to the symmetry of the considered domain, we work only with its quarter discretized by a rectangular grid with a grid refinement under the footing

Benchmark	Used grid	# Equations
FOOT 40 E	41×41×41	206 763
FOOT 60 E	61×61×61	680 943
FOOT 80 E	81×81×81	1 594 323

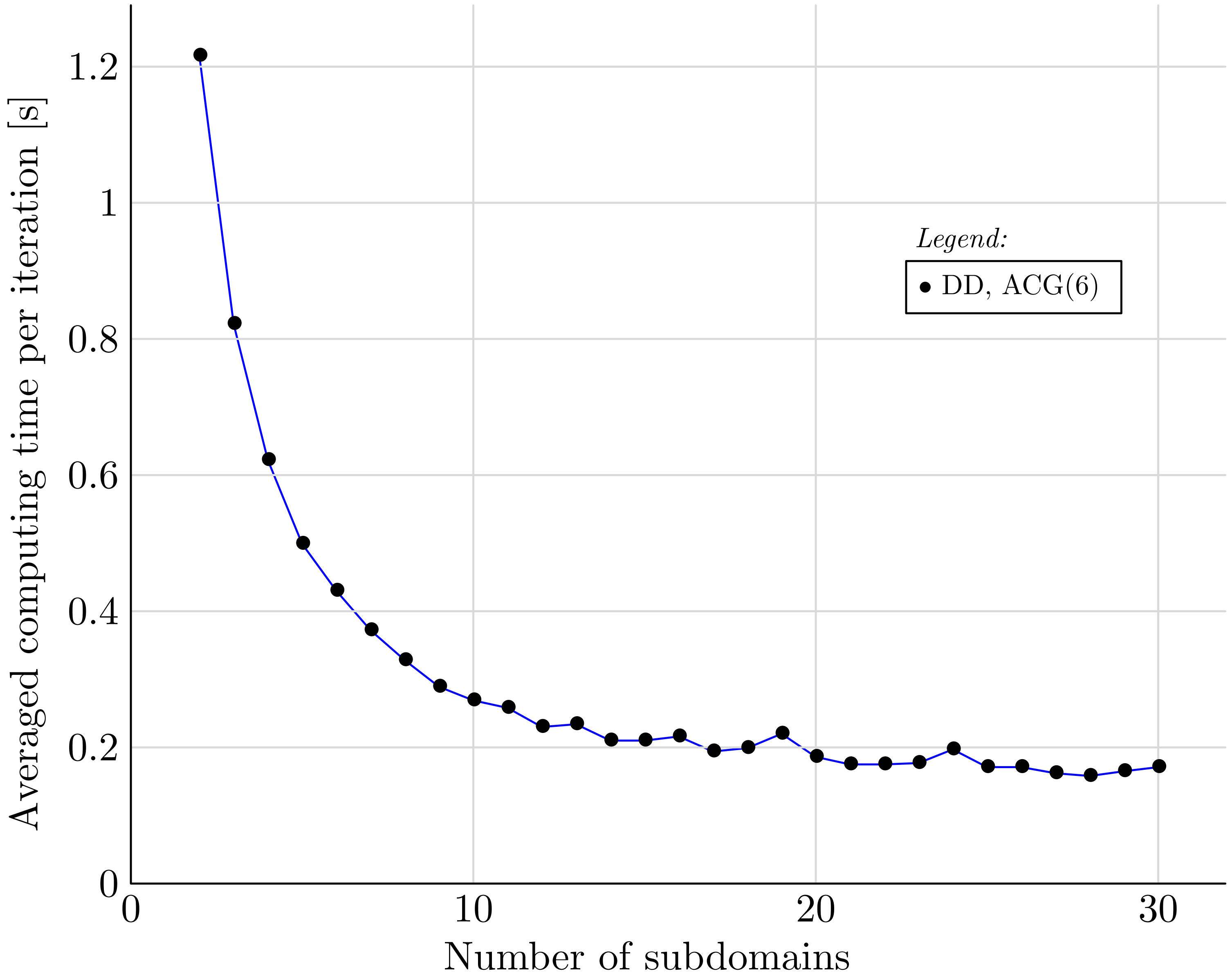
Results of FOOT 80 E on Termit (1)



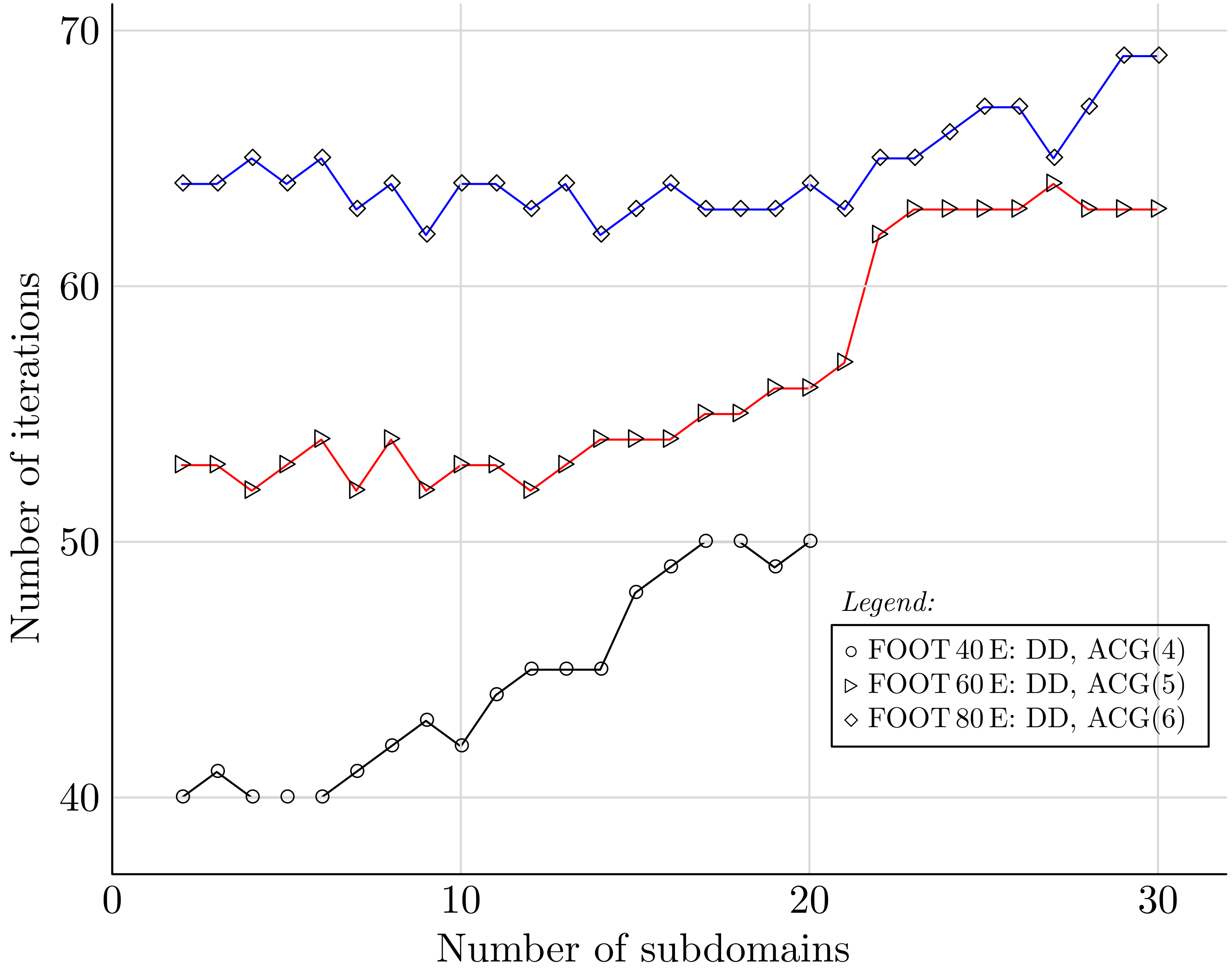
Results of FOOT 80 E on Termit (2)



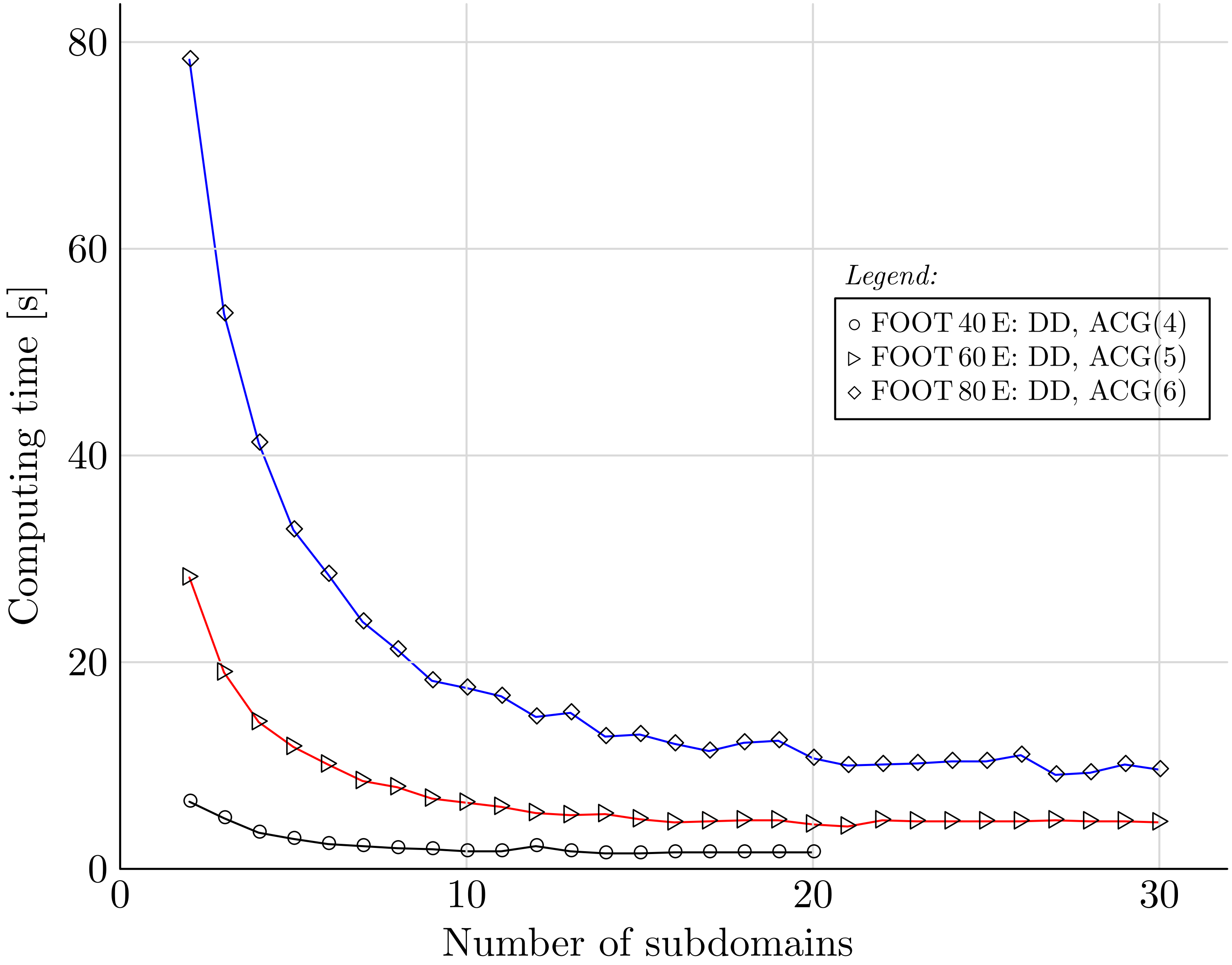
Results of *FOOT 80 E* on *Termit* (3)



Results of FOOT on Termit (1)

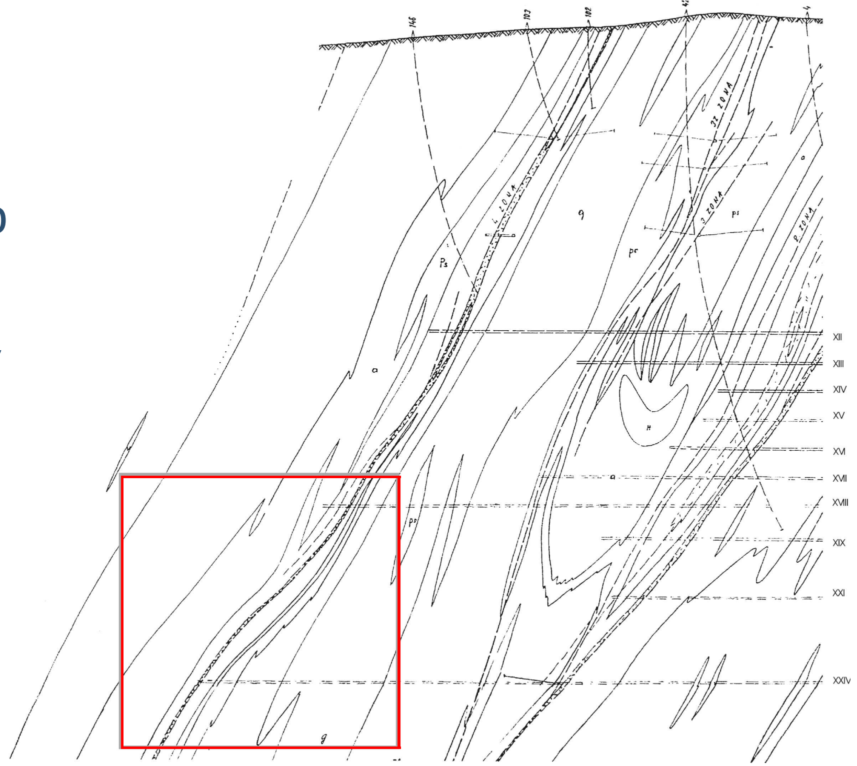


Results of FOOT on Termit (2)



Large-scale real-life problem: DR (1)

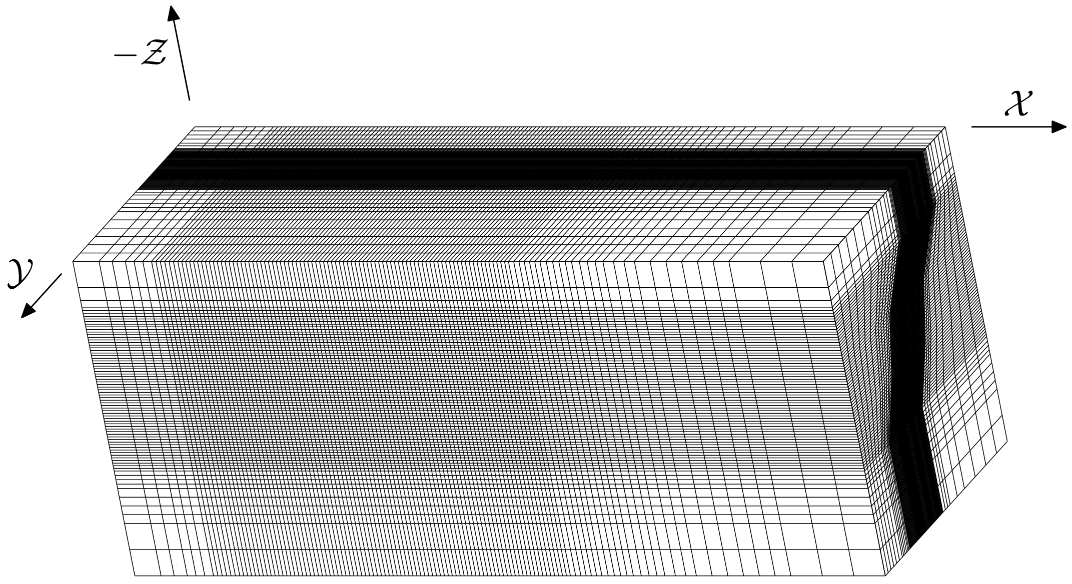
- benchmark is derived from the real-life large-scale mathematical model of a uranium ore mine Dolní Rožínka (DR) in the Bohemian - Moravian Highlands
- the model is considered for the comparison of different mining methods in relation to a development of induced stress fields and a possibility of dangerous rockbursts
- the DR model considers a domain of 1430×550×600 meters located 800 meters bellow surface



East-West cross-section of the deposit Rožná

Large-scale real-life problem: DR (2)

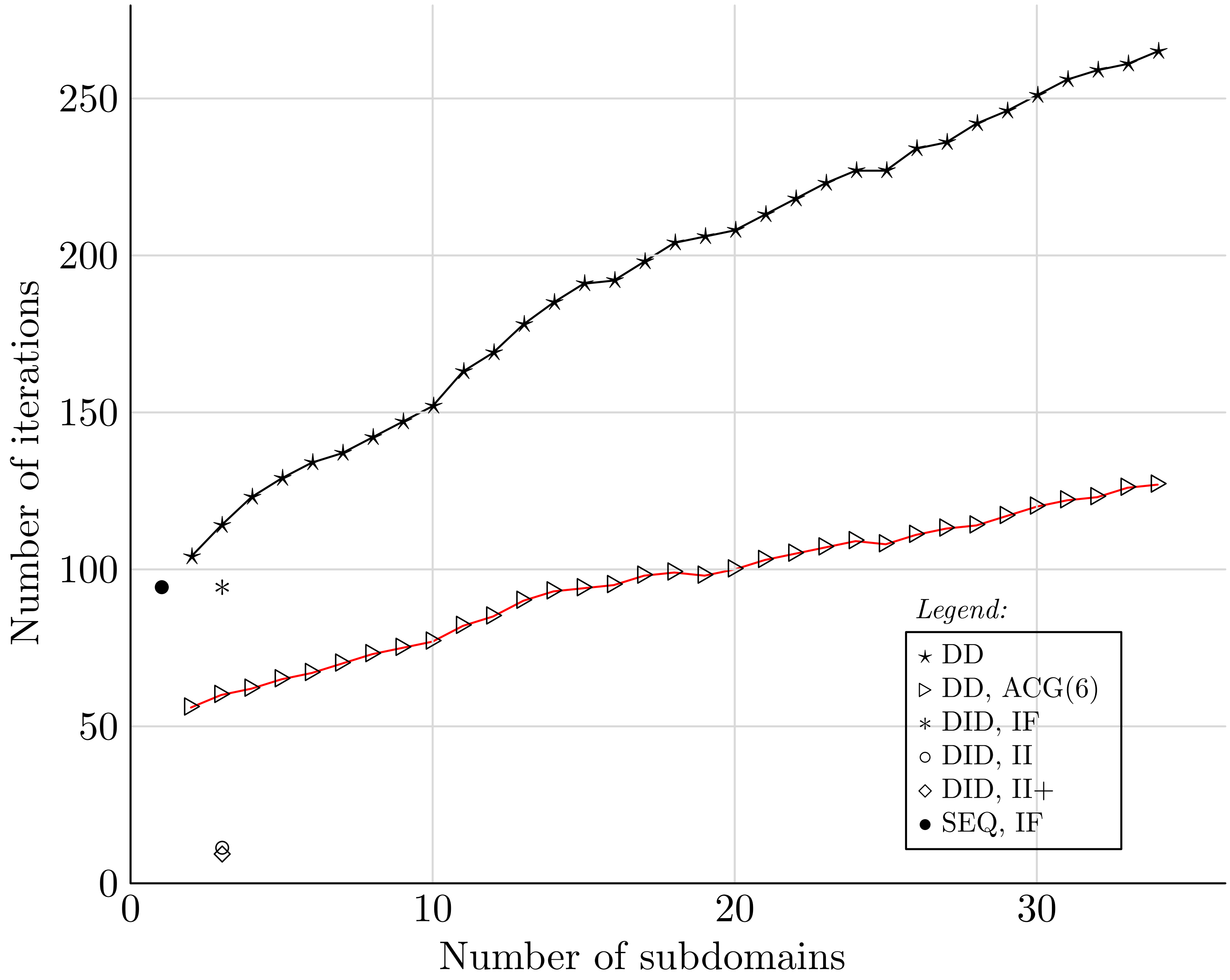
- the domain is discretized by a regular grid of $124 \times 137 \times 76$ nodes and the resulting linear system has 3 873 264 DOF



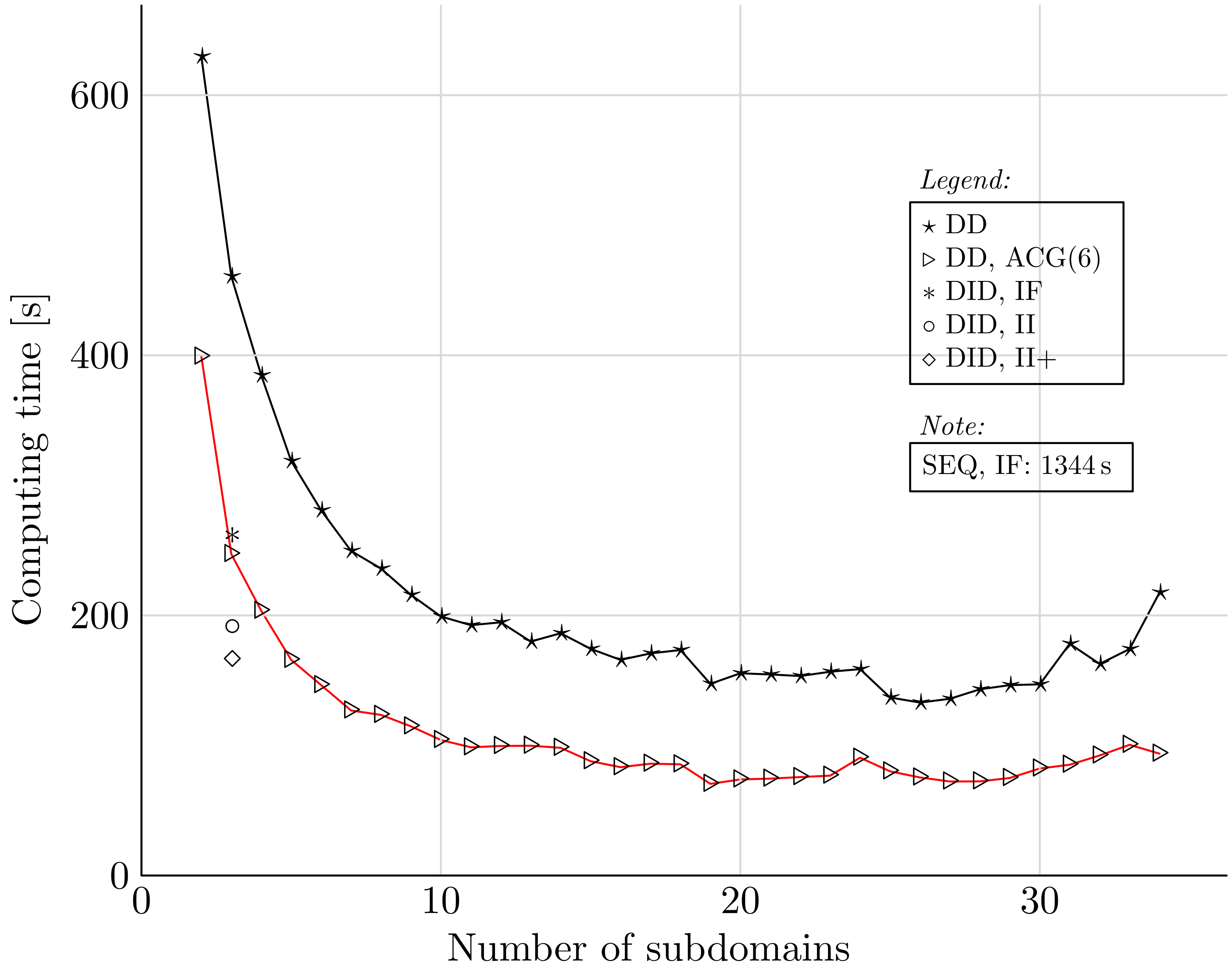
The FE mesh

- the whole task simulates four stages of mining, represented by a four-step sequence of problems with different material distributions
- for needs to use the big-size testing problem only, we work with the last step of the whole modelling sequence

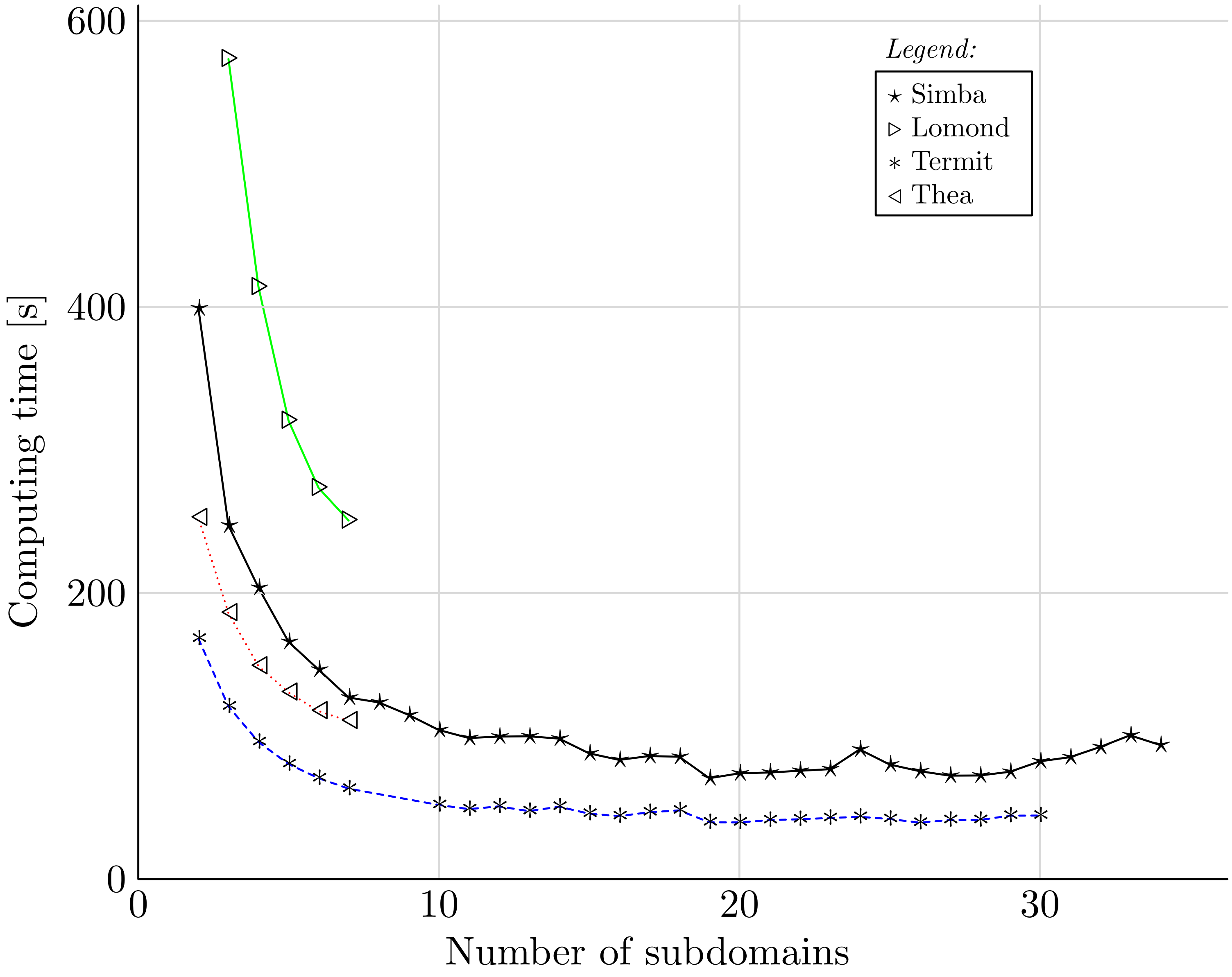
Results of DR on Simba (1)



Results of DR on Simba (2)

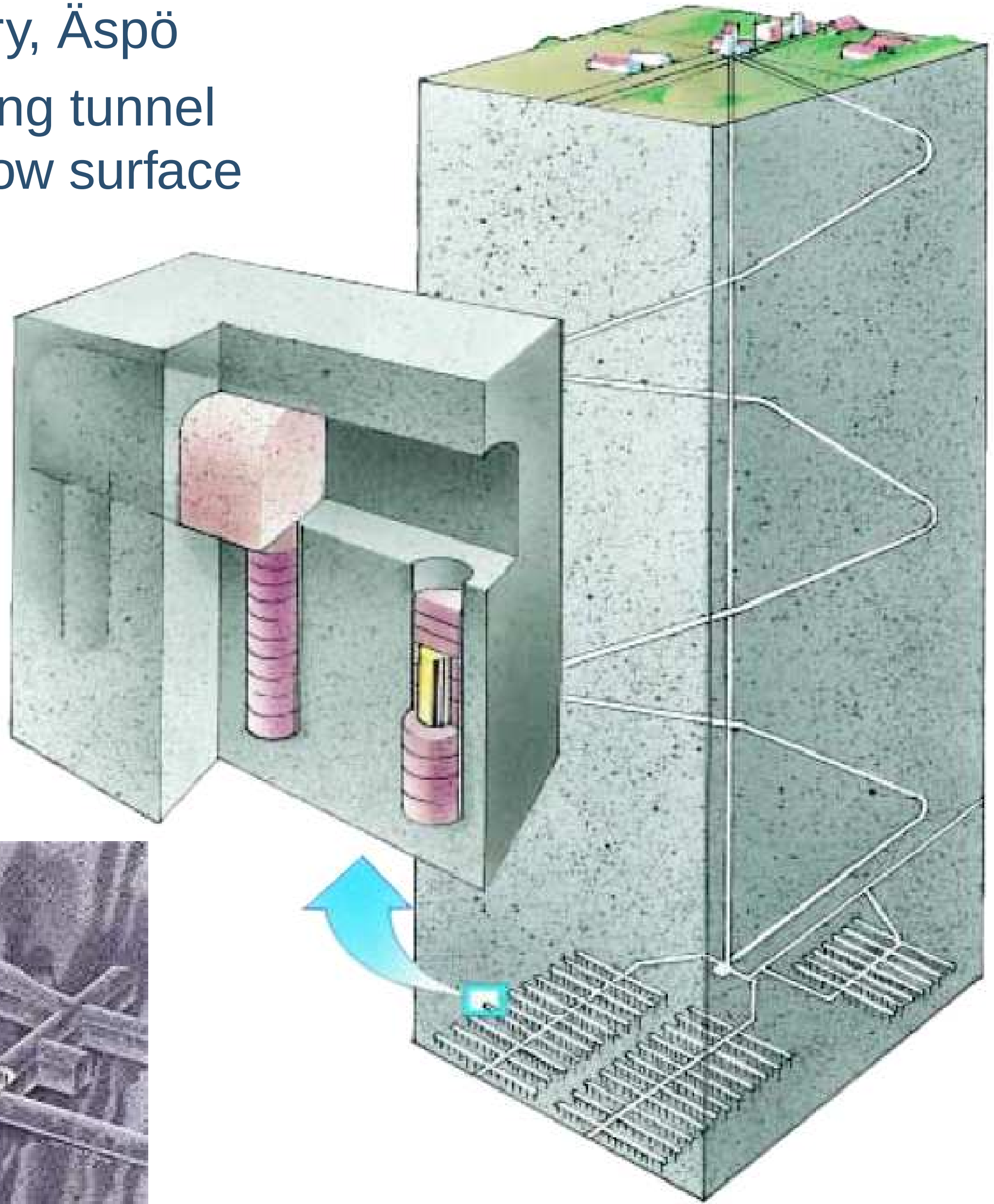


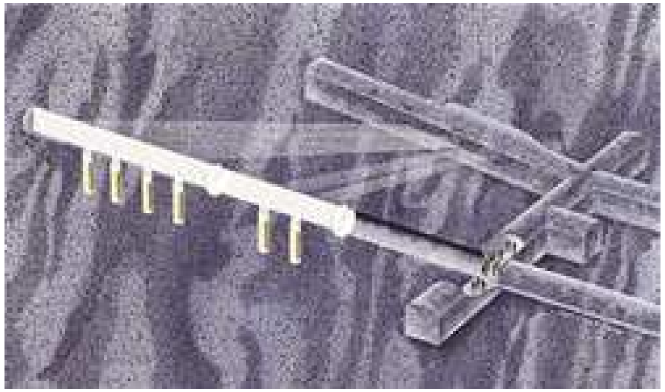
Results of DR on all computers



Modelling of thermo-elasticity: KBS (1)

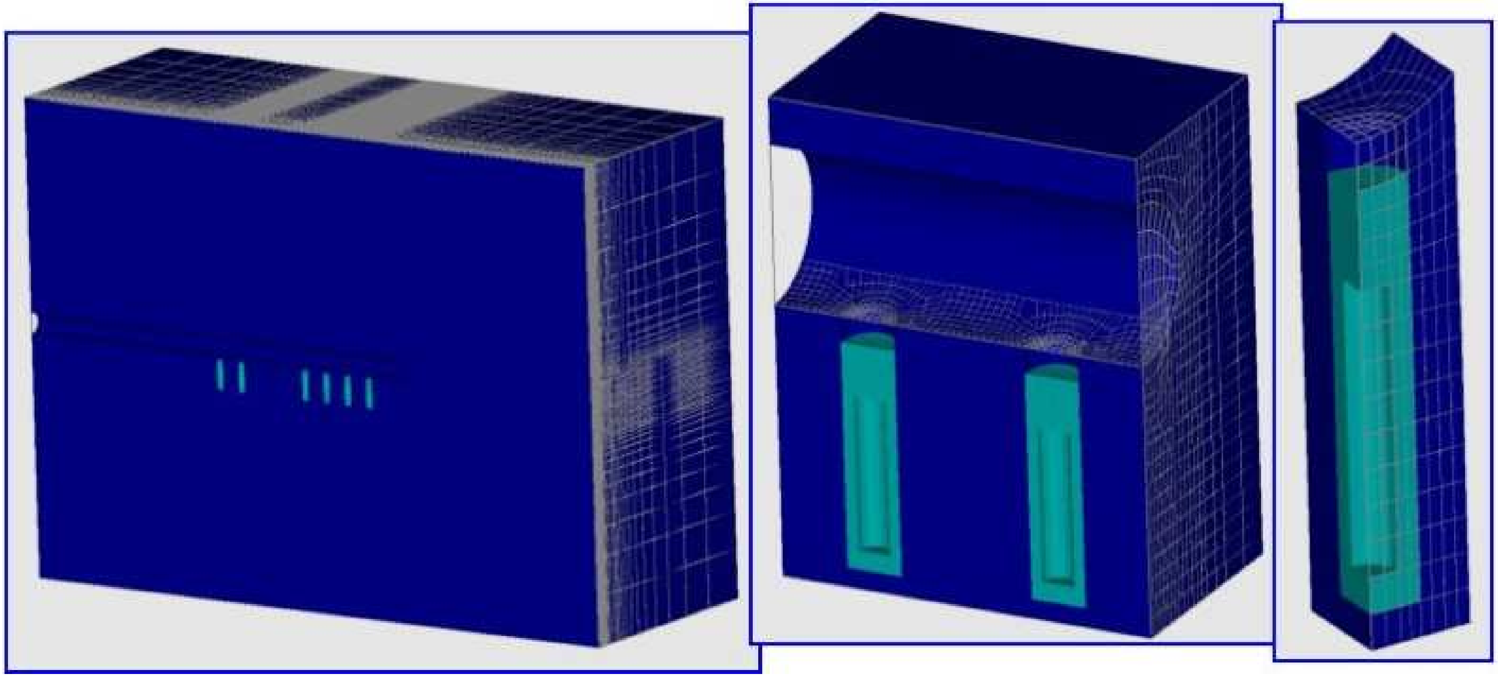
- prototype repository, Äspö
- consists of 65 m long tunnel located 450 m bellow surface
- deposition holes: 1.75 m diameter 8 m deep
- two sections: 4 DHs & 2 DHs





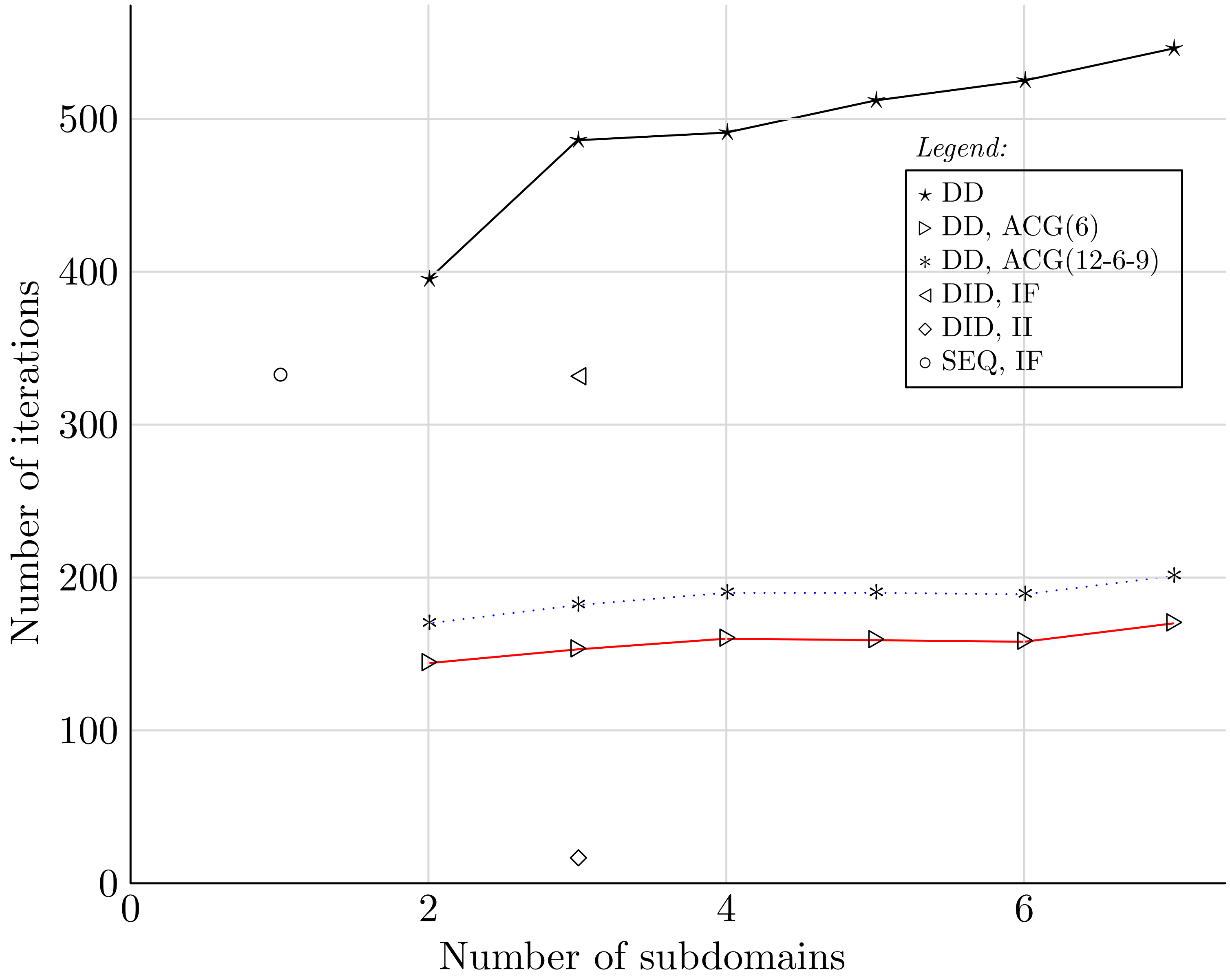
Modelling of thermo-elasticity: KBS (2)

- the FE mesh in 3D has 391×63×105 nodes, the considered time period is 50 years

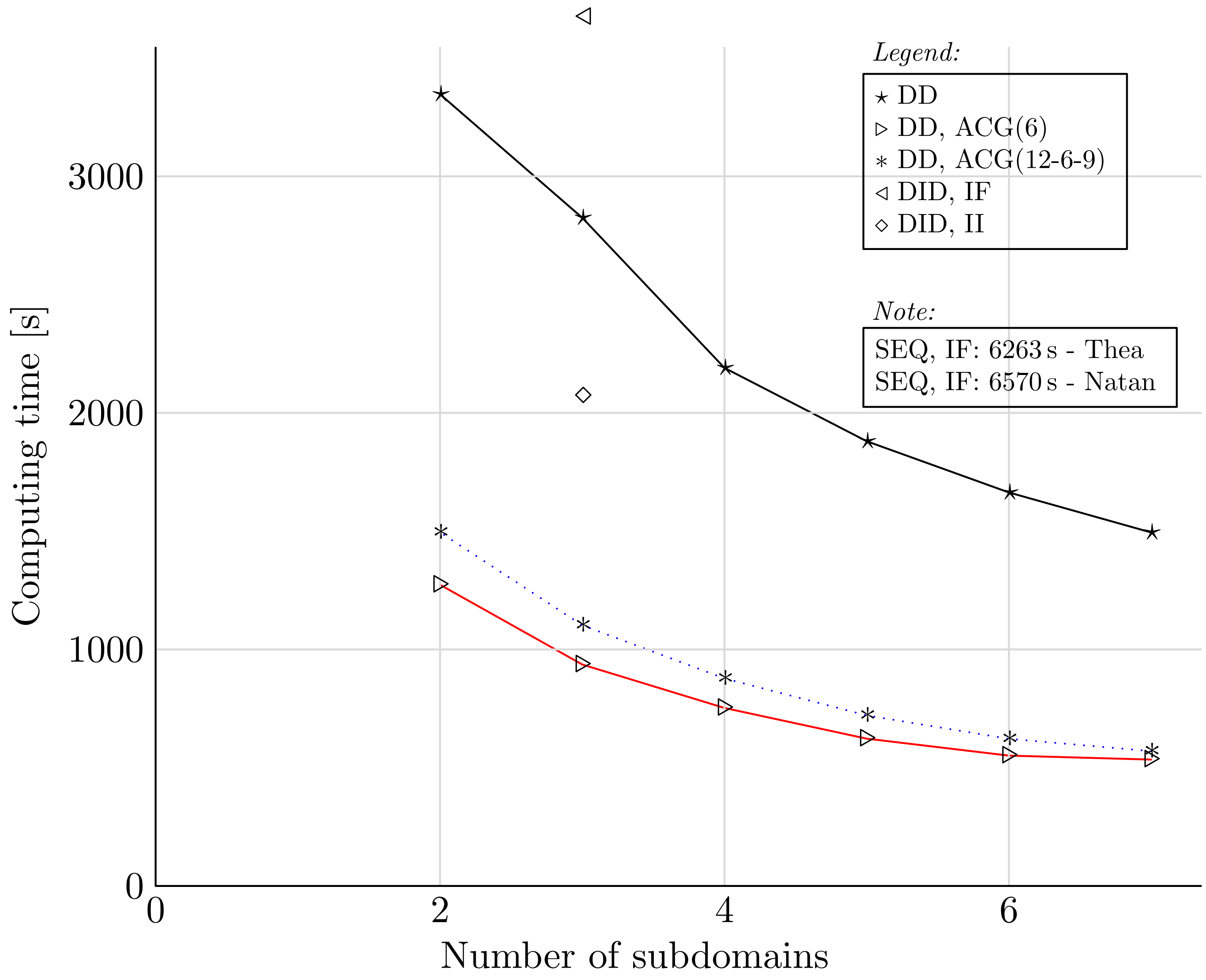


- the thermal part (2 586 465 DOF): radioactive waste as the heat source, heat conduction in rock, buffer and backfill
- the mechanical part (7 759 395 DOF): initial stress loading, tunnel excavation,heat load from the nuclear waste
- the benchmark includes the situation 2 years after the heat loading by the nuclear waste

Results of KBS on Thea (1)



Results of KBS on Thea (2)



Results of KBS on Thea (3)

Solver	# SD	T [s]	S	E
SEQ, IF	1	6263.8		
DID, IF	3	3673.5	1.71	0.57
DID, II	3	2071.7	3.02	1.01
DD	2	3346.9	1.87	0.93
	3	2824.8	2.22	0.74
	4	2190.1	2.86	0.71
	5	1880.1	3.33	0.67
	6	1663.5	3.77	0.63
	7	1494.5	4.17	0.60
DD, ACG(6)	2	1273.3	4.92	1.64
	3	936.1	6.69	1.67
	4	752.5	8.32	1.66
	5	622.7	10.06	1.68
	6	551.0	11.37	1.62
	7	534.3	11.72	1.47

Conclusion

This work outlines parallel implementation of the conjugate gradient method with various preconditioners and shows some applications of the implemented methods. The results prove that such parallel solvers enable efficient solution of large-scale real-life engineering problems even on clusters of common and relatively cheap PC's.

Acknowledgment: Authors are grateful for the support of the Academy of Sceinces of the Czech Republic, contract No. S3086102, the Grant Agency of the Czech Republic, contract No. 105/04/P036, and partners from the Uppsala university through the UPPMAX project *Parallel computing in geosciences*