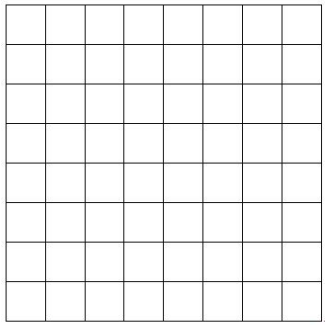
Schur complements and direct solvers applied on a simple decomposition model: trend and behavior of a monolevel strategy

J.P Boufflet, E. Lefrançois, M. Vayssade

UTC, UMR 6599 Heudiasyc & UMR 6066 Roberval

Our objective is to show trend and behavior of a monolevel strategy using a simple domain decomposition model of a regular plate structure (grid)



assumptions:
- domain decomposition
- Schur complement method
- direct solver

This regular grid has been studied by A. George in the 70's using a nested dissection model for sequential computing.

For parallel computing a different approach is needed.

In this context of parallel computing, the feasability and performances depend on the choices made on:

- The decomposition of the initial domain:
  - number of subdomains
  - number of nodes of a subdomains
  - number of nodes on the interfaces, etc.
- the gathering of the computing tasks into parallel tasks
- the generation of the task graph
- the scheduling of this task graph

We first presents the decomposition model that permits to explicitely compute parameters useful to estimate storage and number of operations.

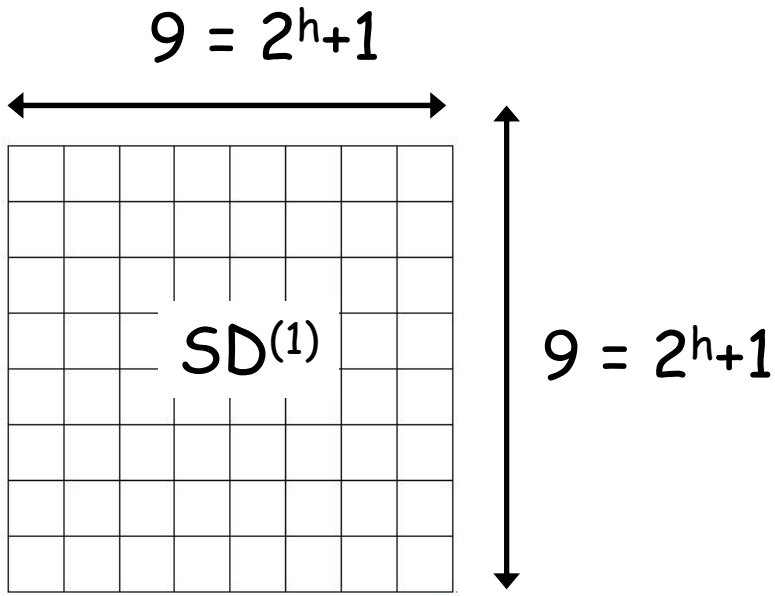
This regular decomposition will permit to model and study different solving strategies for numerical treatment chains

As an example, we use our model on a monolevel strategy

A simple domain decomposition model

Main parameters of the model

- h, grid size
- d, number of decomposition steps



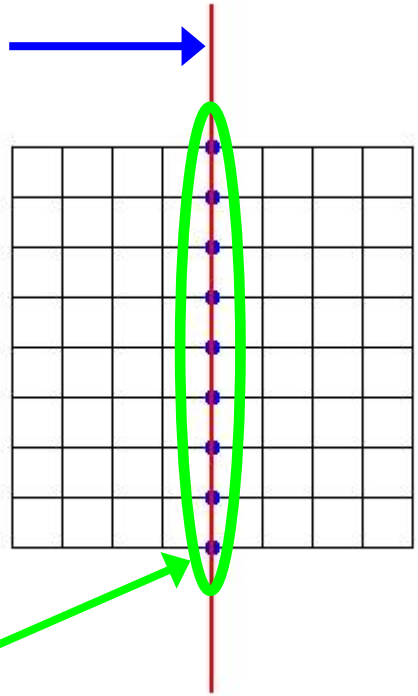
h = 3

d = 0 no decomposition, initial plate

$n_{SD}(d=0) = (2^h+1) \,.\, (2^h+1) = 81$

decomposition step d = 1

One vertical cutting line



number of duplicated nodes:

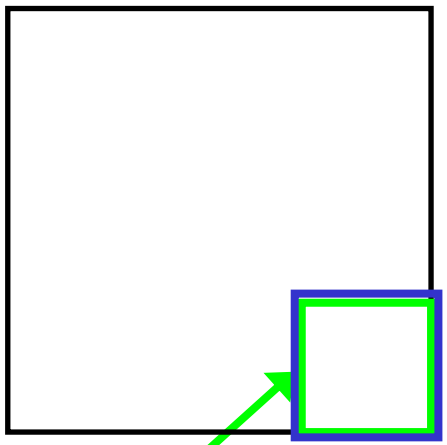
$n_{new\_dup}(d=1) = 2^h+1$

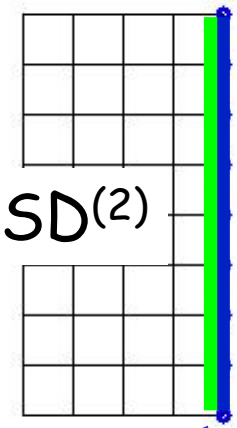
Number of nodes across a cutting line

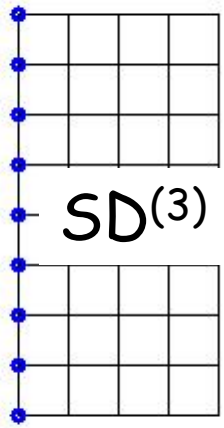
decomposition step d = 1

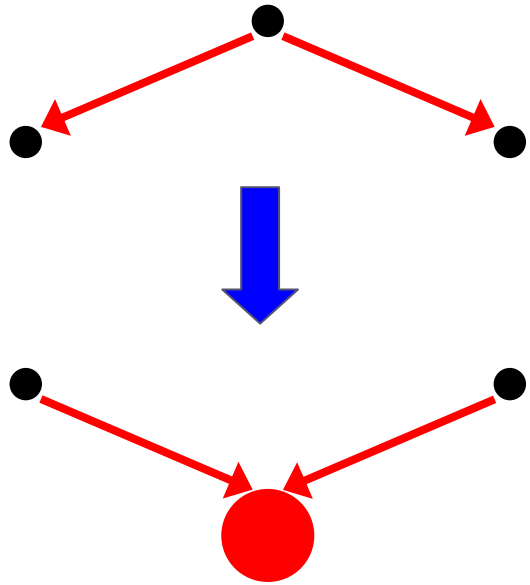
$n_{SD}(d) = (2^{(h-\lceil d/2 \rceil)}+1) \cdot (2^{(h-\lfloor d/2 \rfloor)}+1)$

subdomain matrix:









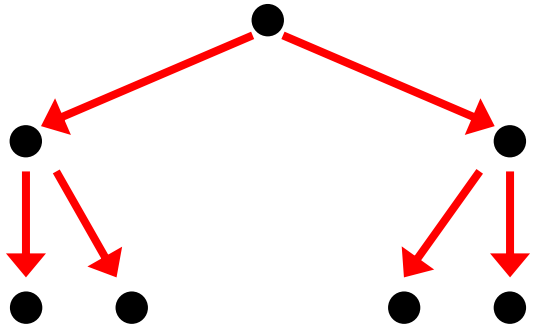
shape of the unique interface

$n_{new\_dup}(d=1) = 2^h+1$

$n_{larg\_intf}(d=1) = 2^h+1$

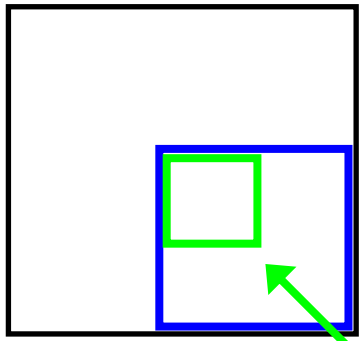
$n_{unique\_interf}(d=1) = 2^h+1$

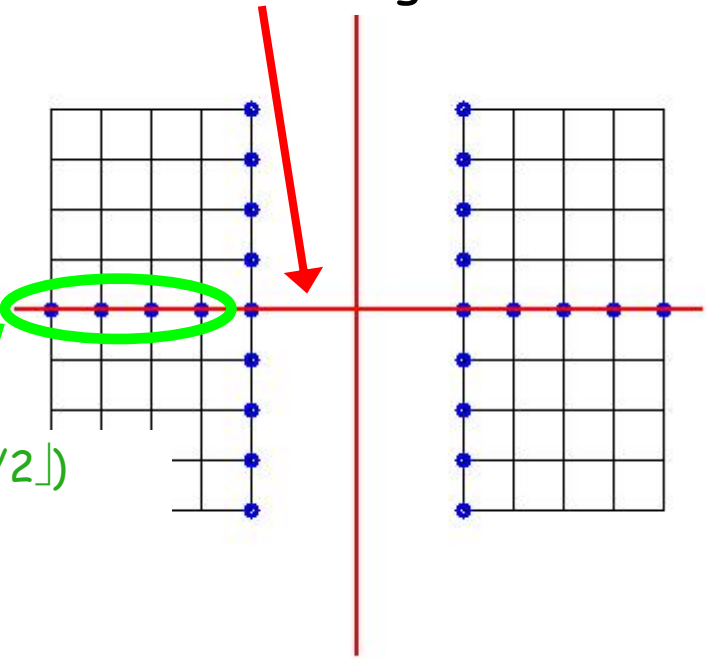
decomposition step d=2



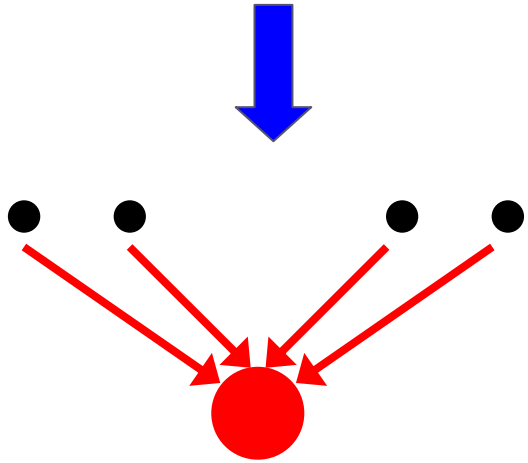
a new horizontal cutting line

subdomain matrix:

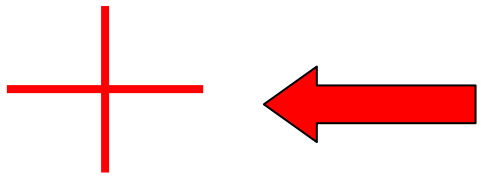




$n_{new\_dup}(d=2) = 2^{(h-\lfloor d/2 \rfloor)}$



shape of the unique interface

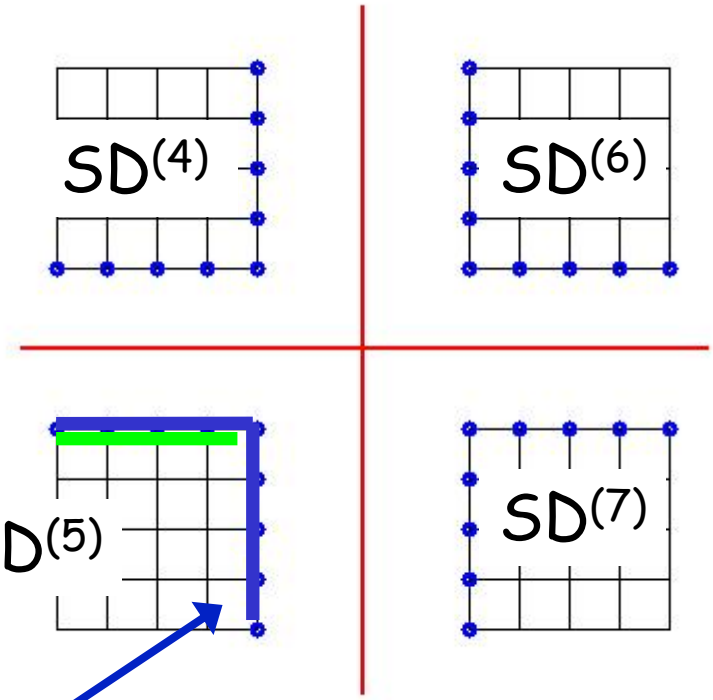


$n_{unique\_interf}(d=2) = 2.(2^h+1)-1$

decomposition step d=2

Number of subdomains:

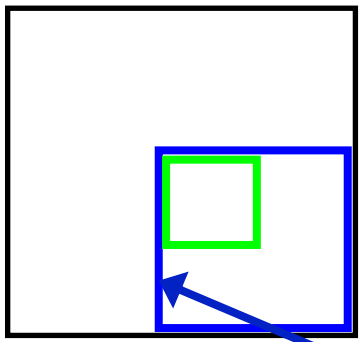
$N_s(d=2) = 2^d = 4$



for a father subdomain number i, we have 2 new subdomains numbered:

1. (2 . i)
2. (2 . i) + 1

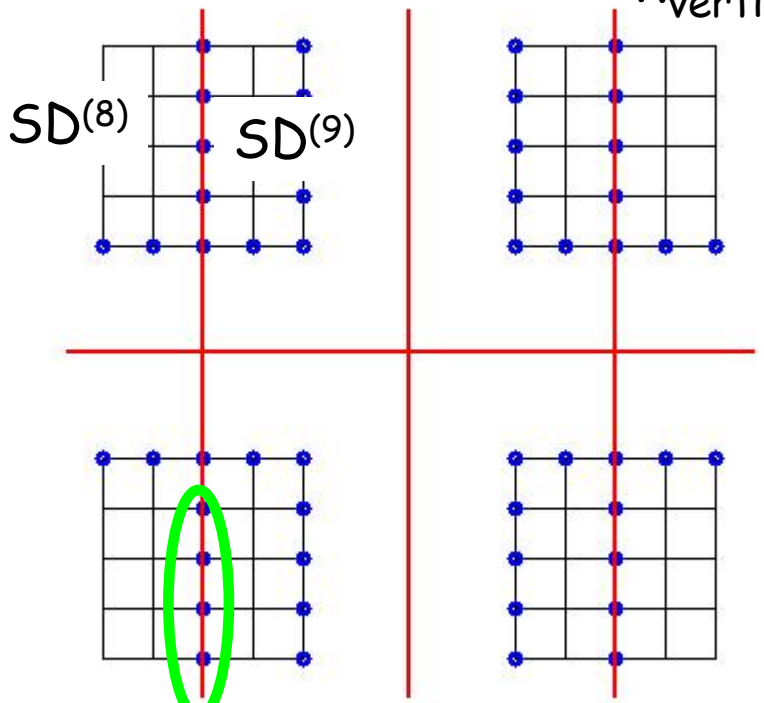
subdomain matrix:



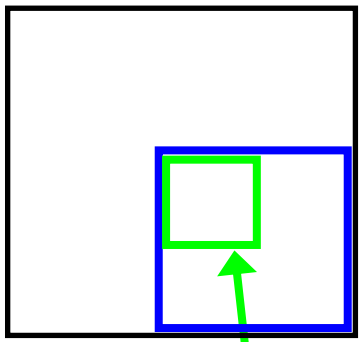
$n_{larg\_intf}(d=2) = 2.(2^{h-1}+1)-1 = 2^h+1$

decomposition step d=3

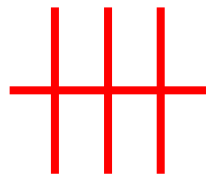
$n_{vertical\_line}(d=3) = 2^{\lceil d/2 \rceil}-1$



subdomain matrix:



shape of the unique interface

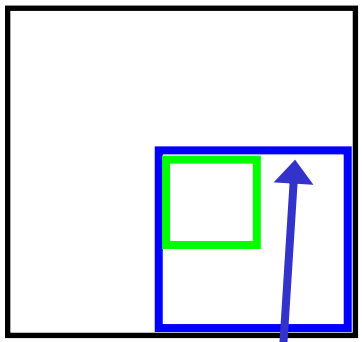


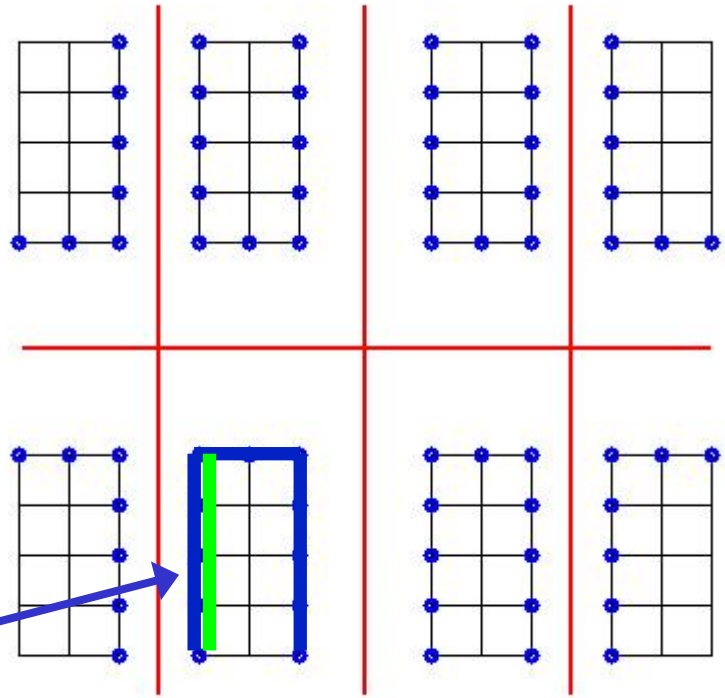
$n_{new\_dup}(d=3) = 2^{(h-\lfloor d/2 \rfloor)}$

$n_{unique\_interf}(d=3) = 4.(2^h+1)-3$

decomposition step d=3

subdomain matrix:

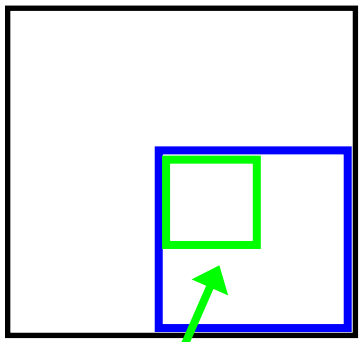


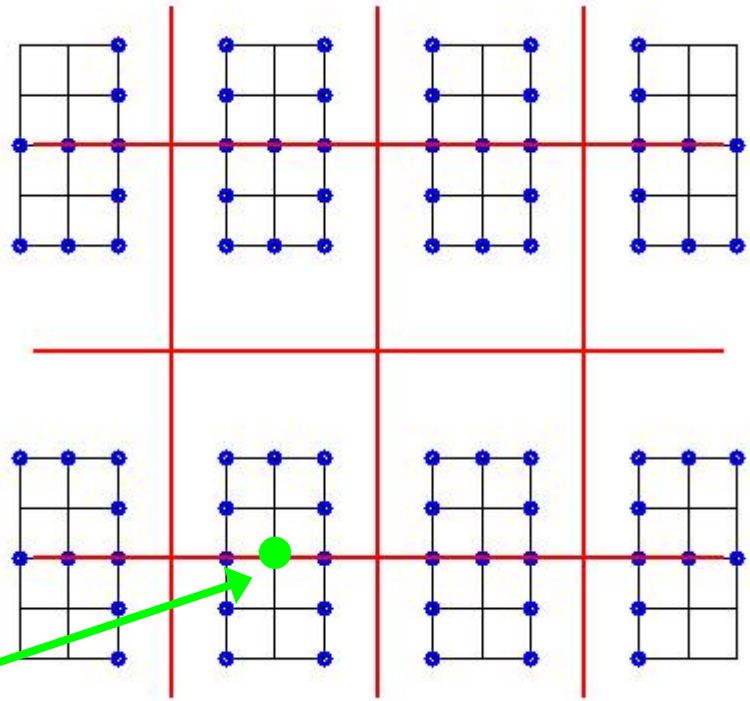


$$n_{larg\_intf}(d=3) = 2.(2^{h-1}+1)+(2^{h-2}+1)-2$$

decomposition step d=4

subdomain matrix:





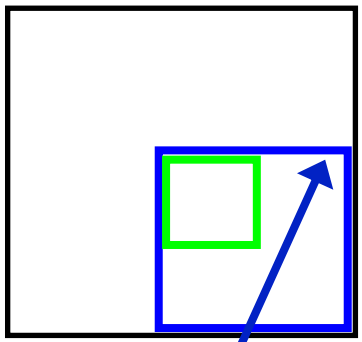
$n_{new\_dup}(d=3) = 2^{(h-\lfloor d/2 \rfloor)}-1$

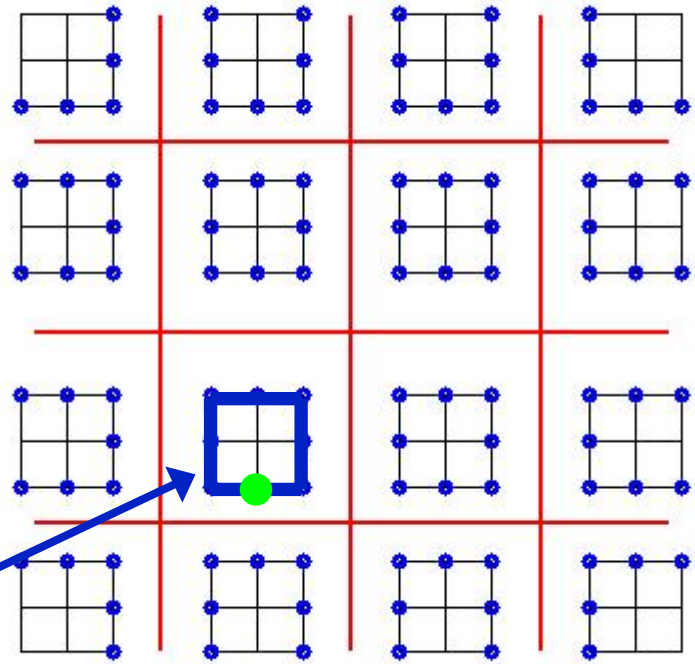
(minimum number of duplicated nodes)

$n_{horizontal\_line}(d) = 2^{\lfloor d/2 \rfloor}-1$

decomposition step d=4

subdomain matrix:

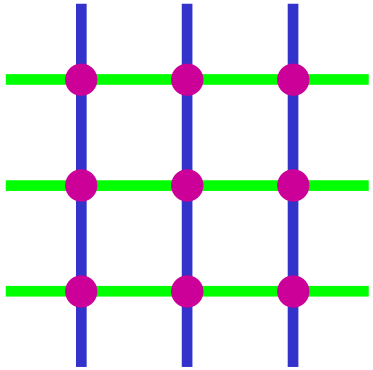


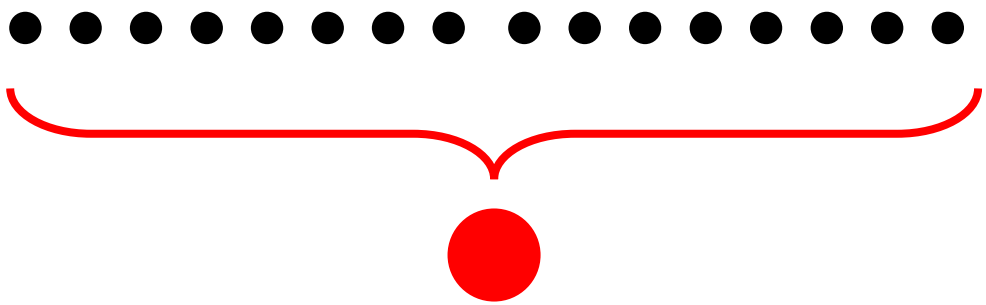


$$n_{larg\_intf}(d=4) = 2.(2^{(h-\lceil d/2 \rceil)} + 1) + 2.(2^{(h-\lfloor d/2 \rfloor)} + 1) - 4$$

unique interface problem, decomposition step d=4

shape of the unique interface





$n_{vertical\_line}(d)$

$n_{horizontal\_line}(d)$

$$n_{unique\_interf}(4) = ((2^{\lceil d/2 \rceil} - 1) + (2^{\lfloor d/2 \rfloor} - 1)).(2^h+1) - ((2^{\lceil d/2 \rceil} - 1).(2^{\lfloor d/2 \rfloor} - 1))$$

Intersection points

Number of nodes across a cutting line

regular behavior for $d \geq 4$

So, we can explicitely compute:

$N_s(d)$: number of subdomains
$n_{SD}(d)$: number of nodes of a subdomain
$n_{vertical\_line}(d)$: number of vertical cutting lines
$n_{horizontal\_line}(d)$: number of horizontal cutting lines
$n_{new\_dup}(d)$: number of new duplicated nodes
$n_{larg\_intf}(d)$: number of nodes of the biggest interface
$n_{unique\_interf}(d)$: number of nodes of a unique interface problem

Estimations for dense case

Subdomain storage: $(n_{SD}(d))^2$

For all subdomains: $N_s(d) \cdot n_{SD}(d)^2$

Unique interface storage: $(n_{unique\_interf}(d))^2$

Sending of blocks to build the unique interface: $(N_s(d)-1) \cdot (n_{unique\_interf}(d))^2$

Unique interface factorization cost: $2/3(n_{unique\_interf}(d))^3$

Partial factorization cost of a subdomain:

$$\sum_{i=0}^{n_{SD}(d) - n_{larg\_intf}(d)} (n_{SD}(d)-i)^2 \longrightarrow n_{SD}(d) - n_{larg\_intf}(d) \text{ factorization steps}$$

At this stage, we can design a model of a solver in the dense case and shows its trends and behaviors

How trends and behaviors evolve when sparsity is exploited ?

We introduce an simple estimation of the maximum semi-bandwidth on the subdomains in order to show the evolutions

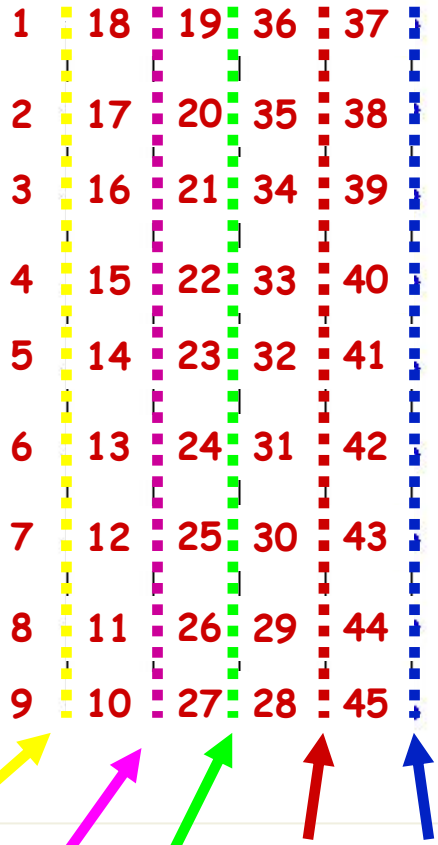
Semi-bandwidth estimation for each shape of subdomain

this relation holds over levels:

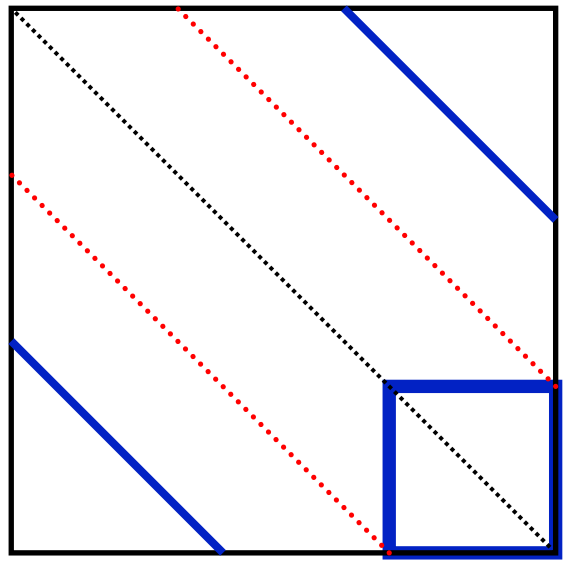
$W_k \leq \max(b_i) \leq W_k + W_{k-1} - 1$

45 x 45 full matrix

9 x 9 block

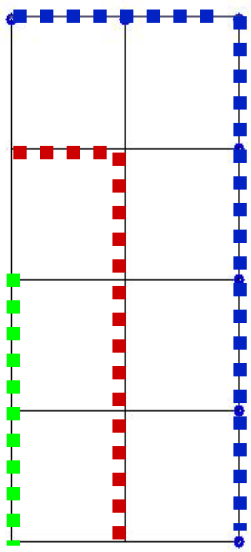


$b_{max} = 2 \cdot n_{larg\_interf}(d) - 1$

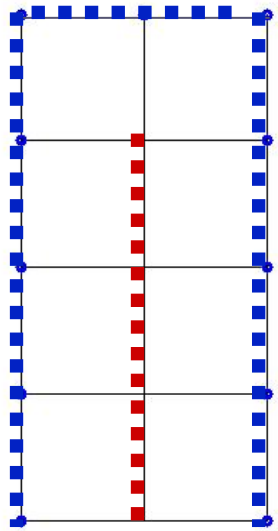


$W_4 \leq W_3 \leq W_2 \leq W_1 \leq W_0 = 9 = W_{max} = n_{larg\_interf}(d)$

Other shape of subdomains

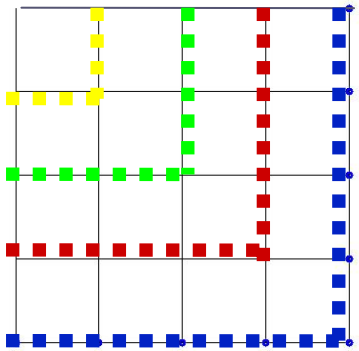
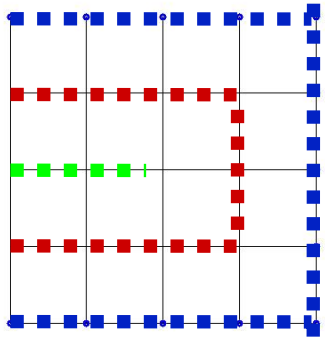
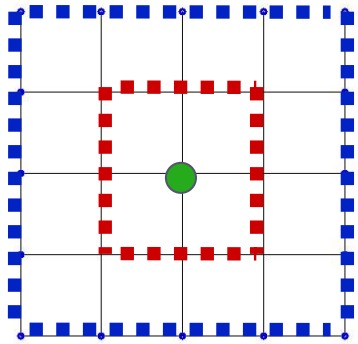


$W_2 \leq W_1 \leq W_0 = W_{max} = n_{larg\_interf}(d)$



$W_1 \leq W_0 = W_{max} = n_{larg\_interf}(d)$

Other shape of subdomains

Memory = $n_{SD}(d) \,.\, (2 \,.\, b_{max}(d) + 1)$

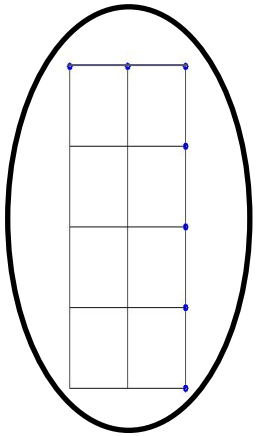
Partial facto cost = $\frac{1}{2} (b_{max}(d))^2 \,.\, (n_{SD}(d) - n_{larg\_interf}(d))$

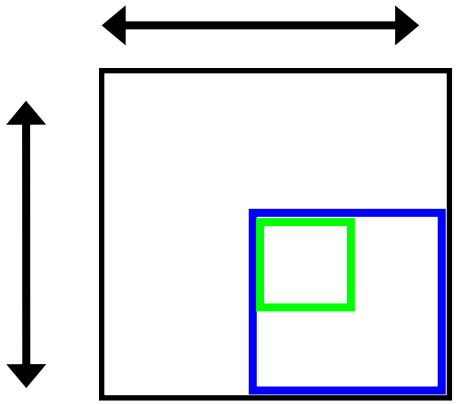
ref : Direct methods for sparse matrices, I. Duff, A Erisman, and J. Reid

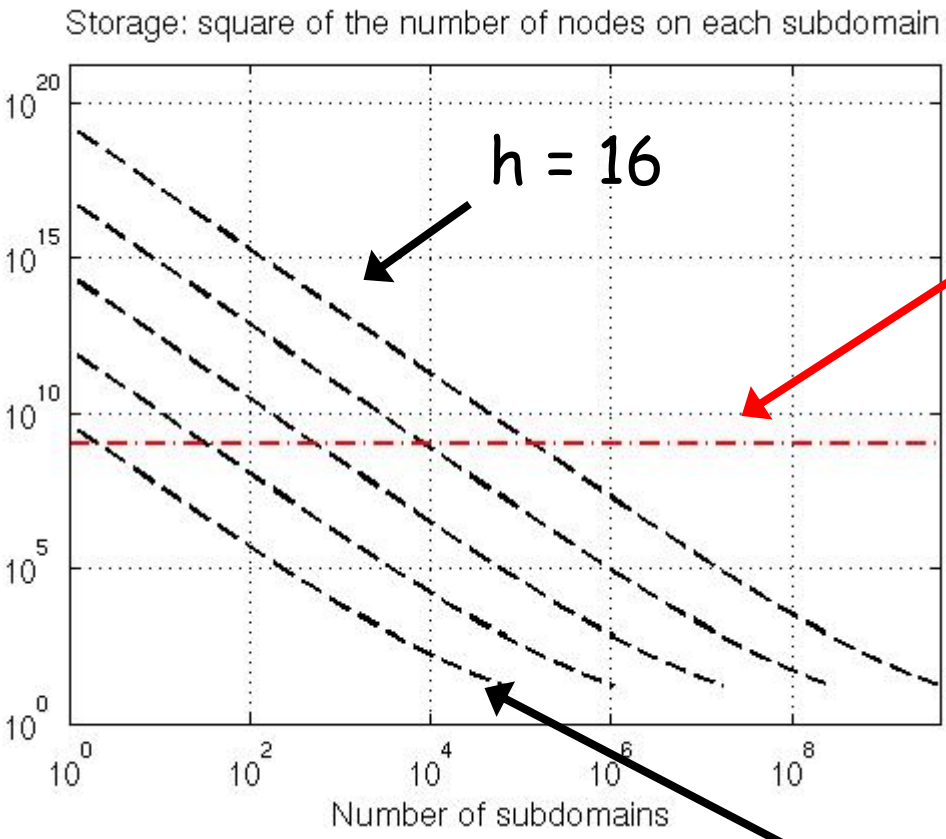
At this stage, we can design a model of a solver for the band case and show its trends and behaviors

This is a rough estimation, far from the results efficient ordering methods can provide, but enough to compare with the dense case

Example 1: storage, dense, subdomain



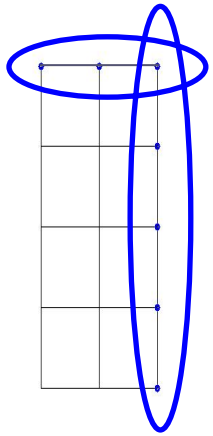


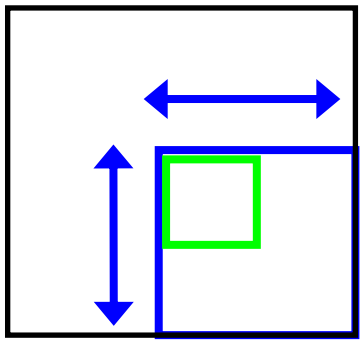


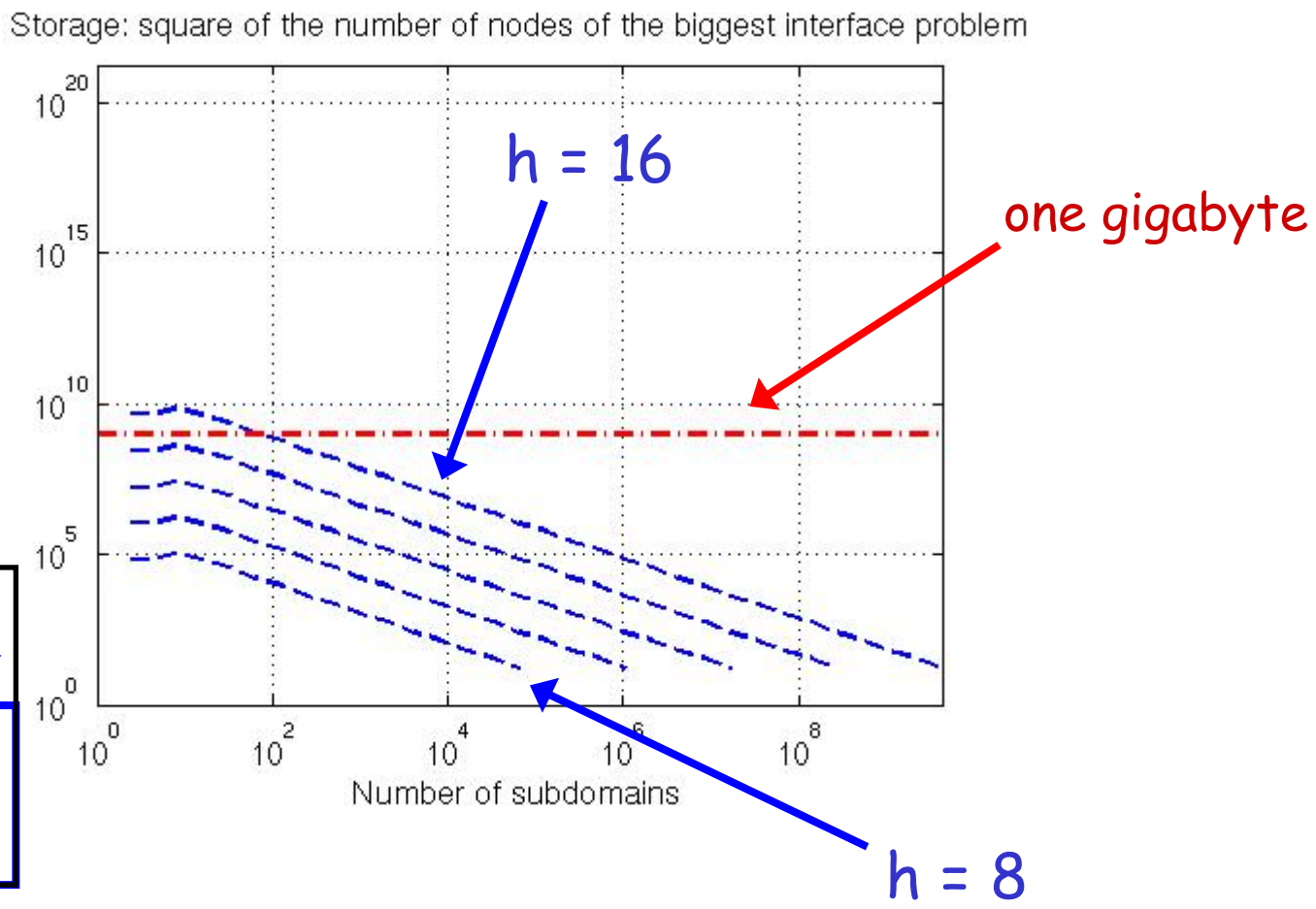
one gigabyte

(loglog plot)

Example 2: storage, dense, interface



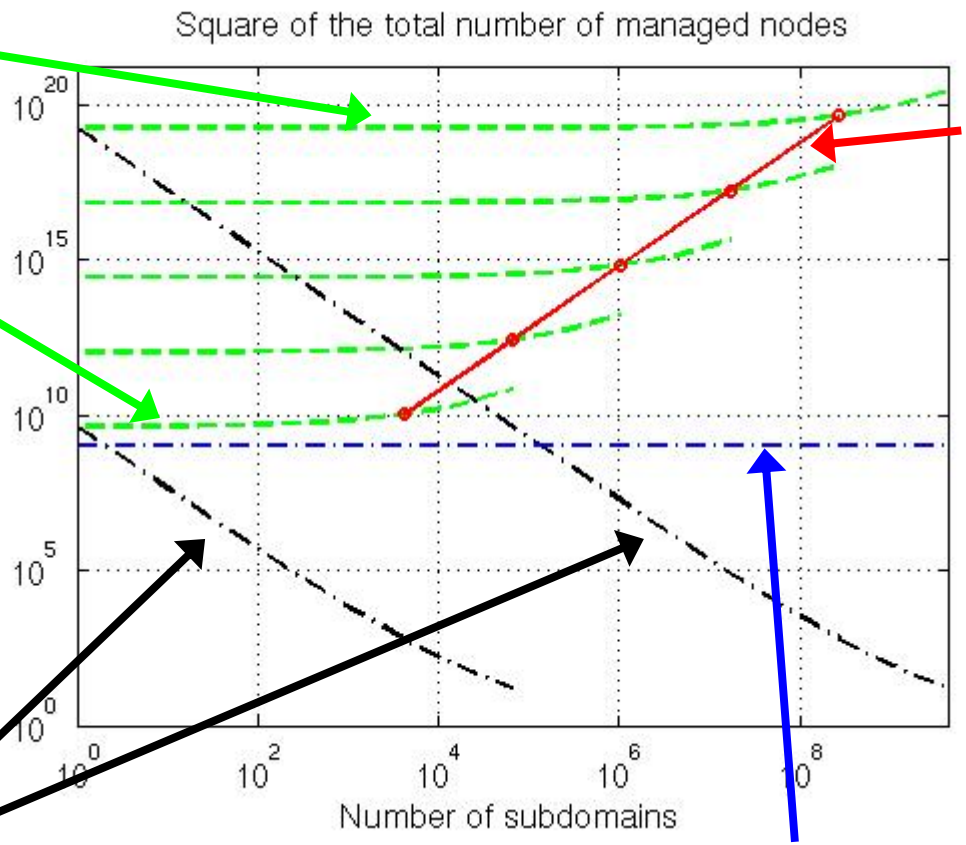




Example 3: storage, dense, total amount

h = 16

h = 8



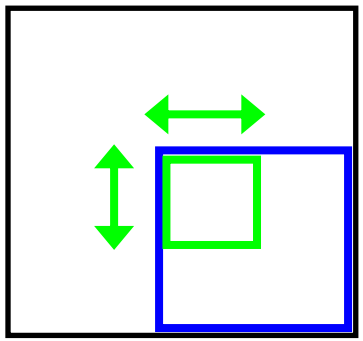
This line shows the points where the initial amount of data is double:

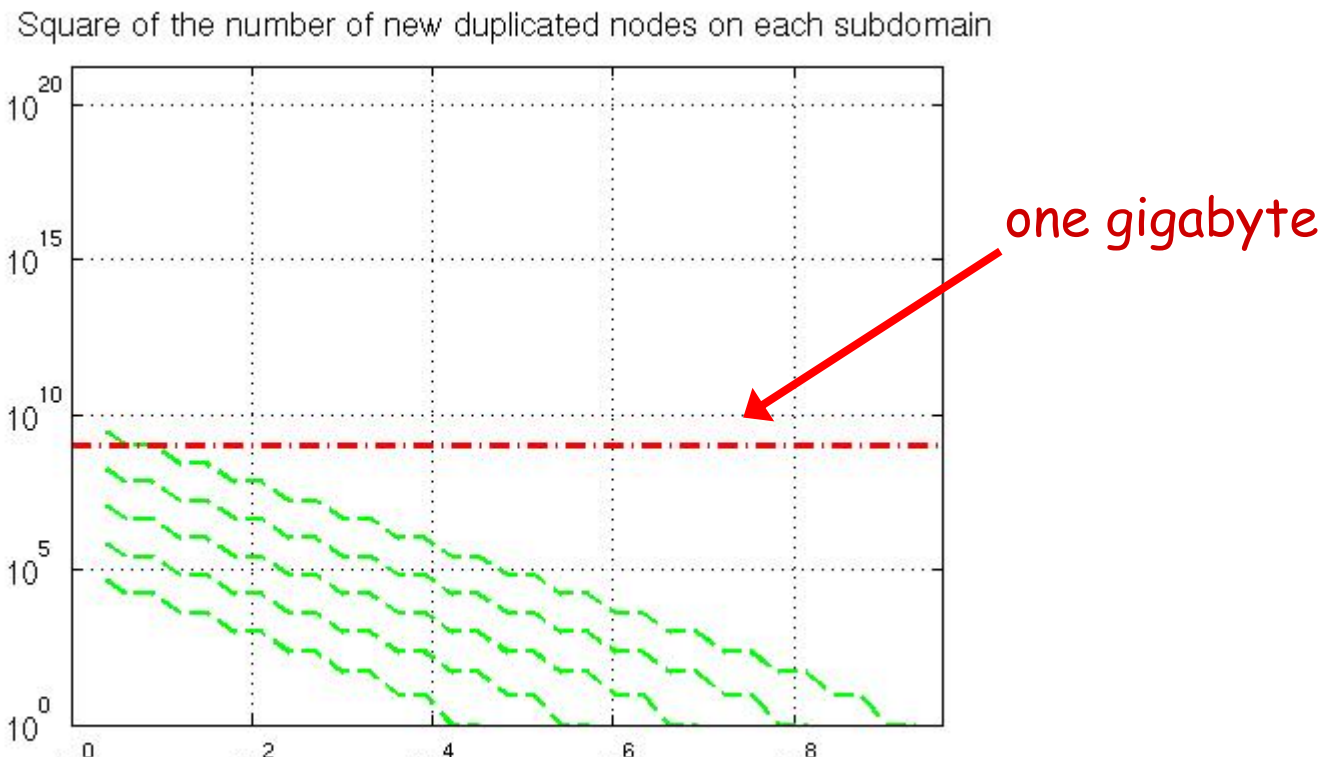
the duplication of nodes has not a significative impact on storage

storage, dense, subdomain h = 8 and h = 16

one gigabyte

Example 4: new duplicated nodes





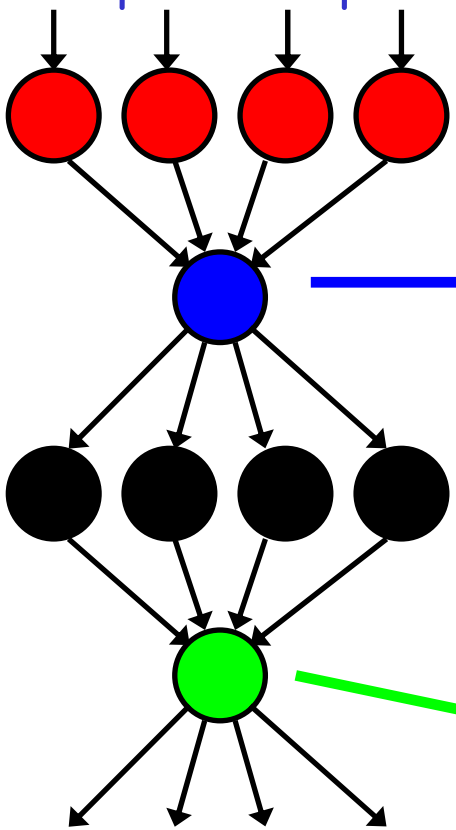
n_{new_dup} can be use to model a multi-level strategy, if we use the reverse order of decomposition to treat partial interface problems

An example to illustrate the use of the model:

trends and behaviors of a monolevel strategy

An example of a monolevel strategy

from the previous part of the chain



type A // tasks: $\mathcal{E}_A$
- subdomains assembling
- partial factorization
- extraction of local Schur complements

a unique type B task: $\mathcal{E}_B$
- local Schur complements assembling
- factorization of a unique interface problem
- backward and forward substitutions of interface variables

type C // tasks: $\mathcal{E}_C$
- backward and forward substitutions of internal variables

a unique type D tasks: $\mathcal{E}_D$
- the vector of variables is built

We consider it as negligible

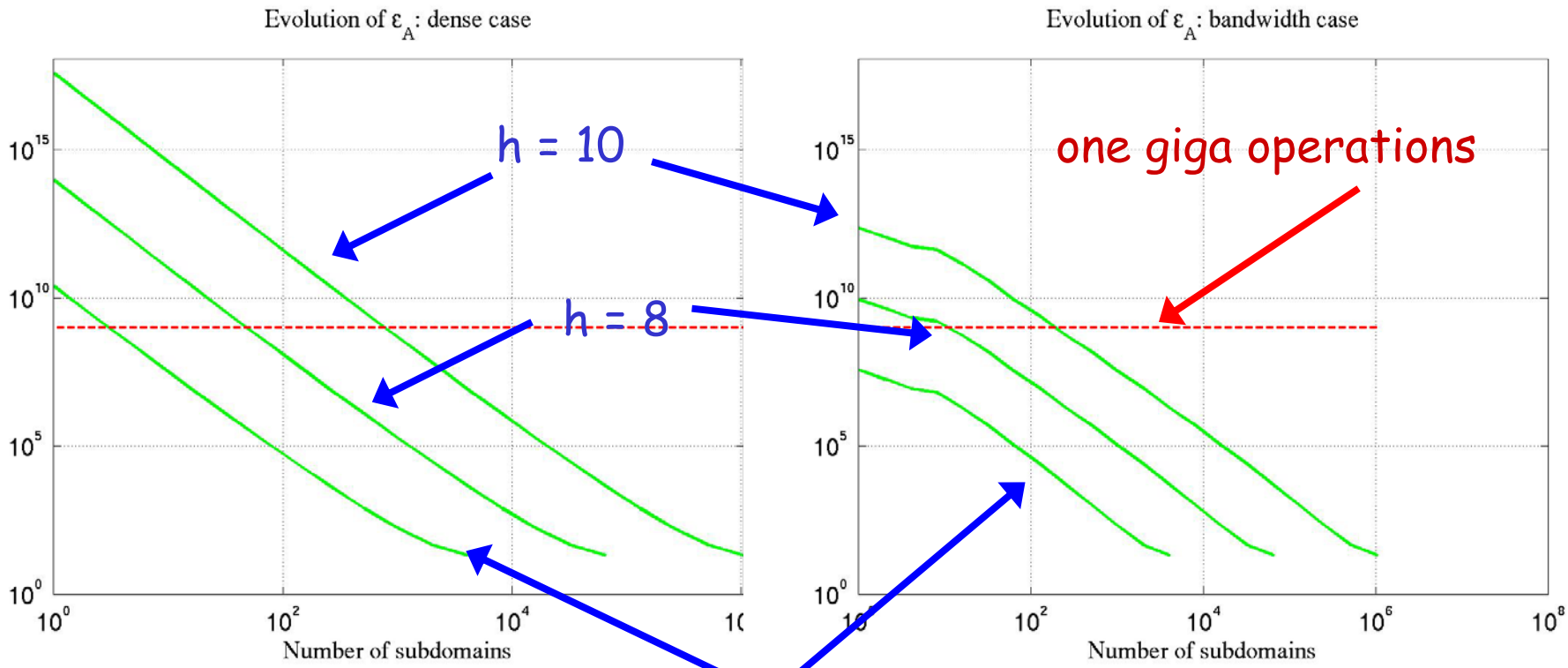
to the next part of the chain

Assumptions

- subdomains are "in place" on processors
- elementary matrices have to be computed prior to be assembled
- we consider here one subdomain per processor

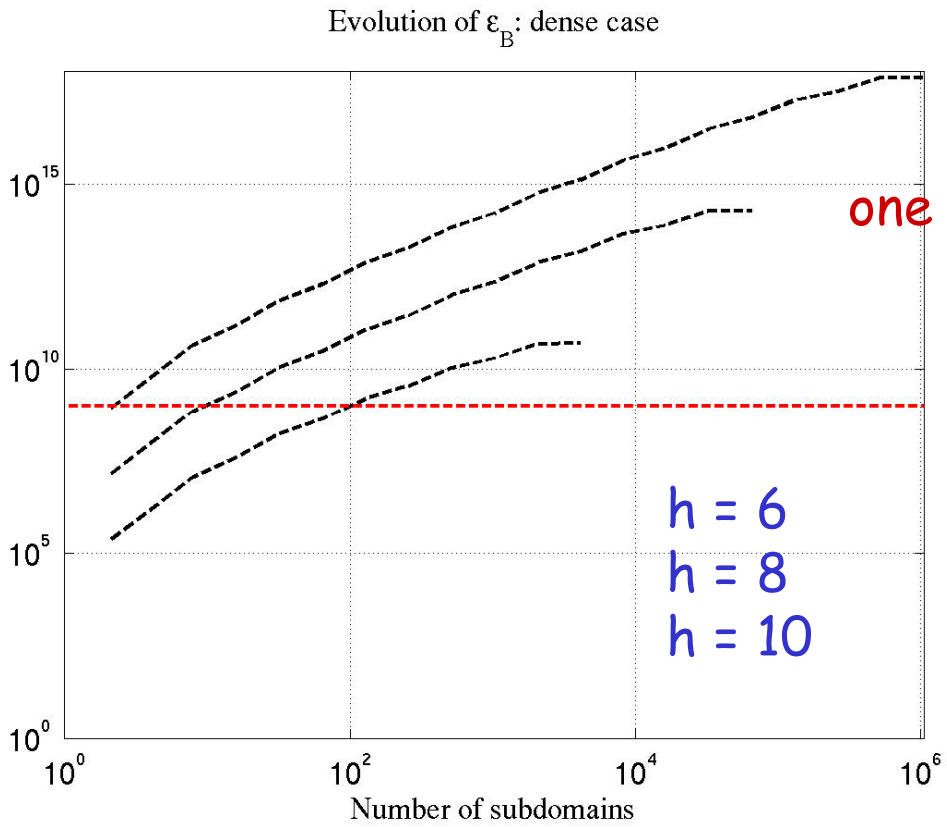
ε_A , ε_B and ε_C are estimations of number of operations explicitely computed with the proposed model

Evolution of ε_A for dense and band



Differences between dense and band models can be shown

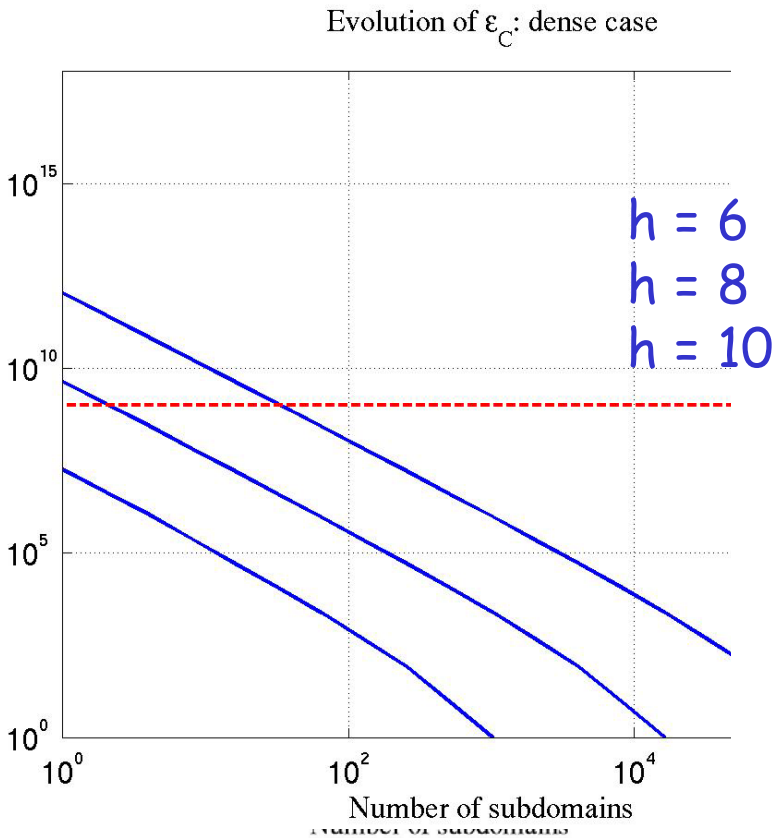
Evolution of ε_B is the same: unique interface matrix is assumed to be dense



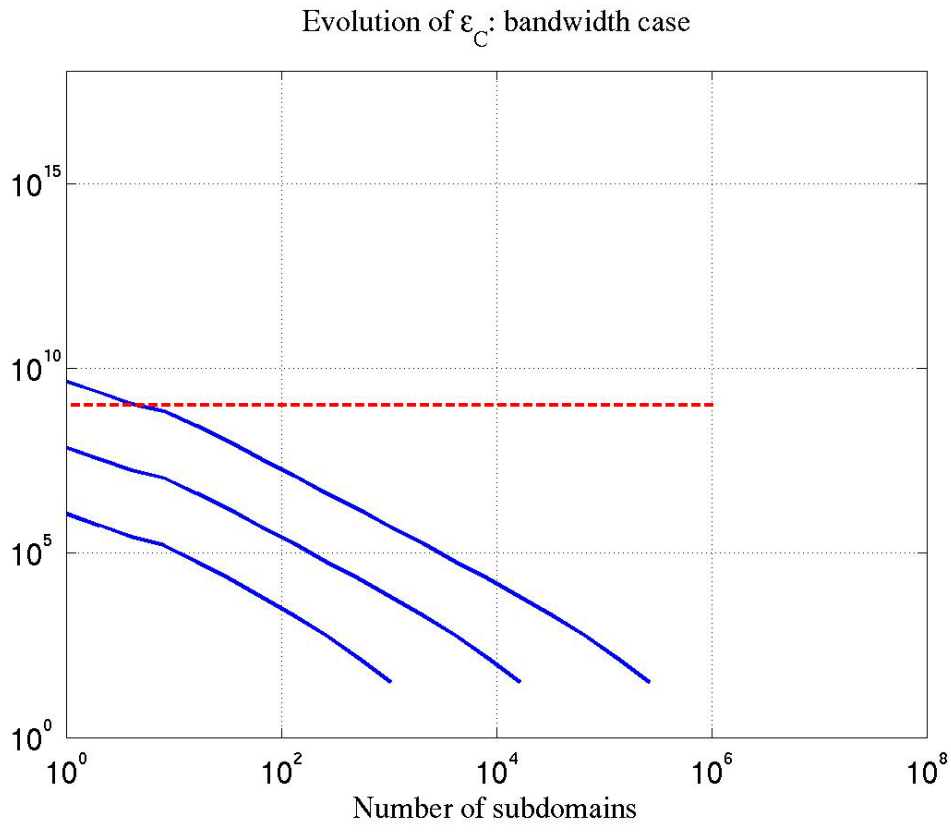
one giga operations

As expected, this cost increases drastically as the number of subdomains.

Evolution of ε_C



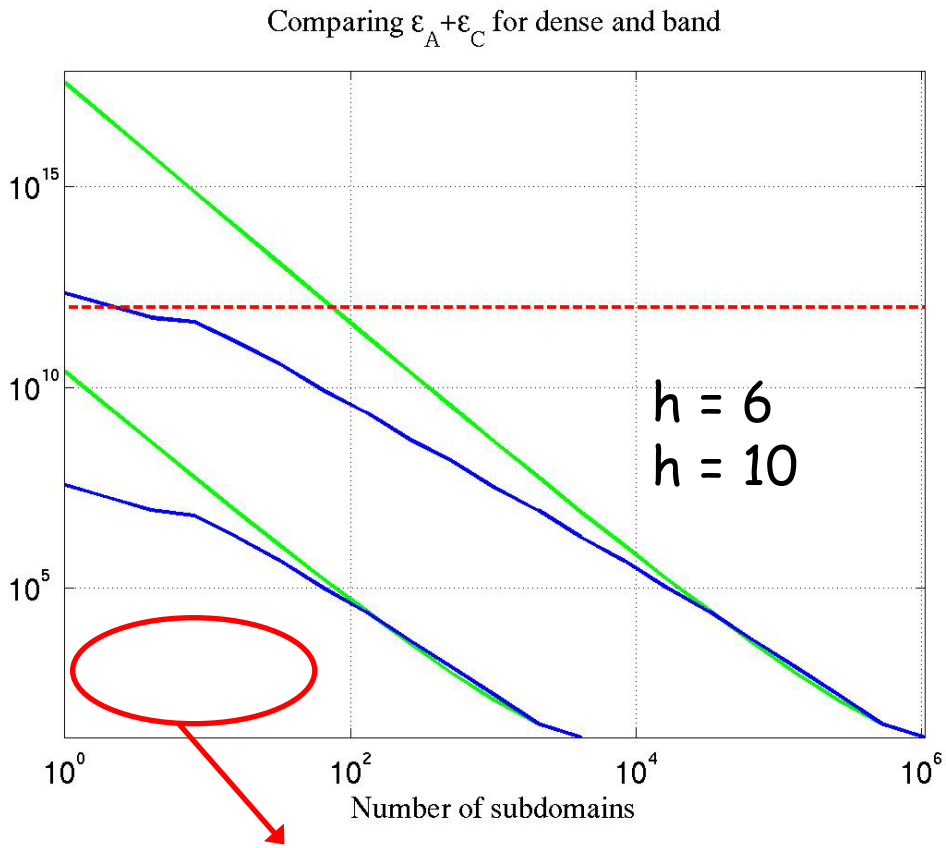
dense



band

Differences between dense and band models can be shown

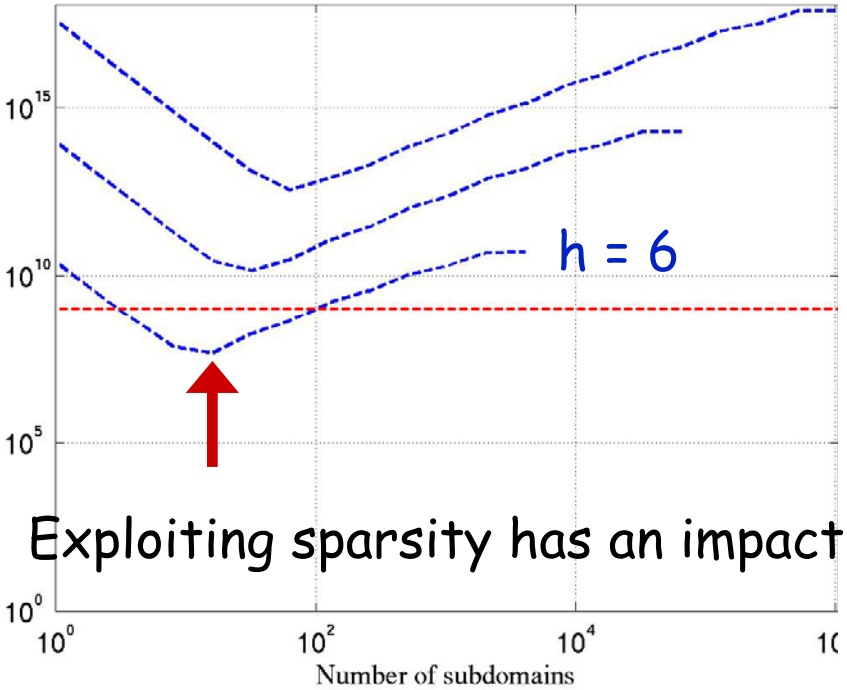
Comparing $\varepsilon_A + \varepsilon_C$ for dense and band



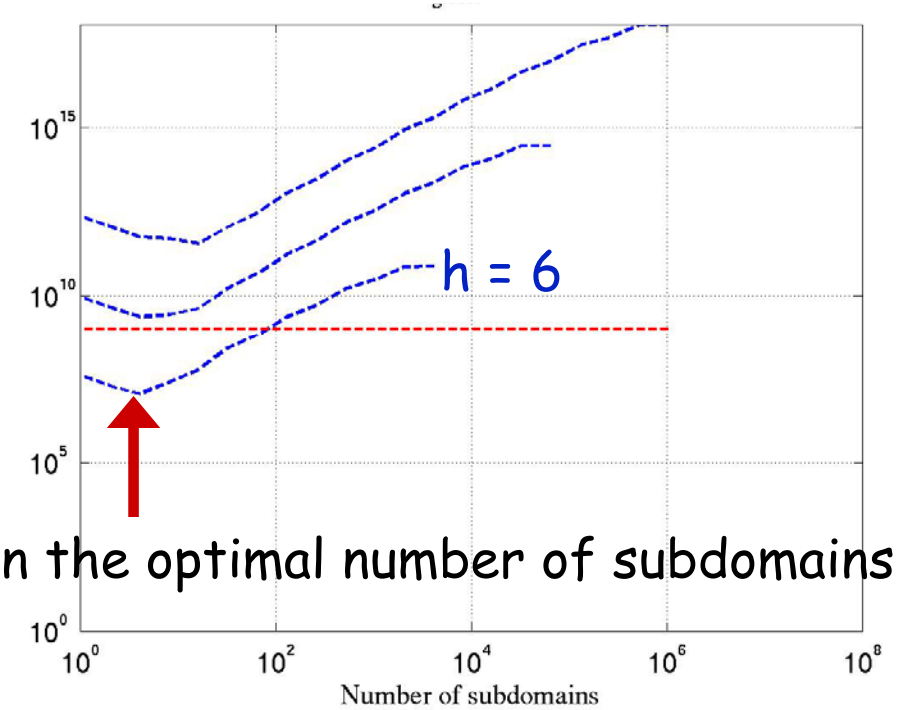
The differences decreases as the number of subdomains:

MUMPS, PaStiX, superLU should be here

Evolution of $\varepsilon_{global} = \varepsilon_A + \varepsilon_B + \varepsilon_C$



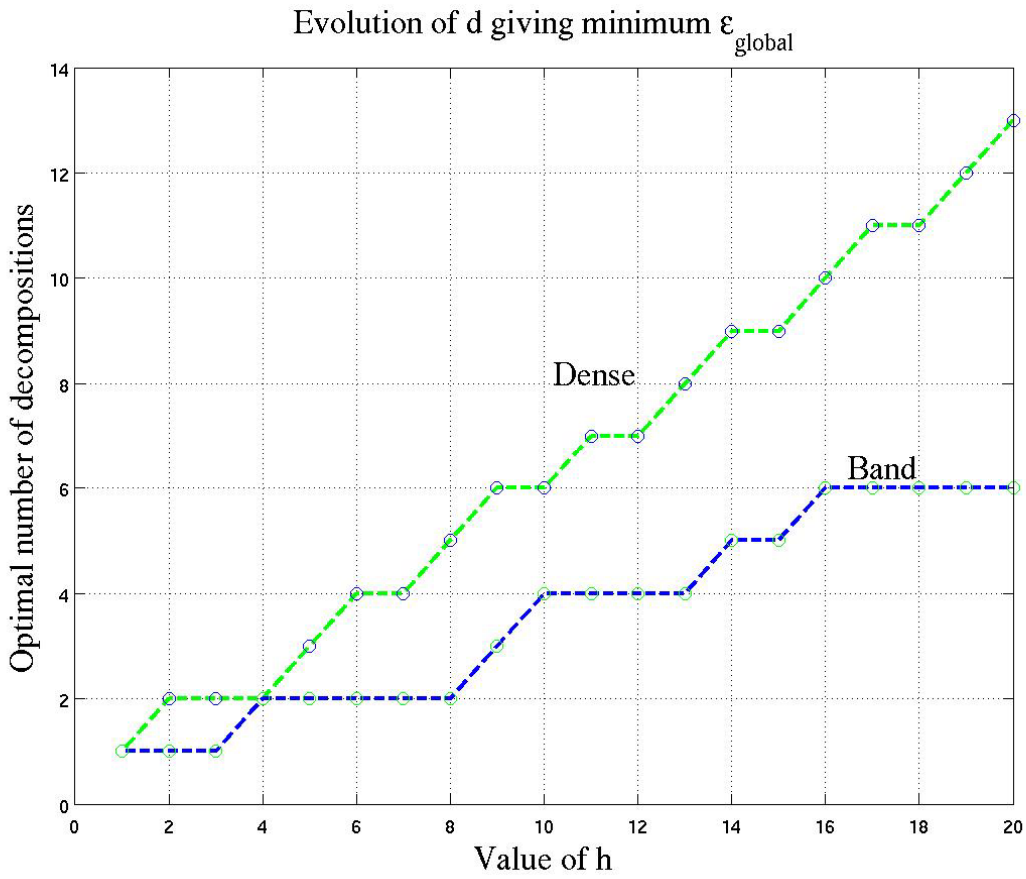
dense



band

Exploiting sparsity has an impact on the optimal number of subdomains

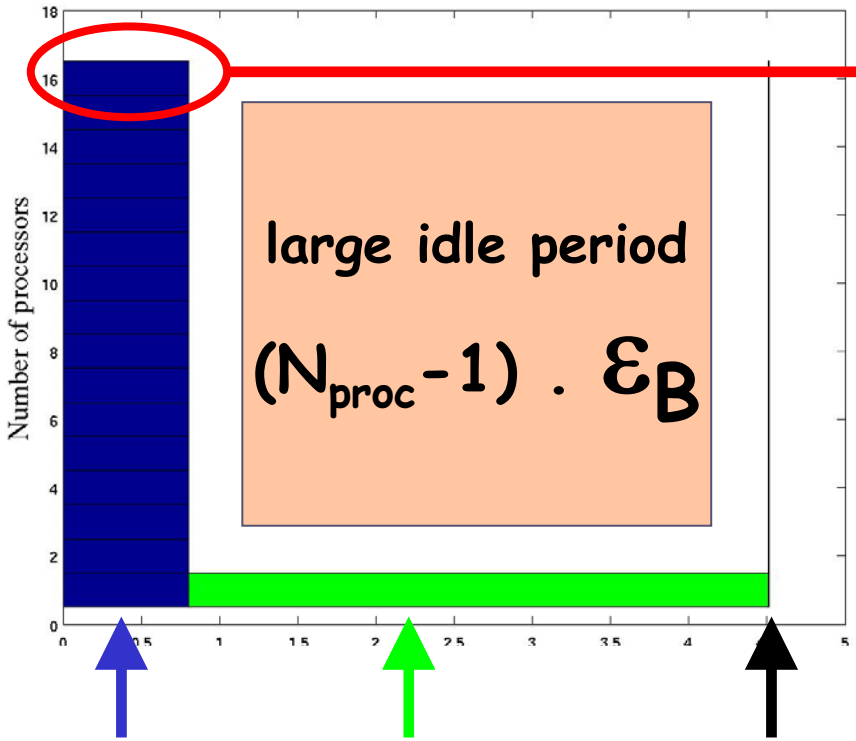
Evolution of d giving the minimum ε_{global}

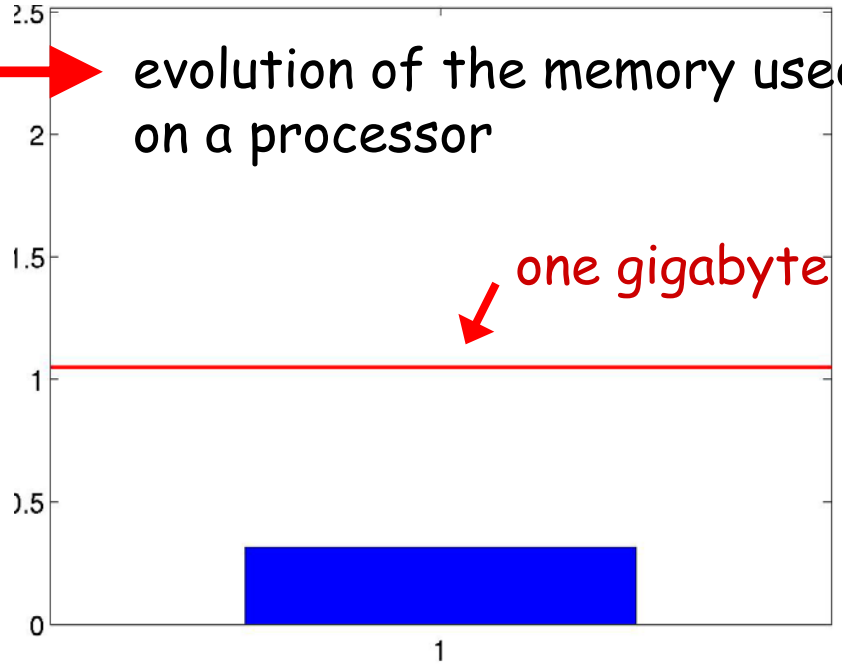


Less decompositions are required to reach optimum in the band case

Schedules for the optimal number of subdomains : one subdomain per processor, dense case

For h = 6, d optimal = 4, N subdomains = 16, N processors = 16



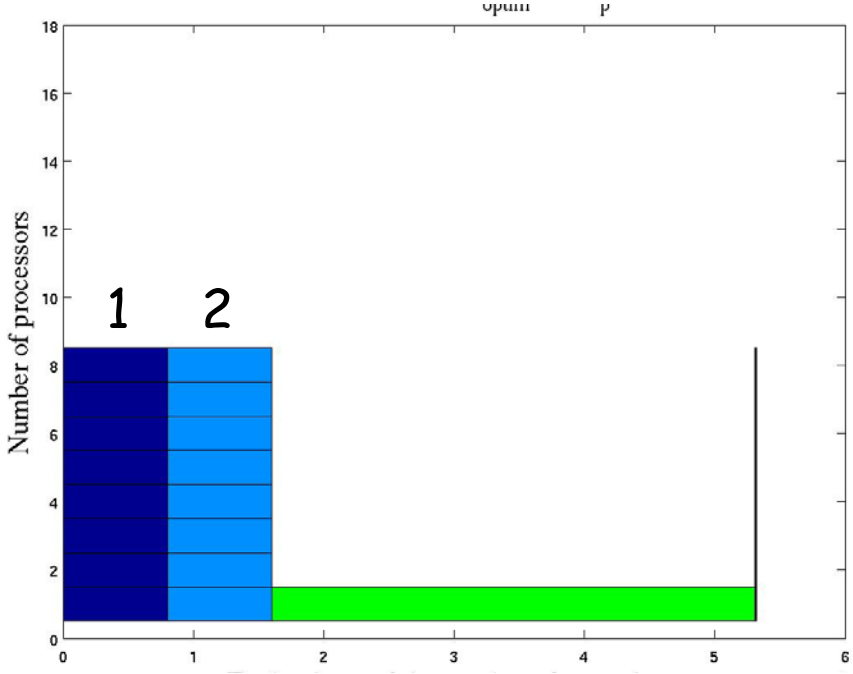


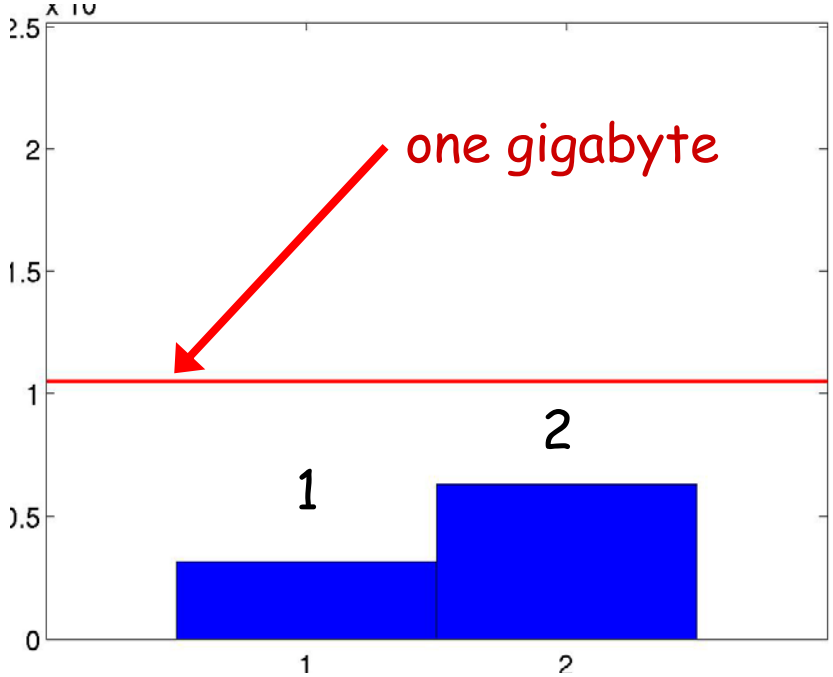
A tasks the B task the C task

We assume that the matrix subdomain is kept in memory

More than one subdomain per processor, dense case

For h = 6, d optimal = 4, N subdomains = 16, N processors = 8



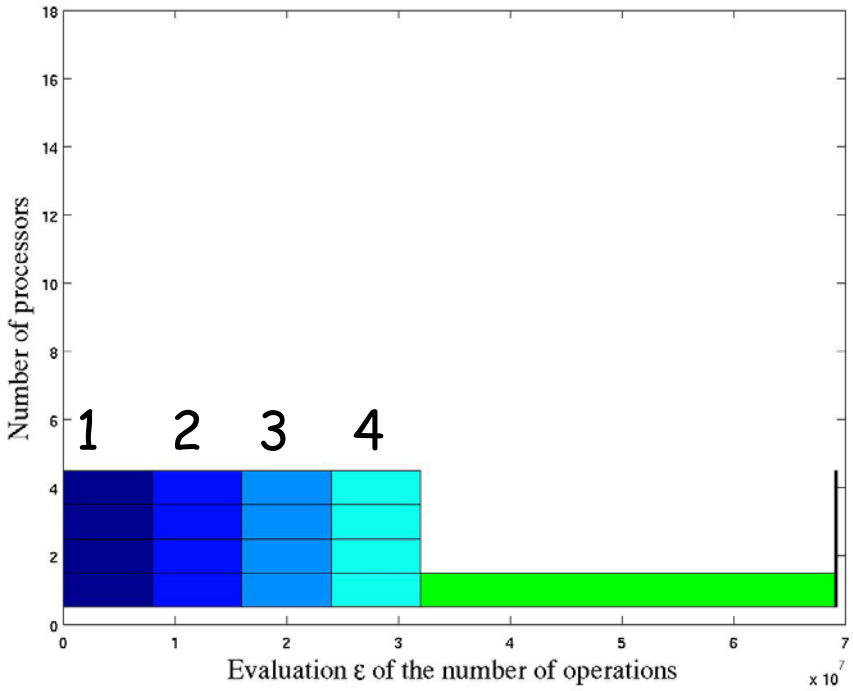


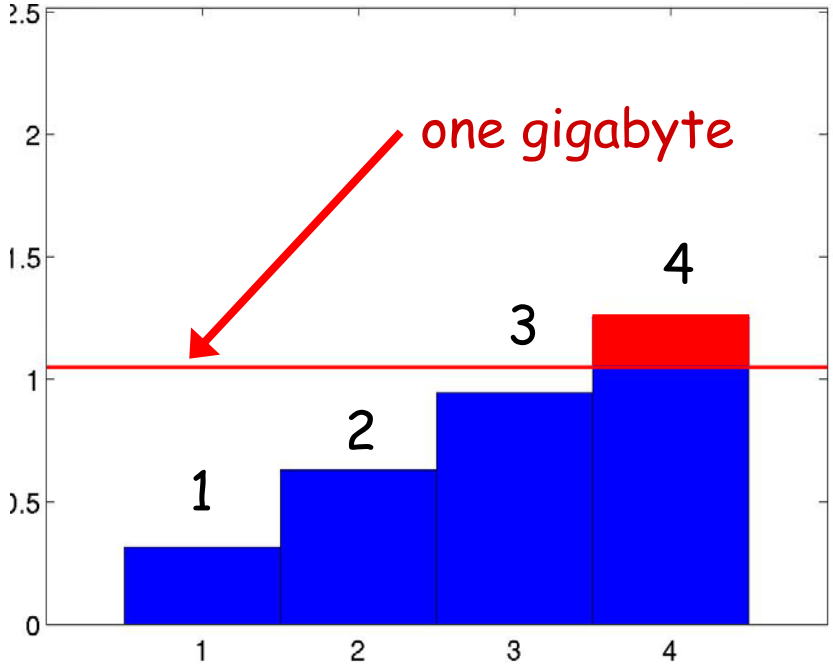
idle period decreases while:

- memory on one processor increases
- computing time increases

More than one subdomain per processor dense case

For h = 6, d optimal = 4, N subdomains = 16, N processors = 4

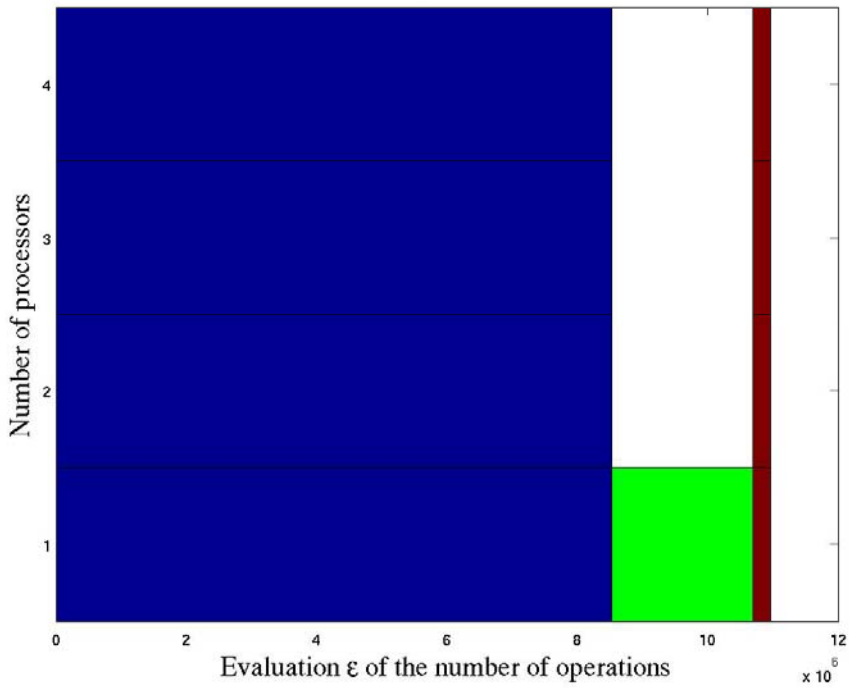


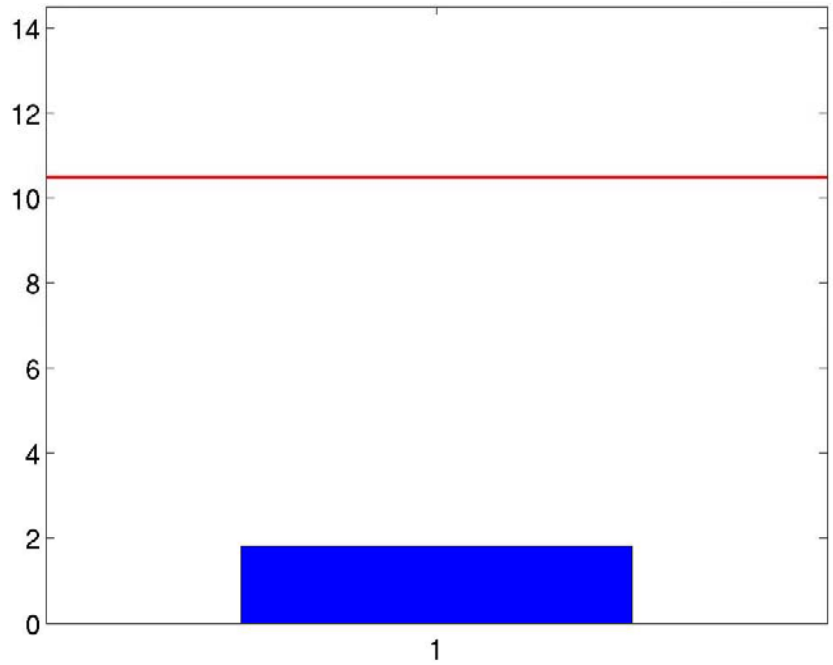


More than one gigabyte is needed in this case

More than one subdomain per processor, band case

For h = 6, d optimal = 2, N subdomains = 4, N processors = 4



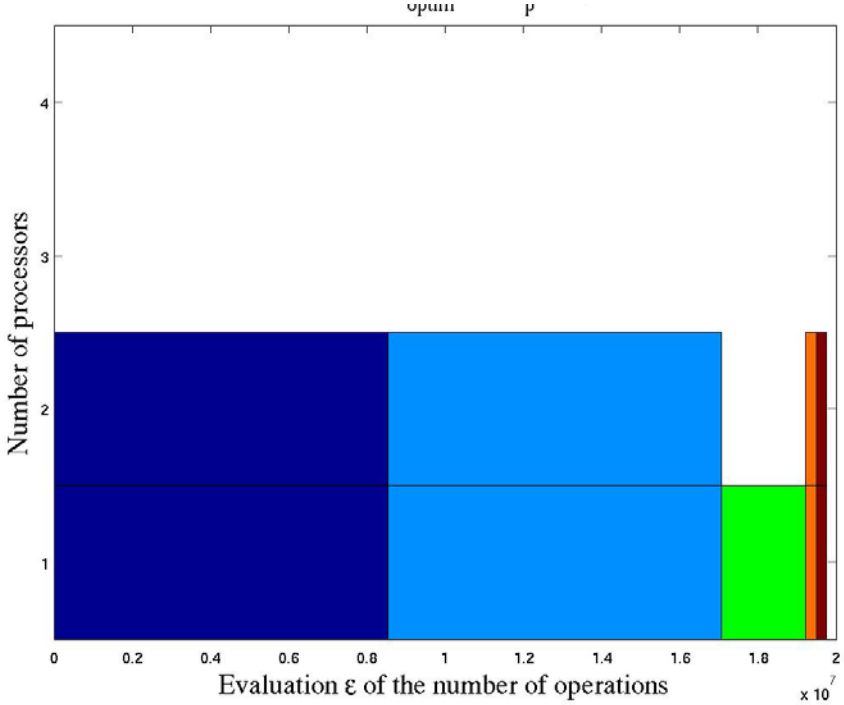


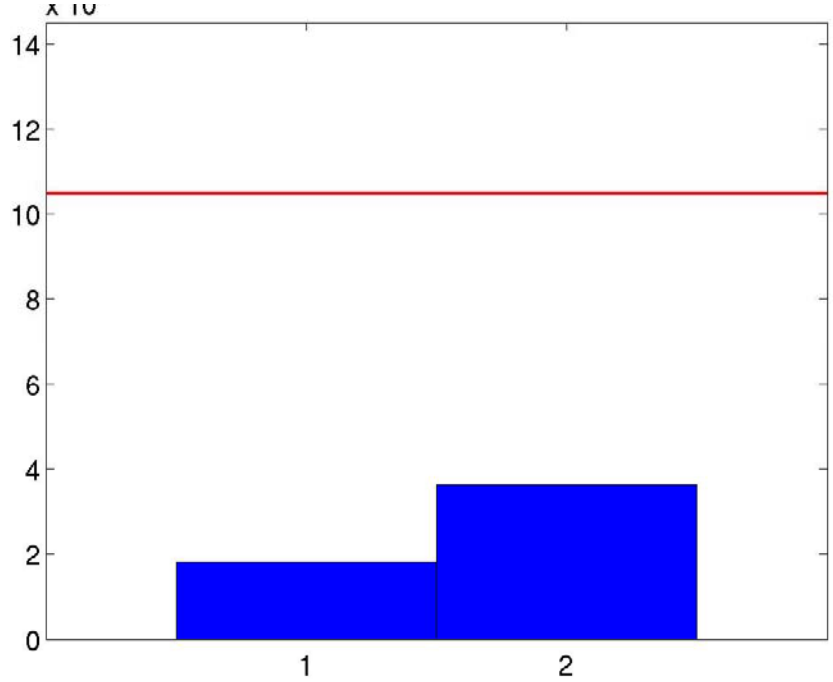
idle period is smaller than in the dense case:

d = 2 rather than 4, so the unique interface problem is smaller

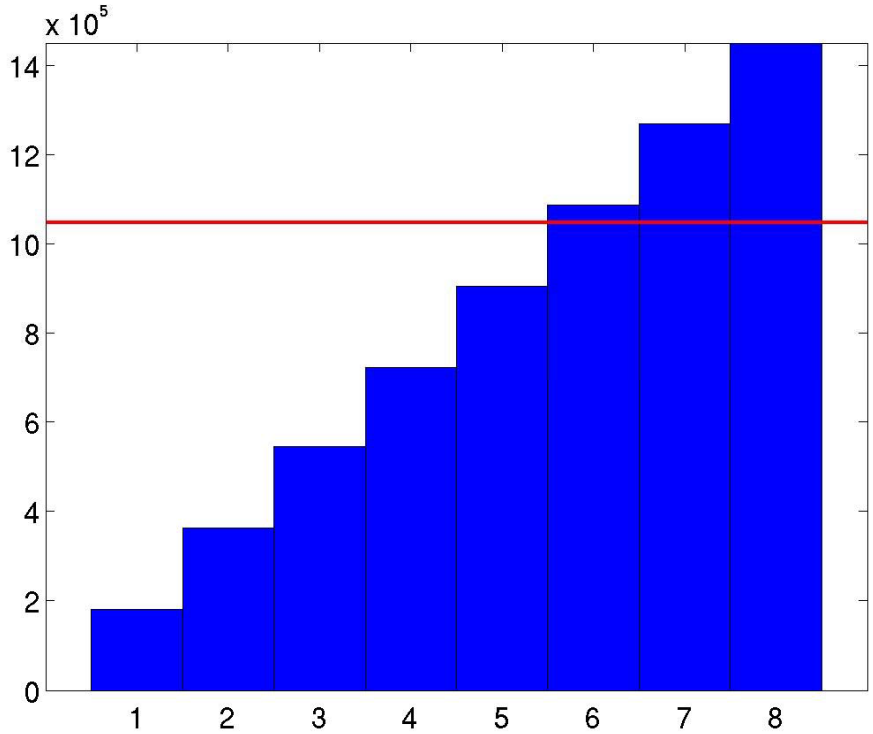
More than one subdomain per processor, band case

For h = 6, d optimal = 2, N subdomains = 4, N processors = 2





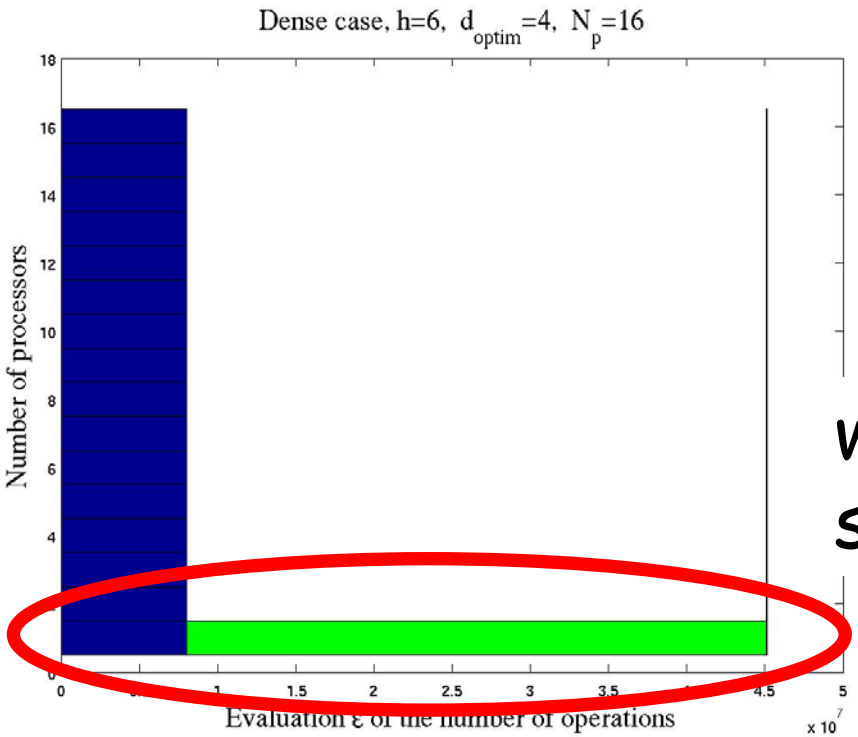
More than one subdomain per processor, band case



Anyway, this simple memory management has limitations.

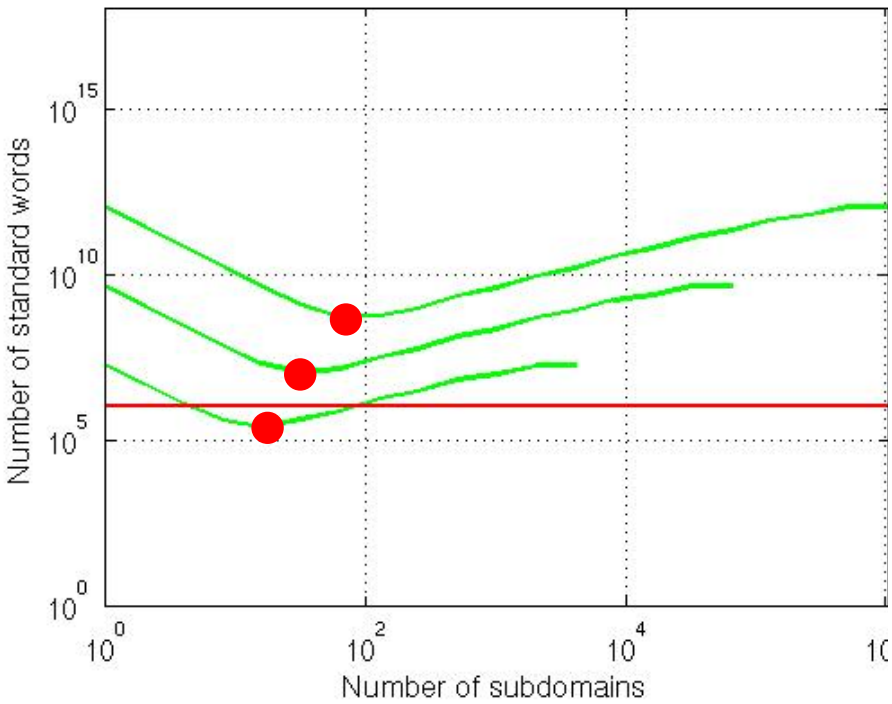
Load/store strategies can be studied but the extra computing costs have to be modelled

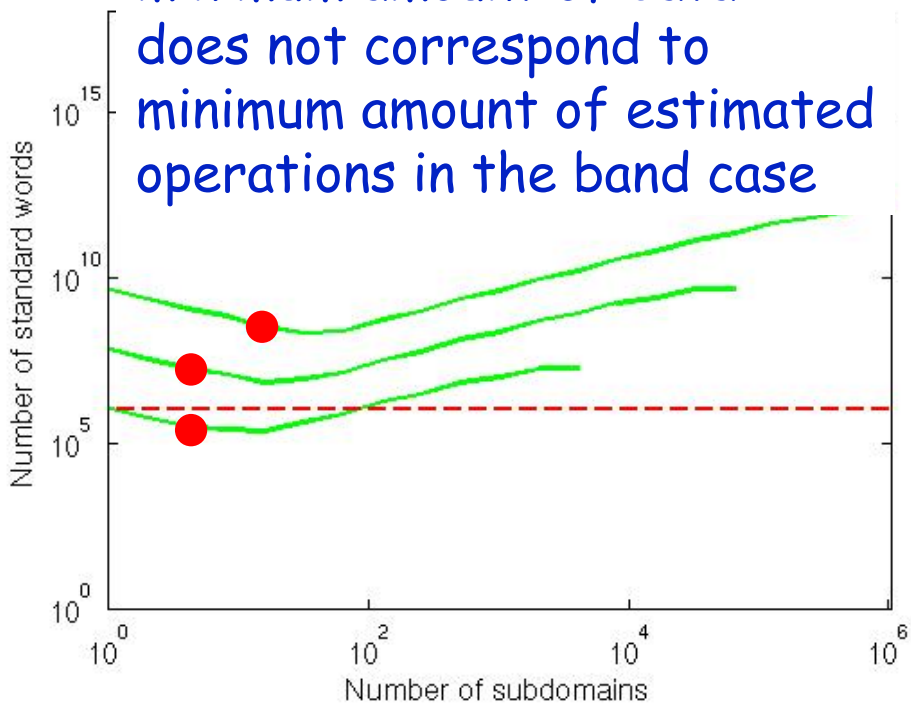
Evolutions of the memory usage on the processor treating the unique interface problem



we consider here one subdomain per processor

Evolutions of the memory usage on the processor treating the unique interface problem, for h=6, 8 & 10

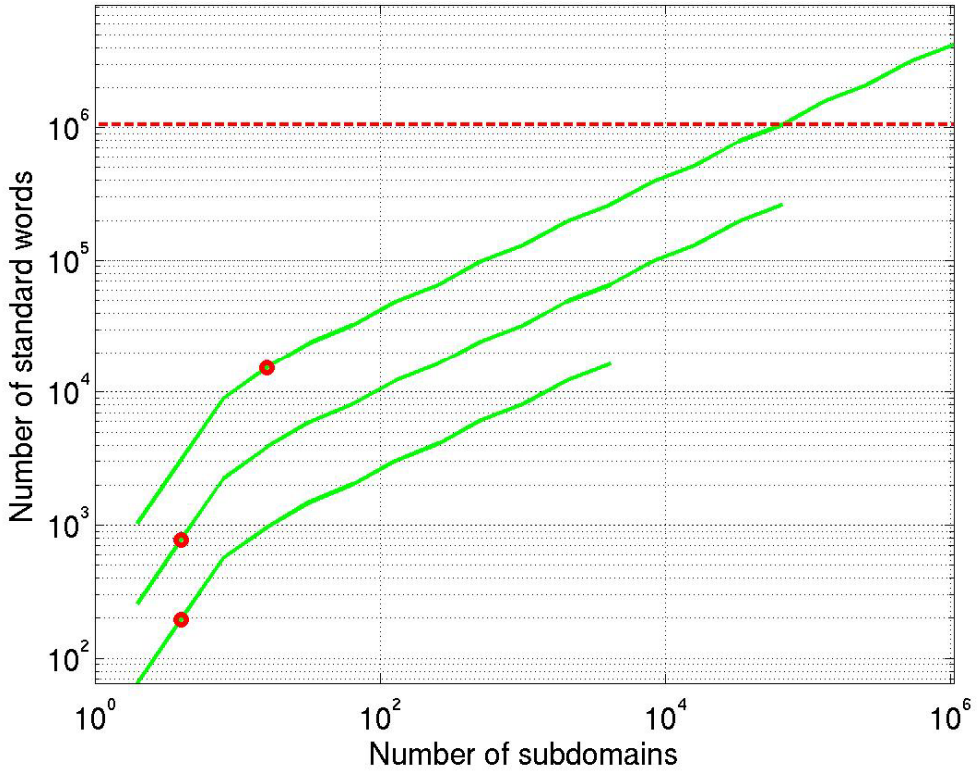




minimum amount of data does not correspond to minimum amount of estimated operations in the band case

Red points show the value of d giving the minimum ε_{global}

Communications



For this simple monolevel strategy, the total amount of communication follows :

$(Ns(d)-1) \cdot (n_{larg\_intf}(d))^2$

That corresponds to (Ns(d)-1) sending of $n_{larg\_intf}(d)$ by $n_{larg\_intf}(d)$ block

We do not focus on communications, however the decomposition model can be used to show them

As expected, this monolevel strategy seems not to be a good candidate using domain decomposition, Schur complement method and direct solver

limitations due to the unique interface problem

Conclusion

- the simple decomposition model permits to compute the parameters of a regular decomposition

- the trends and behaviors of a monolevel strategy have been shown with the model and seems to be in adequation with what we already know about this simple strategy

- multilevel strategies can be modelled (computations, memory management, communications)

I suffer from tinnitus