Méthodes et programmation numériques avancées

Master CHPS, parcours IHPS
2016-2017

Nahid.Emad@uvsq.fr

OBJECTIFS DE L'UE

Permettre de maîtriser

- le principe des méthodes numériques modernes,
- leur adaptation à de nouvelles architectures parallèles et distribuées (massivement parallèles et fortement hiérarchiques),
- leurs développements et optimisations en termes de temps d'exécution, d'espace mémoire, de précision, de communication et de consommation énergétique.
- Développer l'aspect *BigData* des problèmes étudiés (algèbre linéaire en général).

Principaux éléments

- Méthodes itératives pour problèmes de grande taille
- Méthodes hybrides synchrones/asynchrones
- Méthodes numériques d’algèbre linéaire pour le traitement de masses de données (*Big Data)*
- Méthodes de compression des structures creuses.
- Modèles de programmation graphe de tâches, PGAS
- Métriques de performances

PLAN DE COURS

1. Introduction
2. Quel paradigmes de programmation?
3. Approche « *unir & conquérir* »
4. Méthodes: Multiple Krylov Subspace (MKS)
5. Quelques résultats expérimentaux
6. Zoom sur certaines méthodes itératives pour des problèmes d'AL de grande taille (TD/TP)
7. Zoom sur l'hybridation de ces méthodes
8. Programmation orientée graphes

PLAN DE COURS

1. Introduction
2. Quel paradigmes de programmation?
3. Approche « *unir & conquérir* »
4. Méthodes: Multiple Krylov Subspace (MKS)
5. Quelques résultats expérimentaux
6. Zoom sur certaines méthodes itératives pour des problèmes d'AL de grande taille (TD/TP)
7. Zoom sur l'hybridation de ces méthodes
8. Programmation orientée graphes

1.Introduction (suite)

La barrière du calcul Exascale…

- Le prix Gordon Bell: performances soutenues > 14 Petaflops pour de la simulation in Mécanique des fluides (en 2013)

  2016: *“10M-Core Scalable Fully-Implicit Solver for Nonhydrostatic Atmospheric Dynamics.”*

- Nouvelle frontière: **Calcul Exascale** (combien MWatts? – *TOP1 du 11/16: >15*MW)
- Beaucoup de défis émergent (>125Petaflops pour le TPO1: Sunway TaihuLight – Sunway)

1.Introduction (suite)

Les défis…

Characteristics	Tianhe-2	Grand Challenge	Targeted Improvements
System peak	55 Pflops	1+ Eflop/s	Approx. 20x
Power consumption	17.6 MW (2 Gflops/W)	< 20 MW (50 Gflops/W)	Approx. 15x
Memory	1.4 PB	32+ PB	Approx.. 50x
Node Concurrency	24 cores CPU + 171 cores CoP	O(1K) or 10K	Approx. 5x – 50x
Node Interconnect	6.36 GB/Sec	200-400 GB/Sec	Approx. 40x
System Size (nodes)	16,000	O(100,000) – O(1M)	Approx. 6x – 60x
Total system concurrency	3.12 M	O(billion)	Approx. 100x
MTTF	Few/Day	Many/Day	O(?)

Dongarra 2014.

1.Introduction (suite)

La barrière du calcul Exascale…

- *Anticiper les solutions* et former des scientifiques pour les calculs futurs
- Prendre en compte les architectures récentes et émergentes ainsi que des prototypes pour concevoir les futurs langages, systèmes, algorithmes, …
- Proposer de nouveaux paradigmes de programmation (SPMD/MPI pour 10^6 cœurs et 10^9 threads?)
- Penser co-design et proposer des langages orienté-application (*DSL*) et/ou des langages, environnements, … *haut niveau et multi-niveaux*

1.Introduction (suite)

Quelles (nouvelles) méthodes pour les futurs supercalculateurs?

1.Introduction (suite)

- Conception de nouvelles méthodes numériques pour ces architectures … résultant à de nouvelles architectures (**co-design**): adaptation aux architectures, arithmétiques, I/O, des latences, etc. résultant à une approche générale d'**auto-tuning**
- **Hybrider** des méthodes numériques pour la résolution de grandes applications scientifiques de manière asynchrone avec chacune d'elle auto-adaptée
- Rechercher des critères fin, même au niveau d'application et/ou méthode mathématique, pour auto/smart-tuning avec l'intégration d'expertise des utilisateurs finaux.

Introduction

Notre objectif est de mettre au point un paradigme de programmation pour le calcul à grande échelle

et d'en déduire l'introduction des

méthodes numériques modernes et « intelligentes » comme celles de Krylov hybrides smart-tunées.

PLAN DE COURS

1. Introduction
2. Quel paradigmes de programmation?
3. Approche « *unir & conquérir* »
4. Méthodes: Multiple Krylov Subspace (MKS)
5. Quelques résultats expérimentaux
6. Zoom sur certaines méthodes itératives pour des problèmes d'AL de grande taille (TD/TP)
7. Zoom sur l'hybridation de ces méthodes
8. Programmation orientée graphes

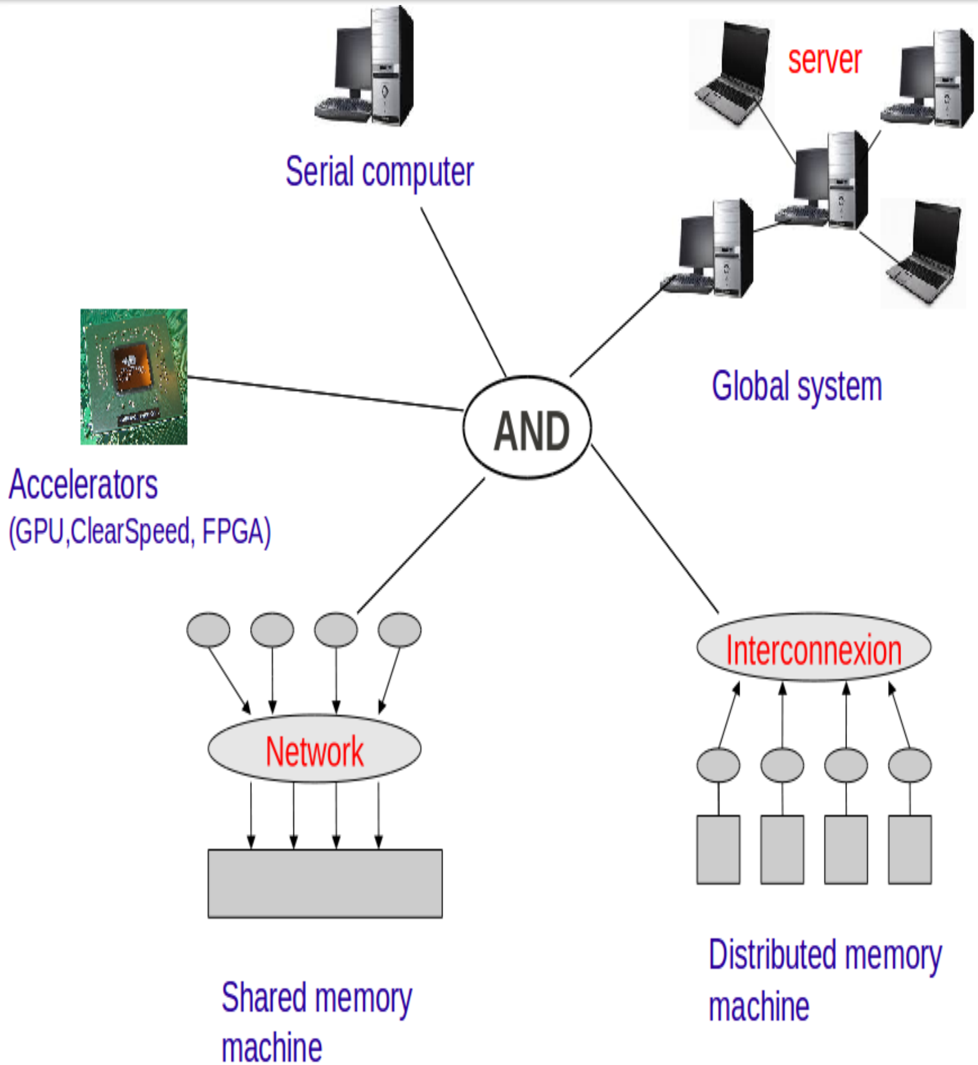
Quel model de programmation?

Eléments clés:

- Minimiser le mouvement des données (attention au coût des communications)
- Consommation énergétique
- Parallélisme multi-niveaux (mémoire, processeur, etc.)
- Ordonnancement multi-niveaux
- Equilibrage de charge mult-niveaux
- Tolérance aux panes

Programing paradigm: component approach

Making (re)use of existing libraries, etc. in the context of extreme scale computing +



Component approach

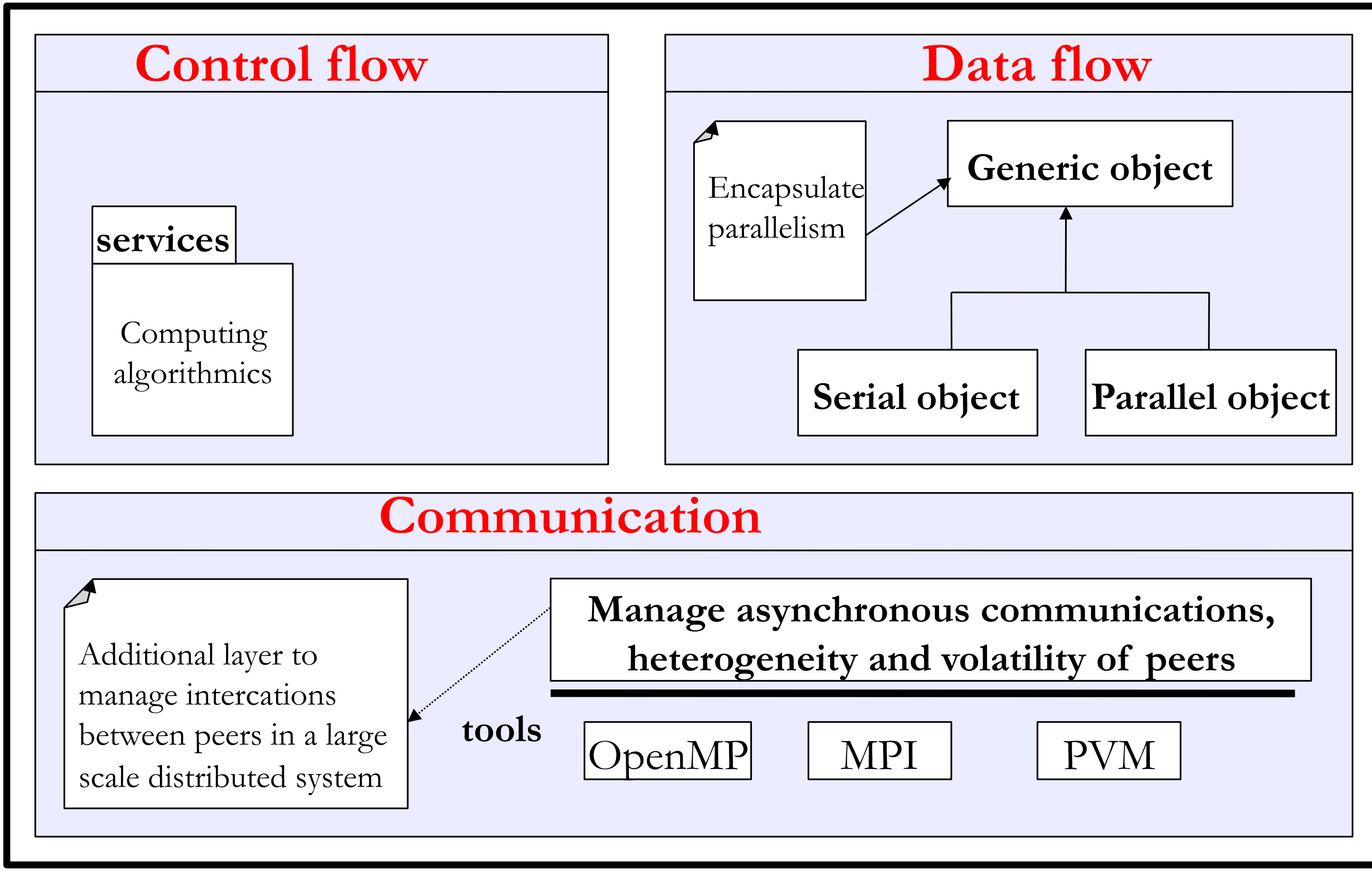
- ✓ Interoperability
- ✓ Reusability
- ✓ Durability

Programing paradigm: graph of tasks

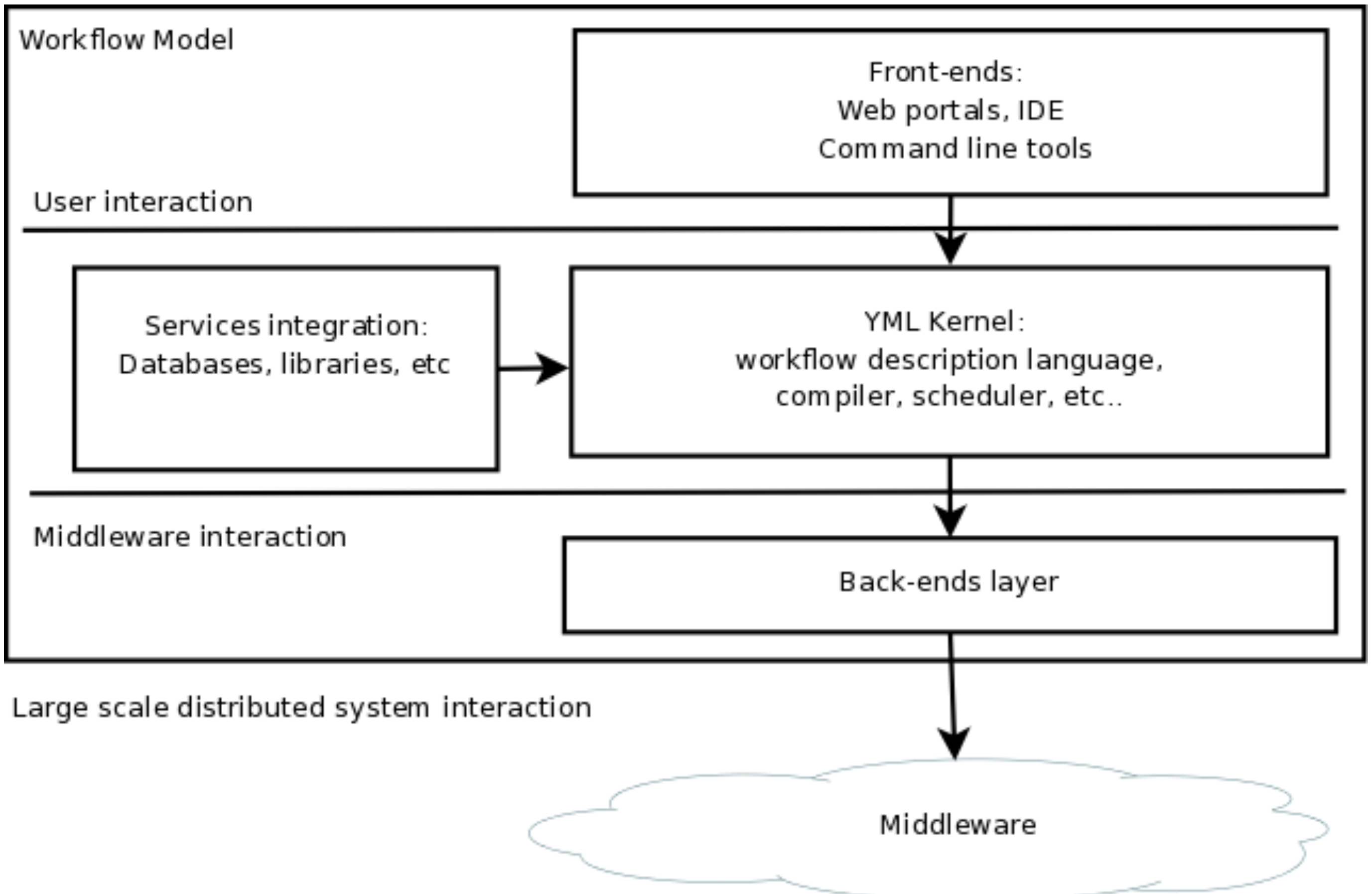
- Graph of very coarse grain components/tasks (data flow oriented/SPMD or PGAS-like or data-parallelism)
  - ✓ limitation of communications to the cores allocated to such components
- A component can be itself a graph of tasks and can be descripted by SPMD PGAS-like model
  - ✓ on each processor, we can program accelerators,
  - ✓ on each core multithreaded optimization can be used
- Users have to be able to give expertise to middleware, runtime system & schedulers

YML: An adapted language and environment

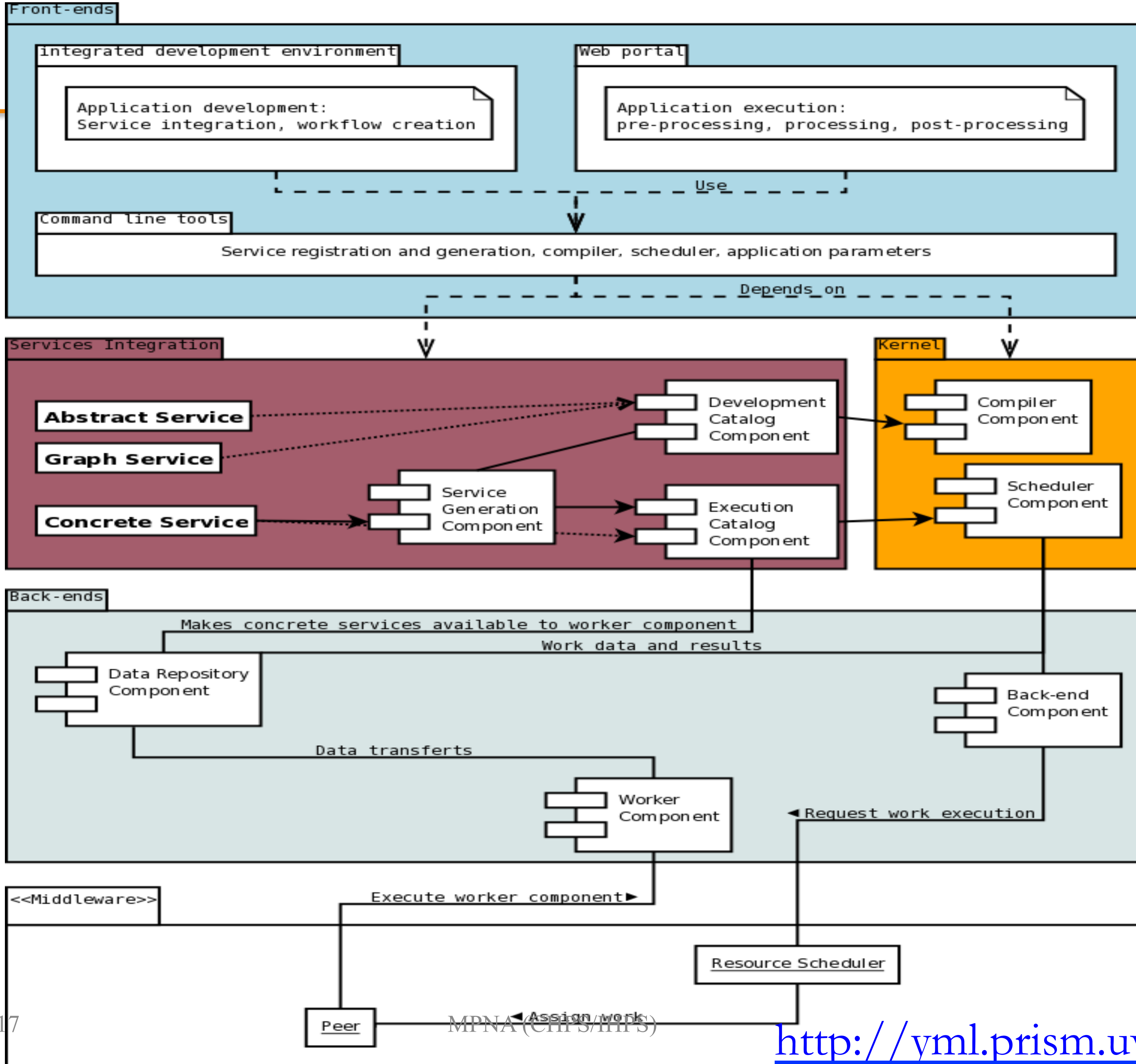
Programming model for sustainable numerical library



Framework model



YML



http://yml.prism.uvsq.fr/

Which characteristics for numerical methods?

Main characteristics based on proposed programing paradigm:

- Avoiding synchronous communication (such as large scalar products, overall synchronization)
- Promoting asynchronicity
- Taking into account heterogeneity
- Introducing fault tolerance
- Encouraging load balancing possibility
- Introducing auto/smart tuning

PLAN DE COURS

1. Introduction
2. Quel paradigmes de programmation?
3. Approche « *unir & conquérir* »
4. Méthodes: Multiple Krylov Subspace (MKS)
5. Quelques résultats expérimentaux
6. Zoom sur certaines méthodes itératives pour des problèmes d'AL de grande taille (TD/TP)
7. Zoom sur l'hybridation de ces méthodes
8. Programmation orientée graphes

Problem to solve

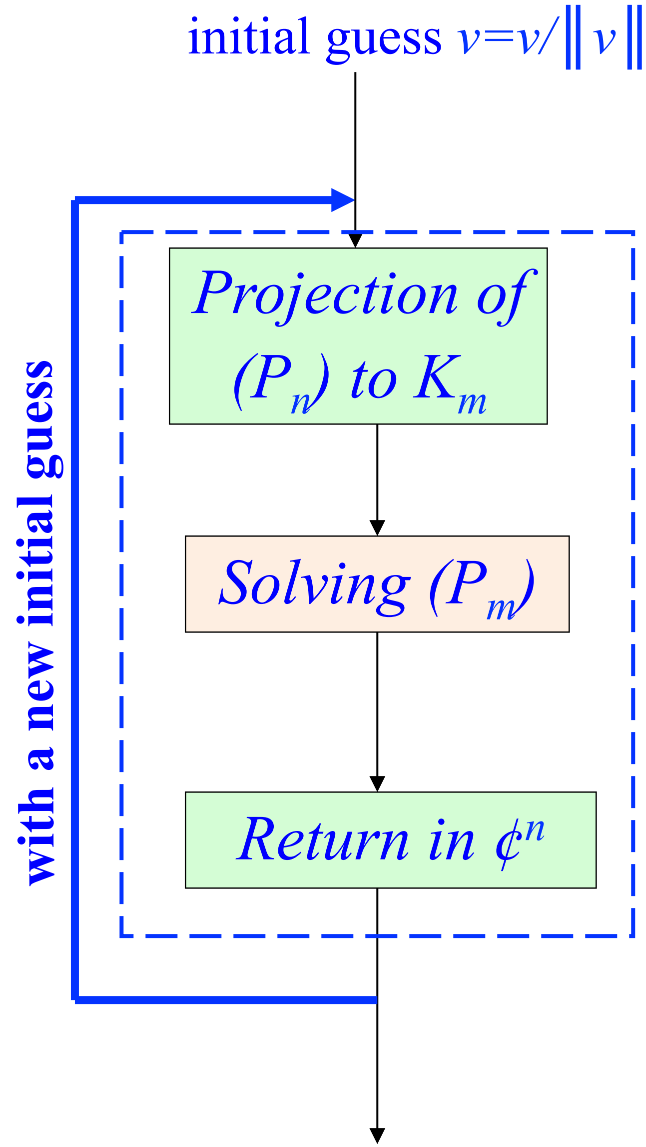
Let A be a very large and sparse matrix

(P_n)

$A \in \mathbb{C}^{nxn}$, $b \in \mathbb{C}^{nxn}$ seek $x \in \mathbb{C}^n$: $Ax=b$
$A \in \mathbb{C}^{nxn}$, seek a few k Ritz pairs $\Lambda_k = (\lambda_1, \ldots, \lambda_k) \in \mathbb{C}^k$ and $U_k = (u_1, \ldots, u_k) \in \mathbb{C}^{nxk}$: $Au_i = \lambda_i u_i$ $(i=1,\ldots,k)$

Restarted Krylov subspace methods

(P_m) *is the* projection of (P_n) in $K_m(A,v)=span(v, Av, \ldots, A^{m-1}v)$



Arnoldi projection/Factorization

$AR(input:A,m,v;output:\ H_m, V_m)$

For $j=1,\ldots,m$ do:

$h_{i,j}=(Av_j, v_i)$, for $i=1,\ldots,j$

$z_j=Av_j-\sum^{j}_{i=1}h_{i,j}v_i$

$h_{j+1,j}=\|z_j\|$

$v_{j+1}=z_j/\ h_{j+1,j}$

Gramm-Schmidt orthogonalization process

$$\boxed{AV_m=V_mH_m+f_me^T_m}\ \text{with}\ \boxed{f_m=h_{m+1,m}v_{m+1}}$$

Eigenproblem in the subspace

$H_my_i=\lambda_iy_i$ *for i=1, …m* (P_m)

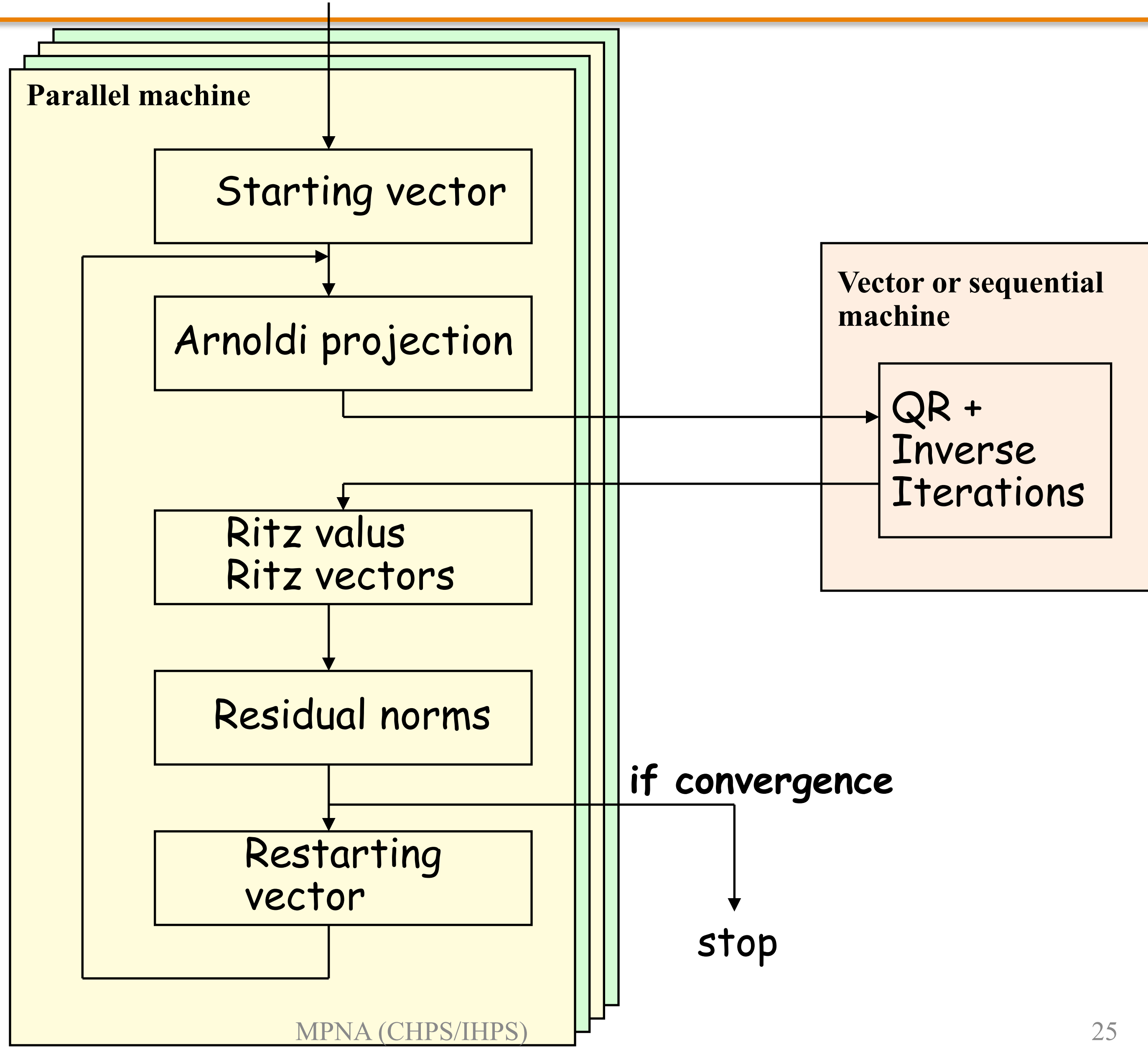
Problem in KSM: $K_m = K(A, v)$

<u>Saad's solution (ERAM)</u> For a fixed small size m, improve $K(A,v,m)$ by updating **explicitly** v and restart the process

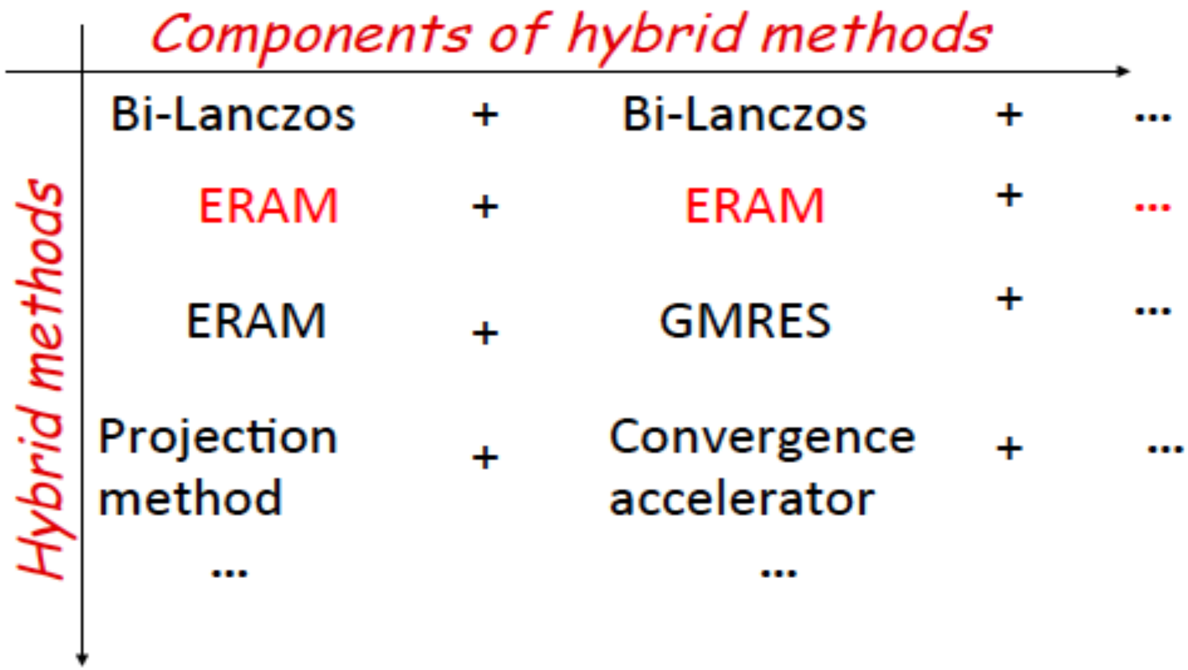
<u>Sorrensen's approach (IRAM)</u> For a fixed small size m, improve $K(A,v,m)$ by updating **implicitly** v and restart the process

Problem with clustered eigenvalues

Parallel restarted KSM



Unite and conquer approach



Saad (Chebyshev acceleration techniques for solving nonsymmetric eigenvalue; 1984), *Brezinski* (hybrid procedures for solving linear systems; 1994), Code coupling (in simulation), ...

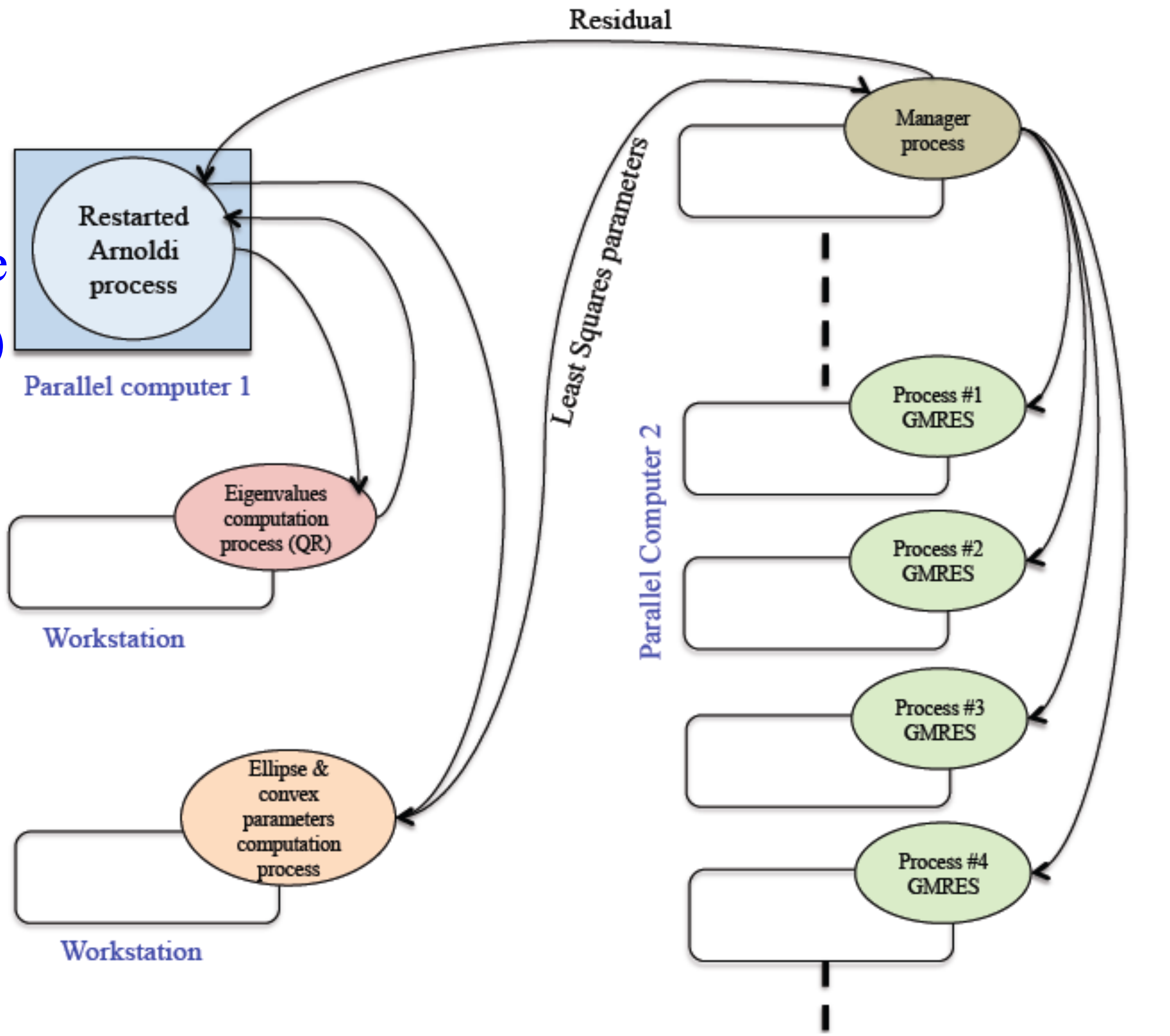
Unite and conquer approach

UCA applied to Krylov subspace methods

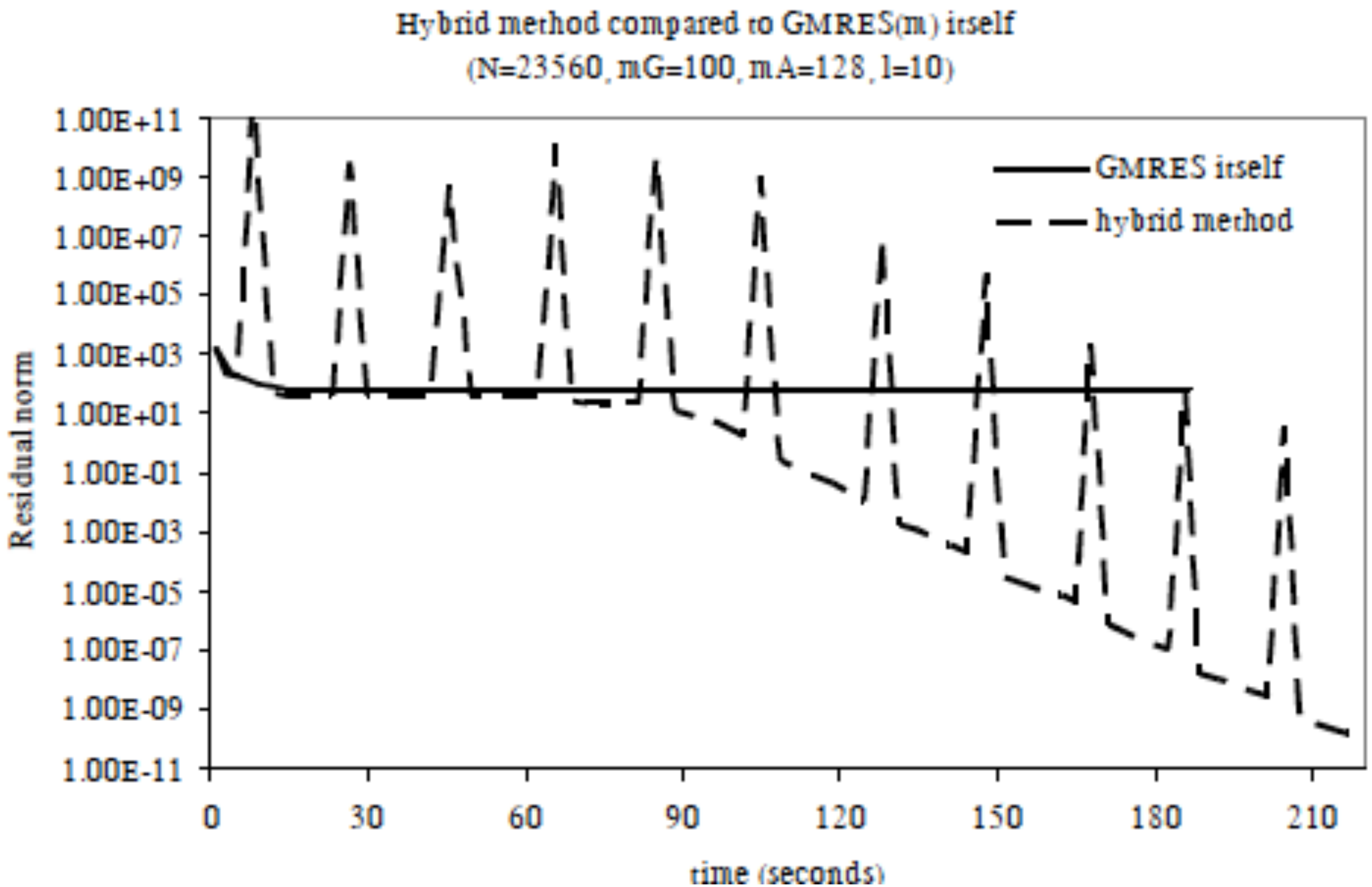
- GMRES
- Least Square
- Convex computing
- ERAM ou IRAM

GMRES/LS-Arnoldi for linear system

- Multi level parallelism (coarse grain & fine grain)
- Asynchronous communication
- Fault tolerance
- Load balancing

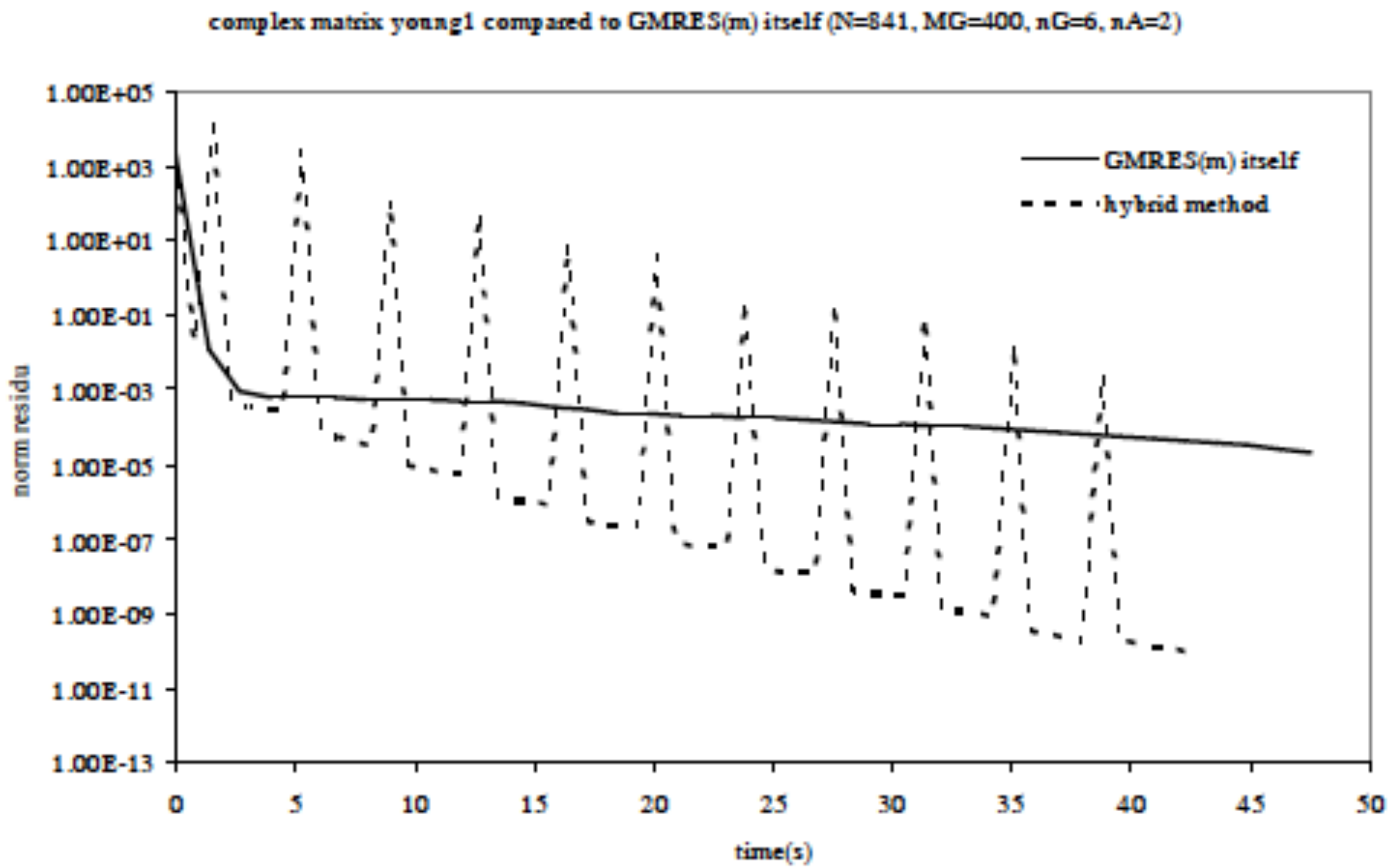


GMRES/LS-Arnoldi on SP3, SP4 & workstations



Haiwu He, C. Bergere, S. Petiton,. A Hybrid GMRES/LS-Arnoldi Method to Accelerate the Parallel Solution of Linear Systems. Computers and Mathematics with Applications 51 (2006) 1647-1662.

GMRES/LS-Arnoldi on GRID'5000

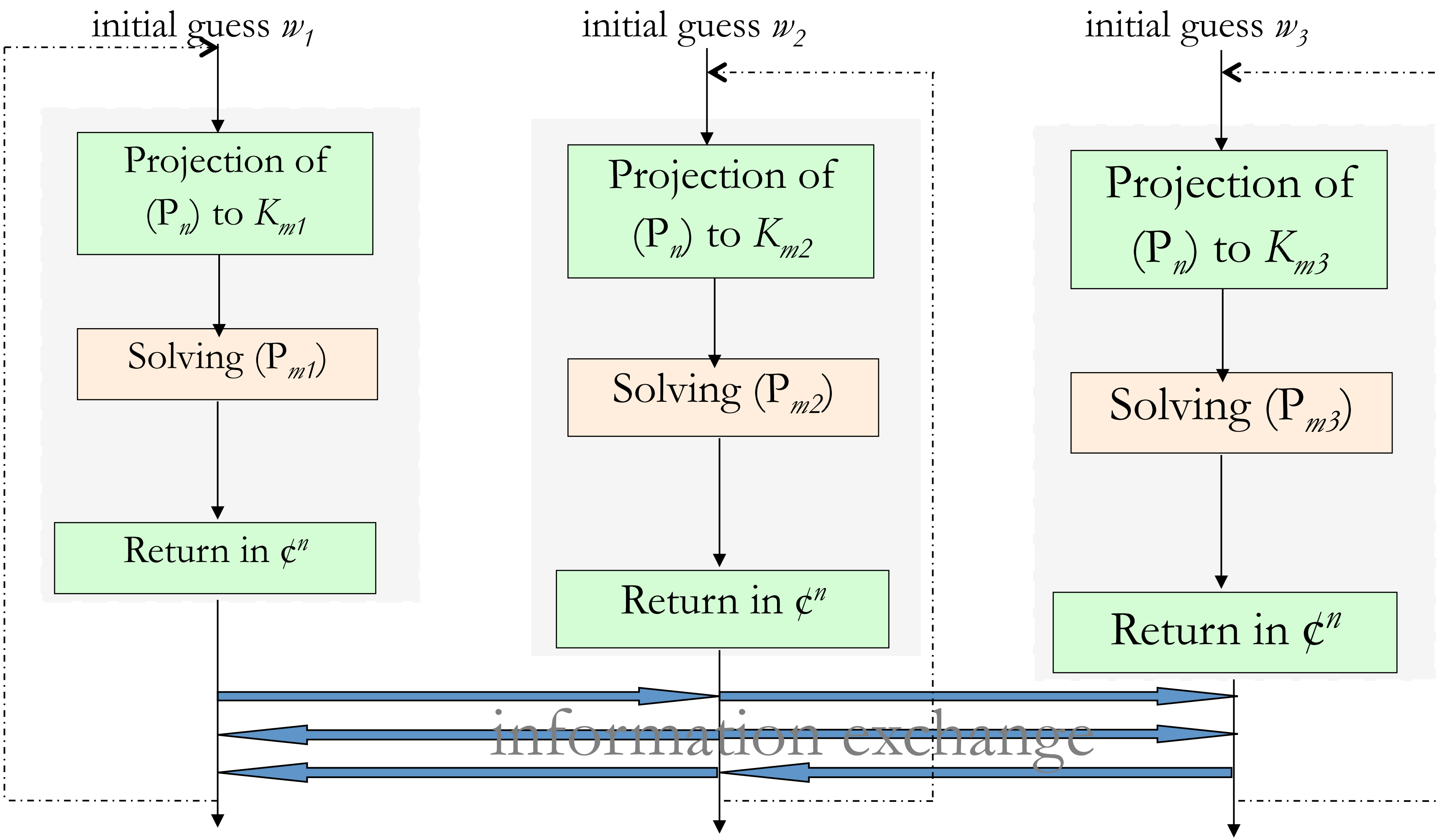


The matrix is symmetric complex (young1c of size 841).
On GRID'5000: GMRES on Nancy & Orsay sites, restarted Arnoldi process on Nancy & Orsay, and Least Square on Bordeaux

PLAN DE COURS

1. Introduction
2. Quel paradigmes de programmation?
3. Approche « *unir & conquérir* »
4. Méthodes: Multiple Krylov Subspace (MKS)
5. Quelques résultats expérimentaux
6. Zoom sur certaines méthodes itératives pour des problèmes d'AL de grande taille (TD/TP)
7. Zoom sur l'hybridation de ces méthodes
8. Programmation orientée graphes

UC approach: multiple RKSM for eigenproblem



Synchronous or Asynchronous

UC approach: multiple RKSM

The choice of the initial guess

The choice of the subspace size m?

- m too small: not appearance of the desired eigenvalues
- m too large : high computation cost

Unite and conquer approach :

The projection of the problem on several subspaces, K_{mi}, and the computation of the initial guess by taking the intermediary eigen-information into account.

$$m_i \in [m_{min}, \ldots, m_{max}]$$

Definition of the subspaces

- Different subspaces

$K_{mi} = span\{v_i,\ Av_i,\ \ldots,\ A^{mi-1}v_i\}$

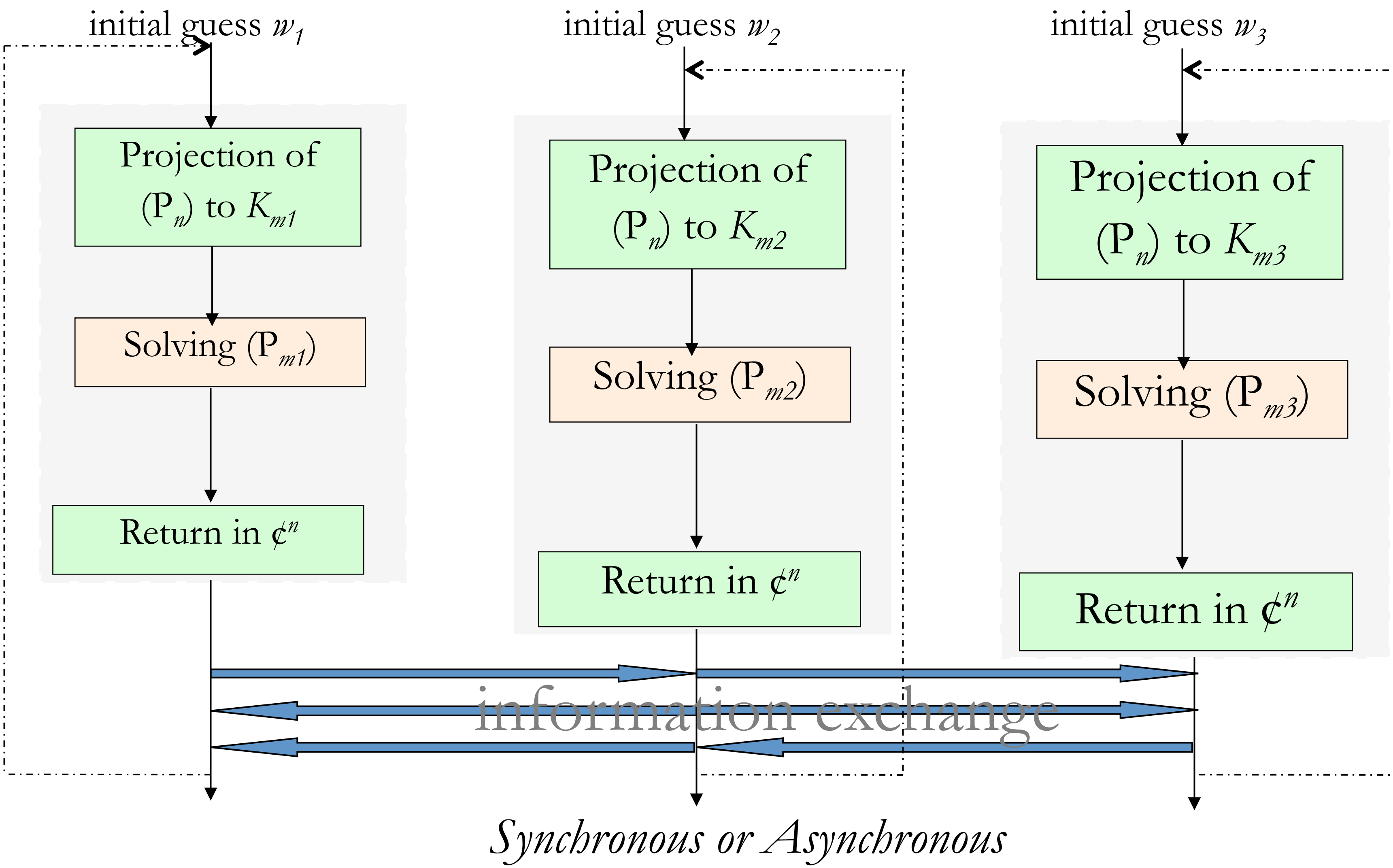
with $m_i \neq m_j$ and $v_i \neq v_j$ for $i,j \in [1,\ldots,l]$ and $i \neq j$

- Nested subspaces

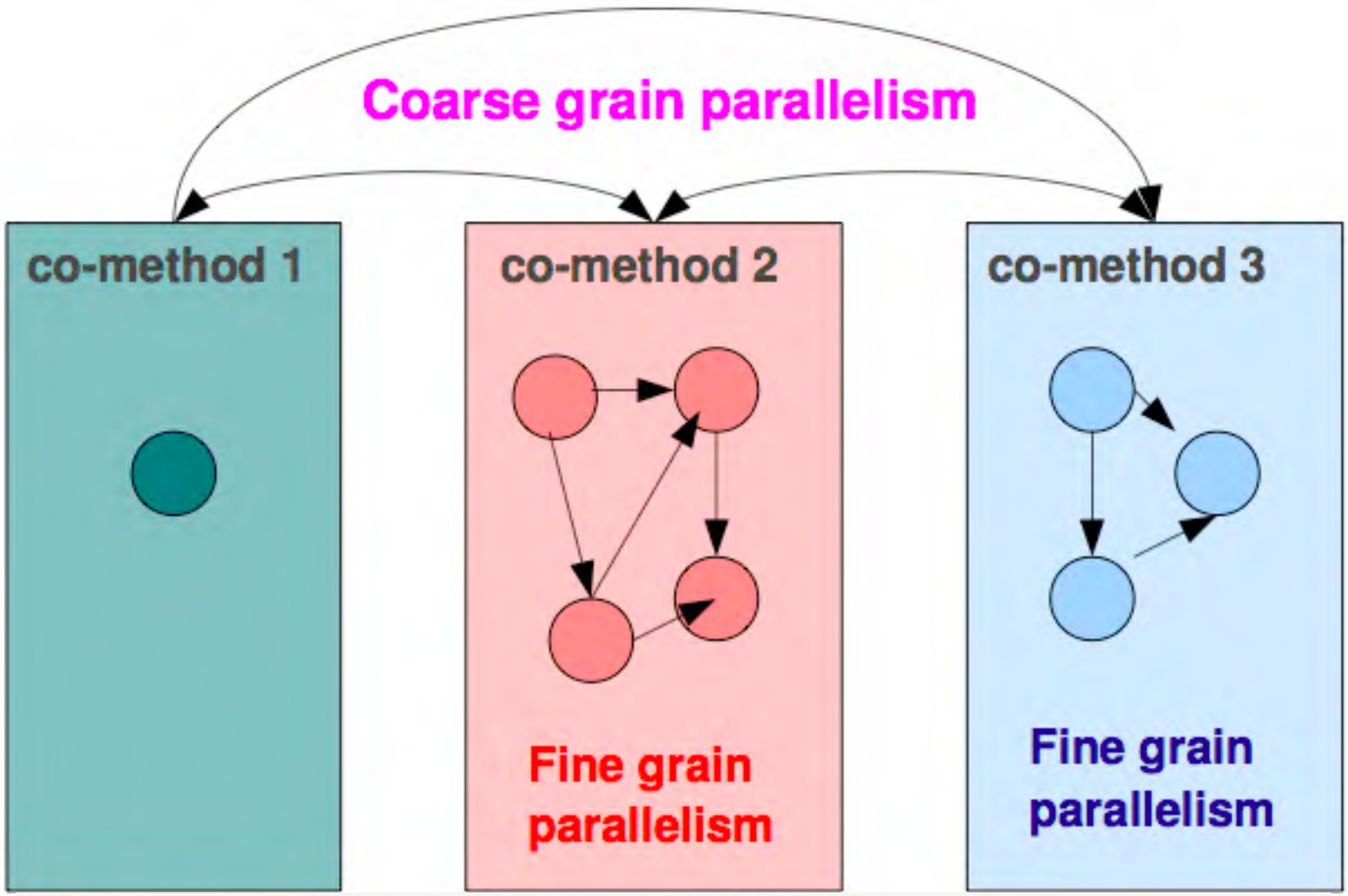
$K_{ml} = span\{v_1, Av_1,\ \ldots,\ A^{m1-1}v_1,\ A^{m1}v_1,\ \ldots,\ A^{m2-1}v_1,\ A^{m2}v_1,$
$\ldots, A^{m3-1}v_1,\ \ldots,\ A^{ml-1}v_1\}$

with $K_{m1} \subset K_{m2} \subset K_{m3} \subset \ldots \subset K_{ml}$

UC approach: multiple RKSM for eigenproblem

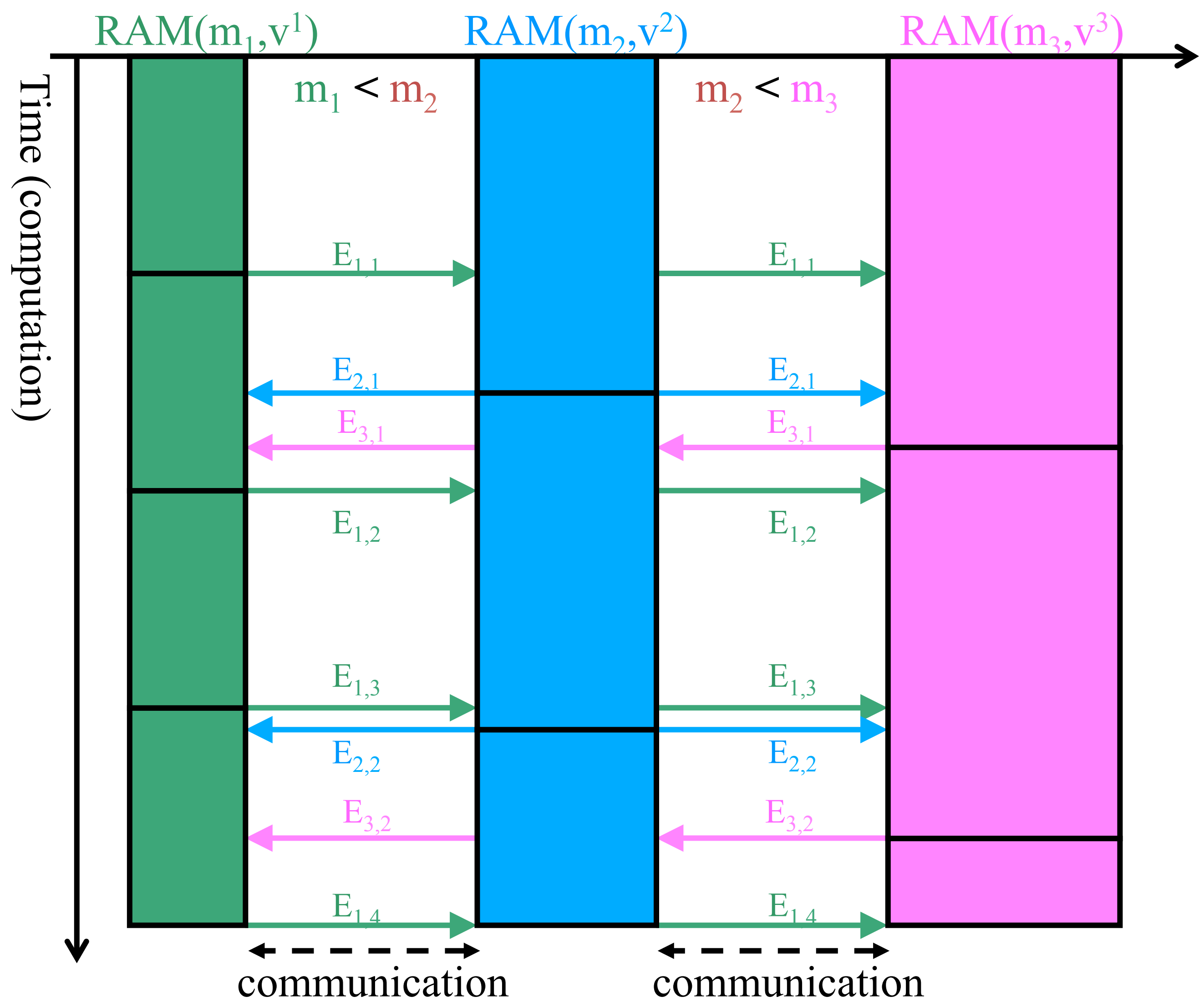


Multiple RKSM

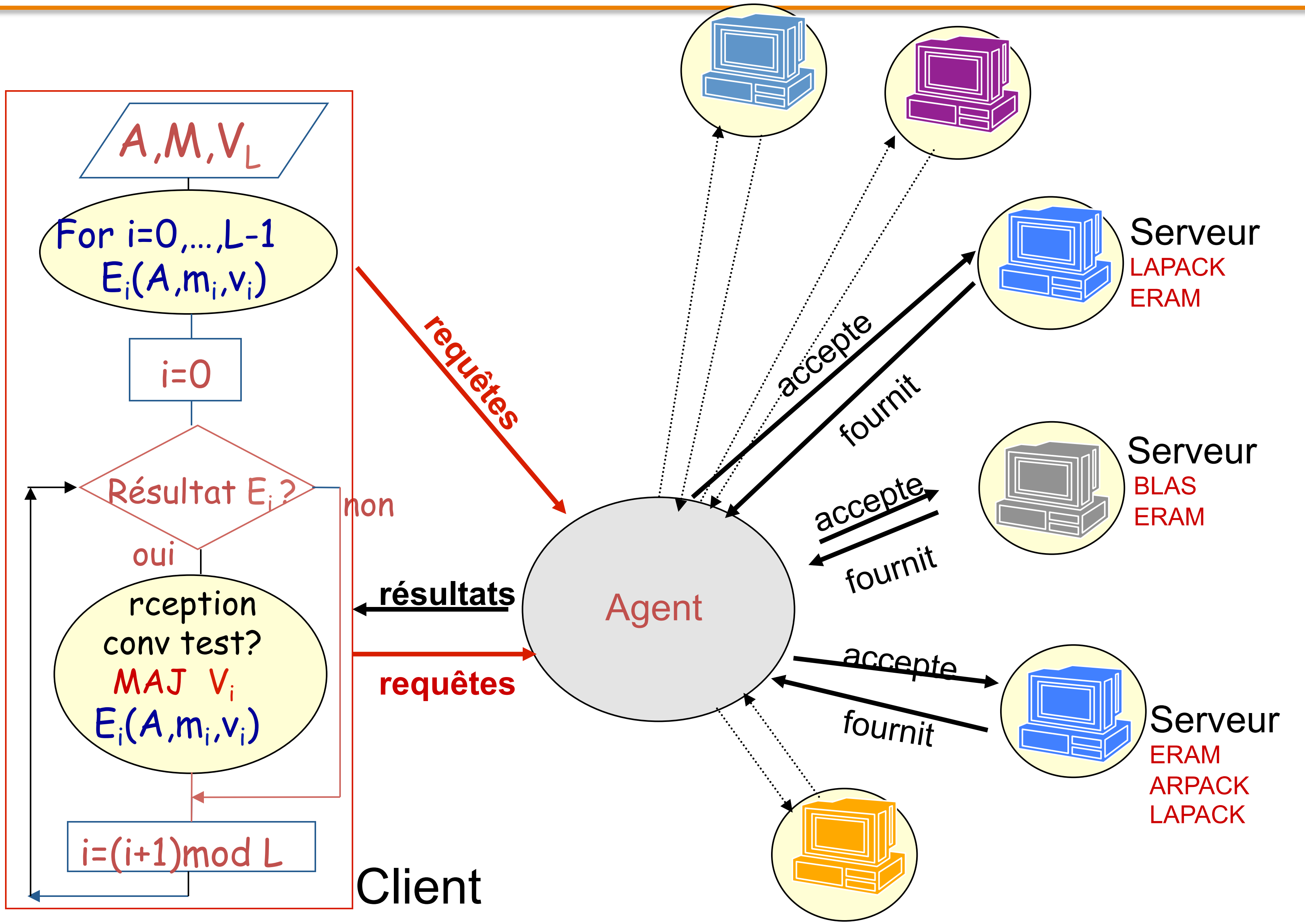


Asynchronous communication between coarse grain components

Asynchronous MRAM



MERAM with NetSolve on distributed system



Characteristics of UC methods

- Multi level parallelism (coarse grain and fine grain)
- Asynchronous communication
- Fault tolerance
- Great potential to dynamic load balancing
- Many parameters, many reuse software components
- Need well suited «standard» programming tools

well suited to petascale & future exascale computing system

A realistic example of MERAM

Let be $n=10^{11}$, $c=10^3$ *and* $m=10^{3.}$

- Each restarting cycle of an ERAM requires *100 petaflop* and *10 Peta bytes memory space*.
- Let *iter* be the number of iterations required until convergence and *b* the number of ERAM's instances defining MERAM. With realistic hypotheses *iter=100* and *b =10*, MERAM would requires (*iter×b and b×10)*:

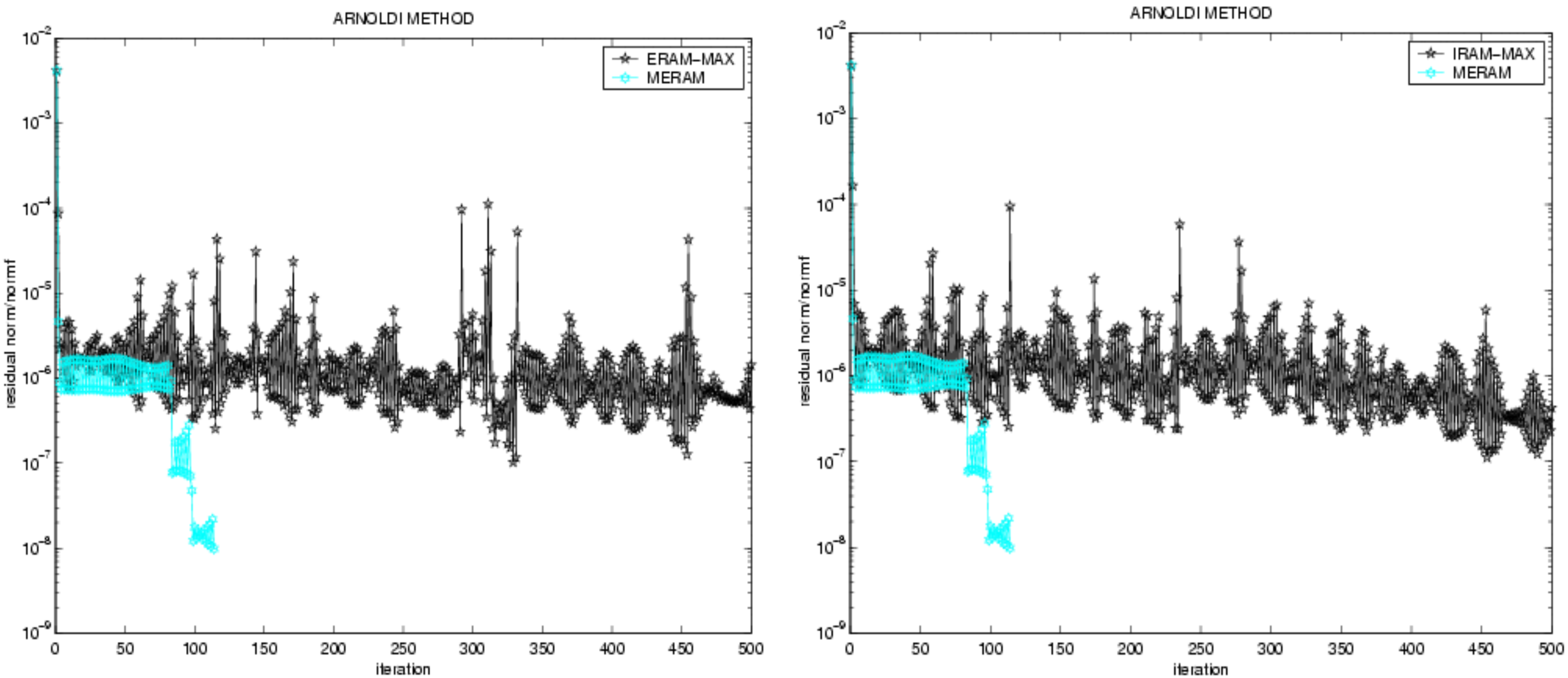
100 Exa flop

100 peta bytes memory space.

PLAN DE COURS

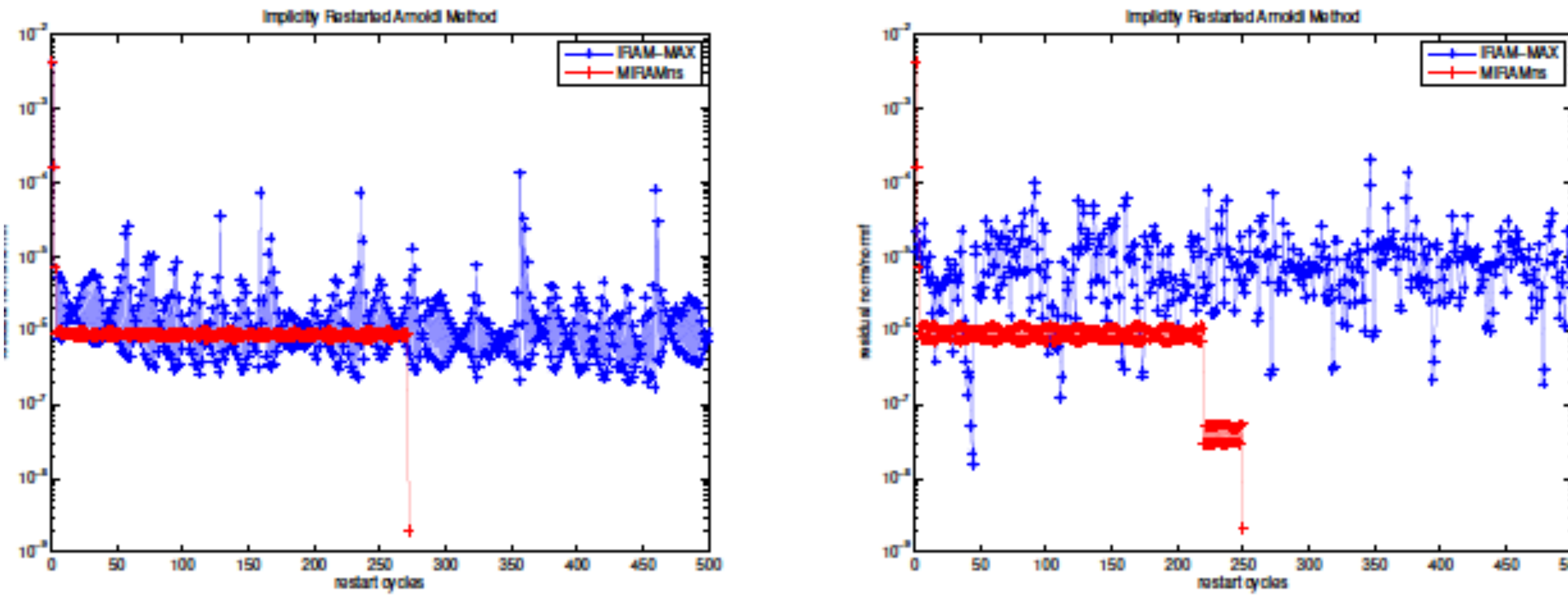
1. Introduction
2. Quel paradigmes de programmation?
3. Approche « *unir & conquérir* »
4. Méthodes: Multiple Krylov Subspace (MKS)
5. Quelques résultats expérimentaux
6. Zoom sur certaines méthodes itératives pour des problèmes d'AL de grande taille (TD/TP)
7. Zoom sur l'hybridation de ces méthodes
8. Programmation orientée graphes

MERAM vs ERAM and IRAM



af23560, k=2, tol=10⁻⁸, MERAM(5,7,10), V=(z, z, z), ERAM(10), IRAM(10) with z=(1/√n,…,1/√n)

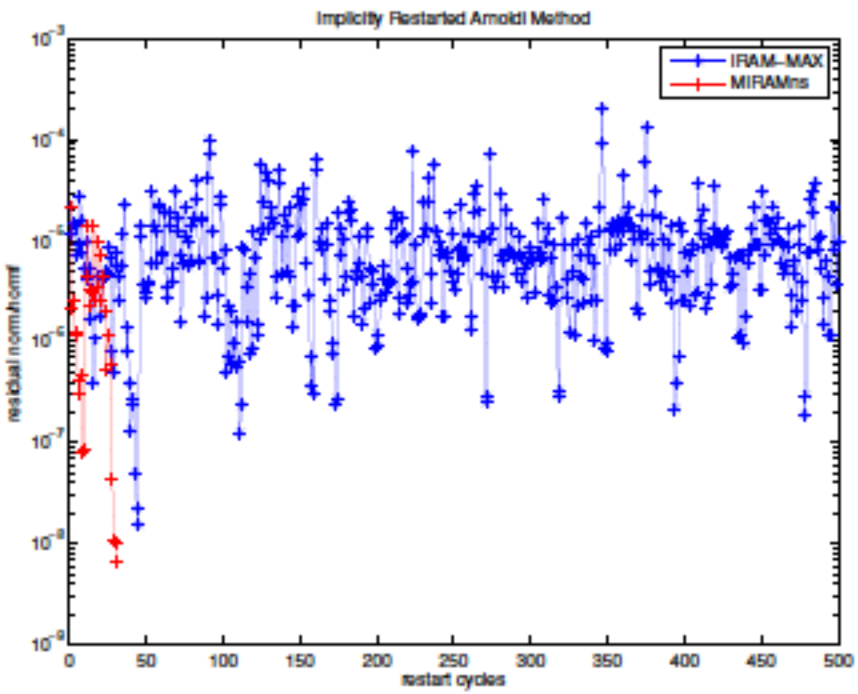
MIRAM vs IRAM



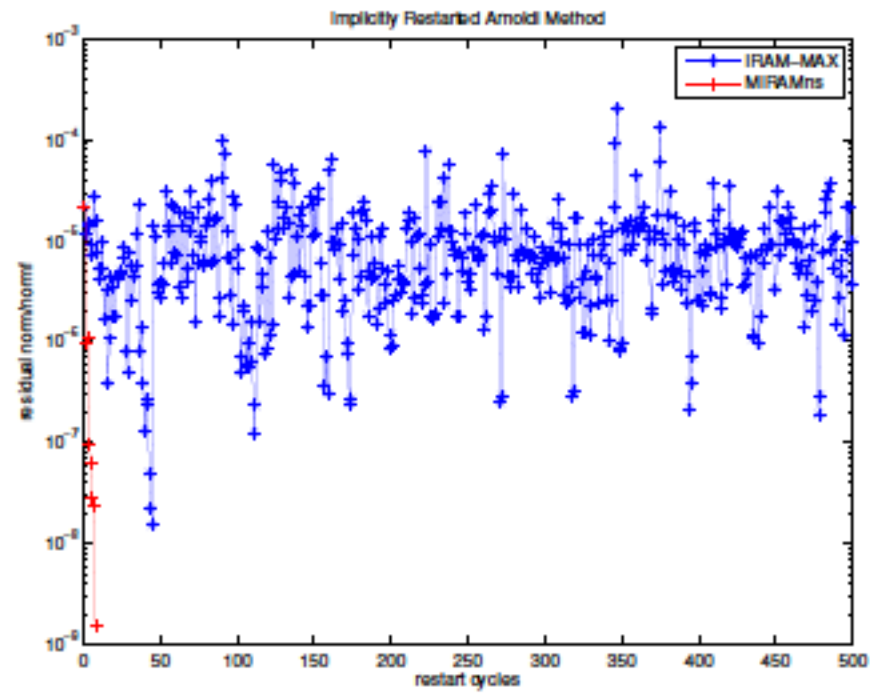
(a) $m = 10$, $x = z_n$ (b) $m = 22$, $x = s_n$

af23560, k=2, tol=10^{-8}, MERAM(5,7,10), IRAM(10), IRAM(5+7+10) with $z_n=(1/\sqrt{n},\ldots,1/\sqrt{n})^T$, $s_n=(1,1,0.1,\ \ldots 0.1)^T$

MIRAM vs IRAM(max)



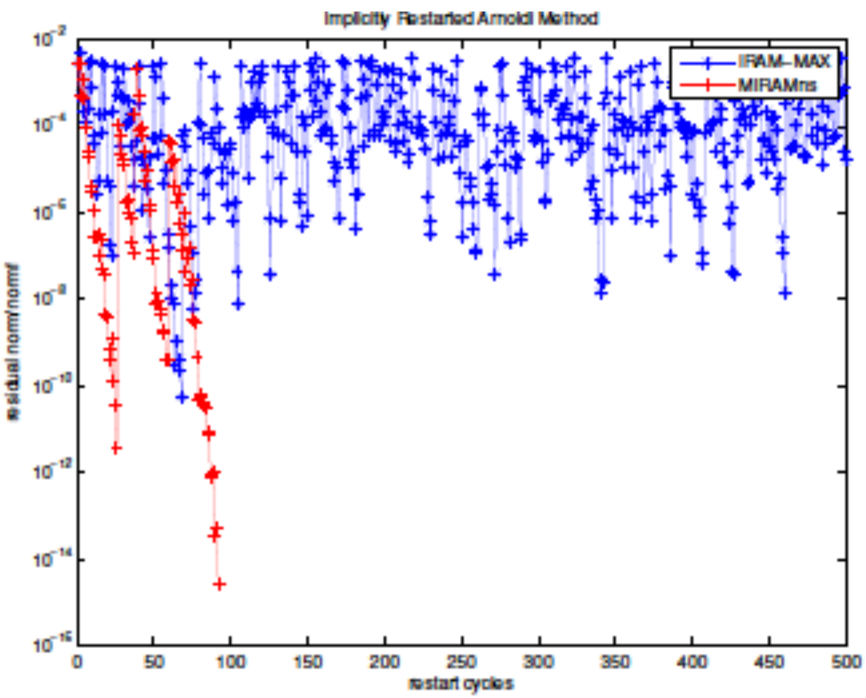
(a) MIRAMns(16, 19, 22)



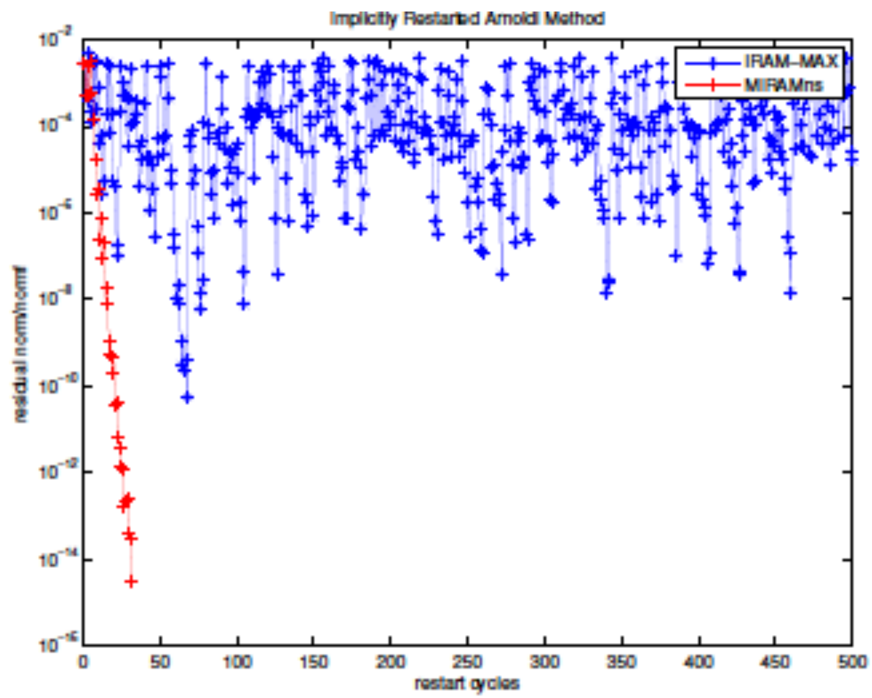
(b) MIRAMns(10, 13, 16, 19, 22)

af23560, k=2, tol=10^{-8}, IRAM(22), s=$(1,1,0.1, \ldots 0.1)^T$

MERAM vs IRAM (max)



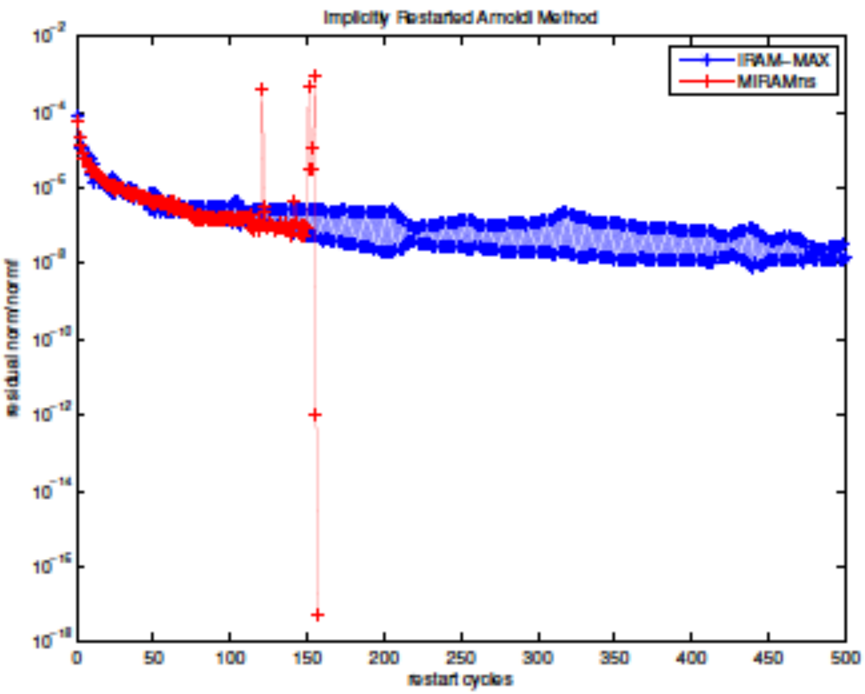
(a) MIRAMns(8, 18, 20)



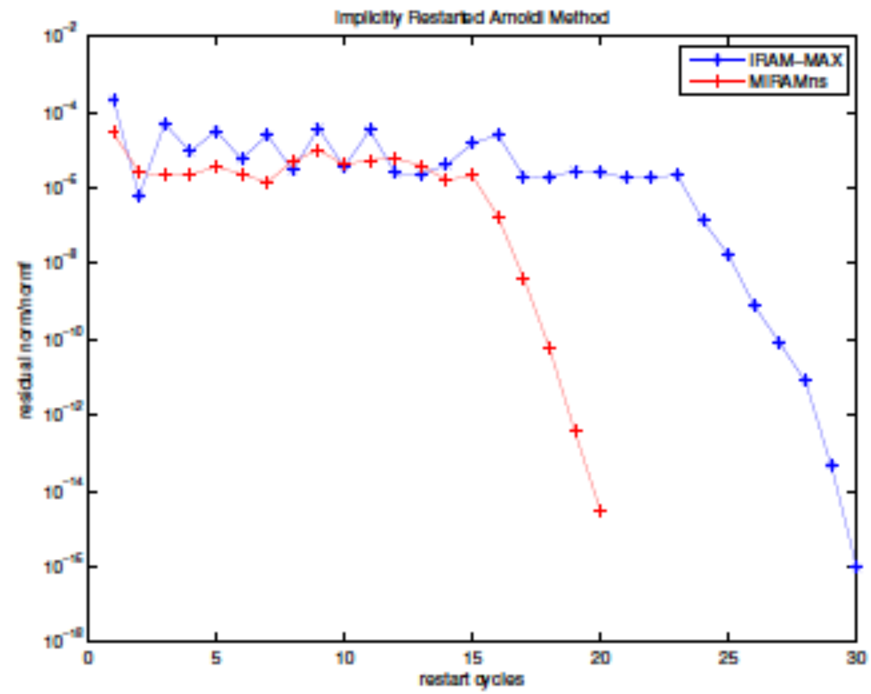
(b) MIRAMns(8, 11, 14, 17, 20)

bfw782a, k=2, tol=10^{-14}, IRAM(20), with z=($1/\sqrt{n}$,…,$1/\sqrt{n}$)

$\mathrm{MIRAMns}(5, 8, 11, 14, 17, 20)$ versus $\mathrm{IRAM}(20)$



(a) Matrix *roadNet-PA*, $k = 2$



(b) Matrix *com-Youtube*, $k = 4$

$tol=10^{-14}, t_n=(1,1,0,\ldots,0)^T$

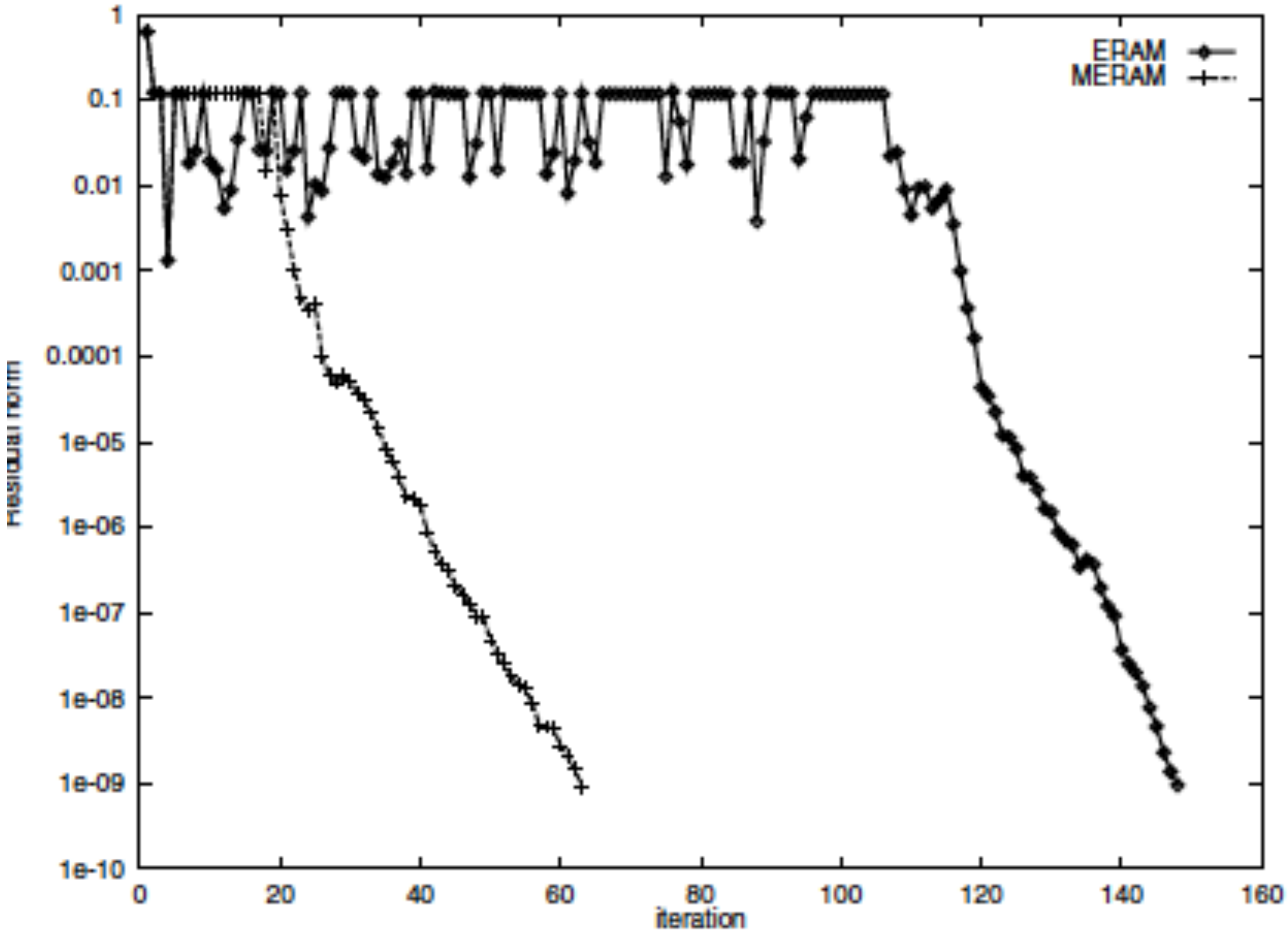
Hardware platforms

- **Grid'5000** : Cluster of clusters distributed over 10 distant sites and > 5000 cores.
- **Carver** : IBM iDataPlex System at NERSC/LBNL (9984 cores, 1120 nodes of 8 cores & 80 noeds of 12 cores).
- **K Computer**: 864 rack x 102 nodes x 1 CPU = 88,128 CPUs. Node: SPARC64 VIIIfx (8core) CPU 128GFLOPS/node
- A distributed systems with PC workstations in several distant sites (USA, France)

Test matrices

Matrix	Size of matrix	nonzero elements
af23560.mtx	23560	484256
bfw782a.mtx	782	7514
A9_1000.mtx	1000	2998
west0989.mtx	989	3537
AM_1000.mtx	1000	2998
sherman3.mtx	5005	20033
roadNet-PA.mtx	1088092	3083796
com-Youtube.mtx	1134890	2988374
WikiTalk.mtx	2394385	5046614

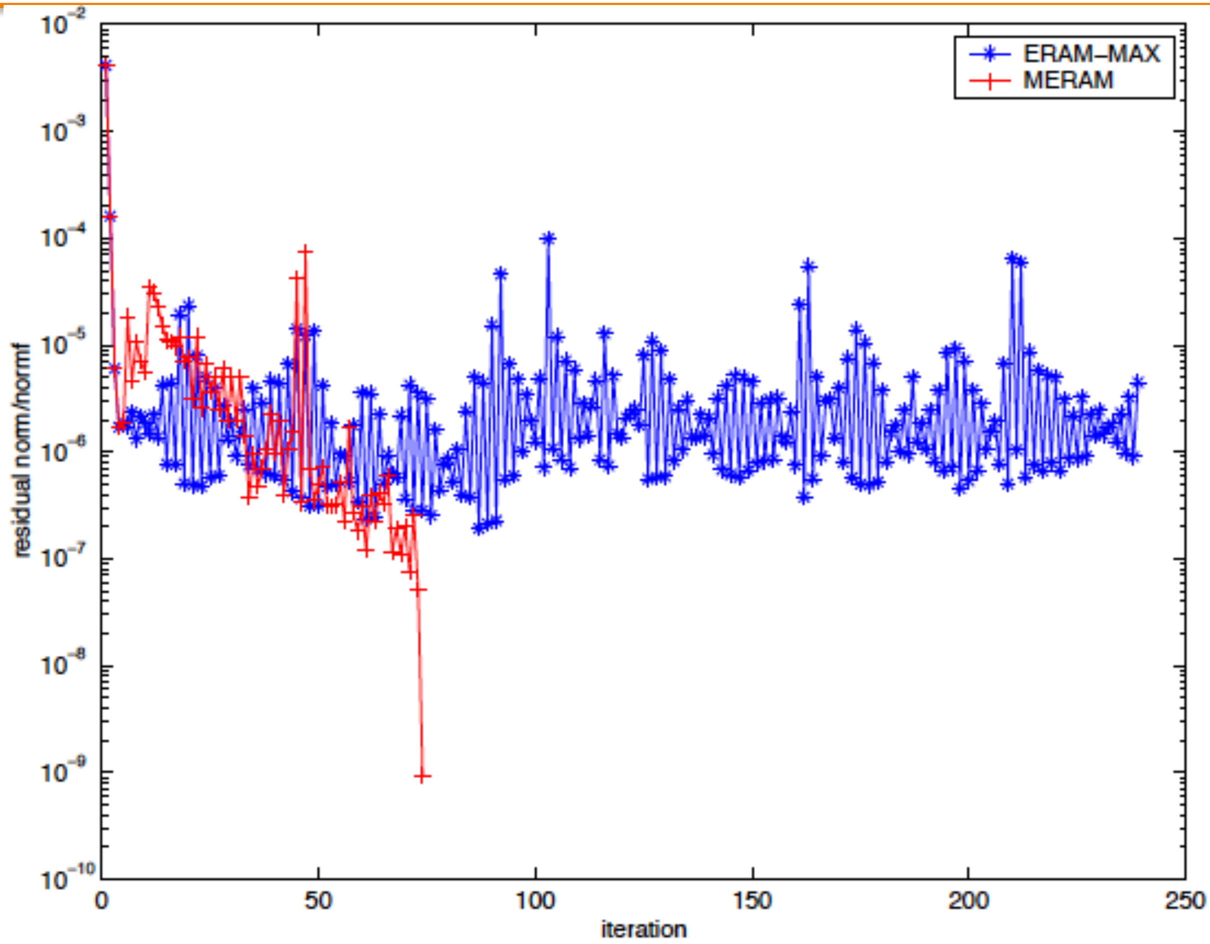
MERAM on CM5+CM200 and 2 PC platform



MERAM((40,40), [z,z]) versus ERAM(40,z) with 63-diagonal matrix, z=(1/√n , …, 1/√n)^T, s=4 algebraically largest eigenvalues are desired

N. Emad, S. Petiton, and G. Edjlali, Multiple Explicitly Restarted Arnoldi Method for Solving Large Eigenproblems, SJSC, 27(1):253–277, 2005

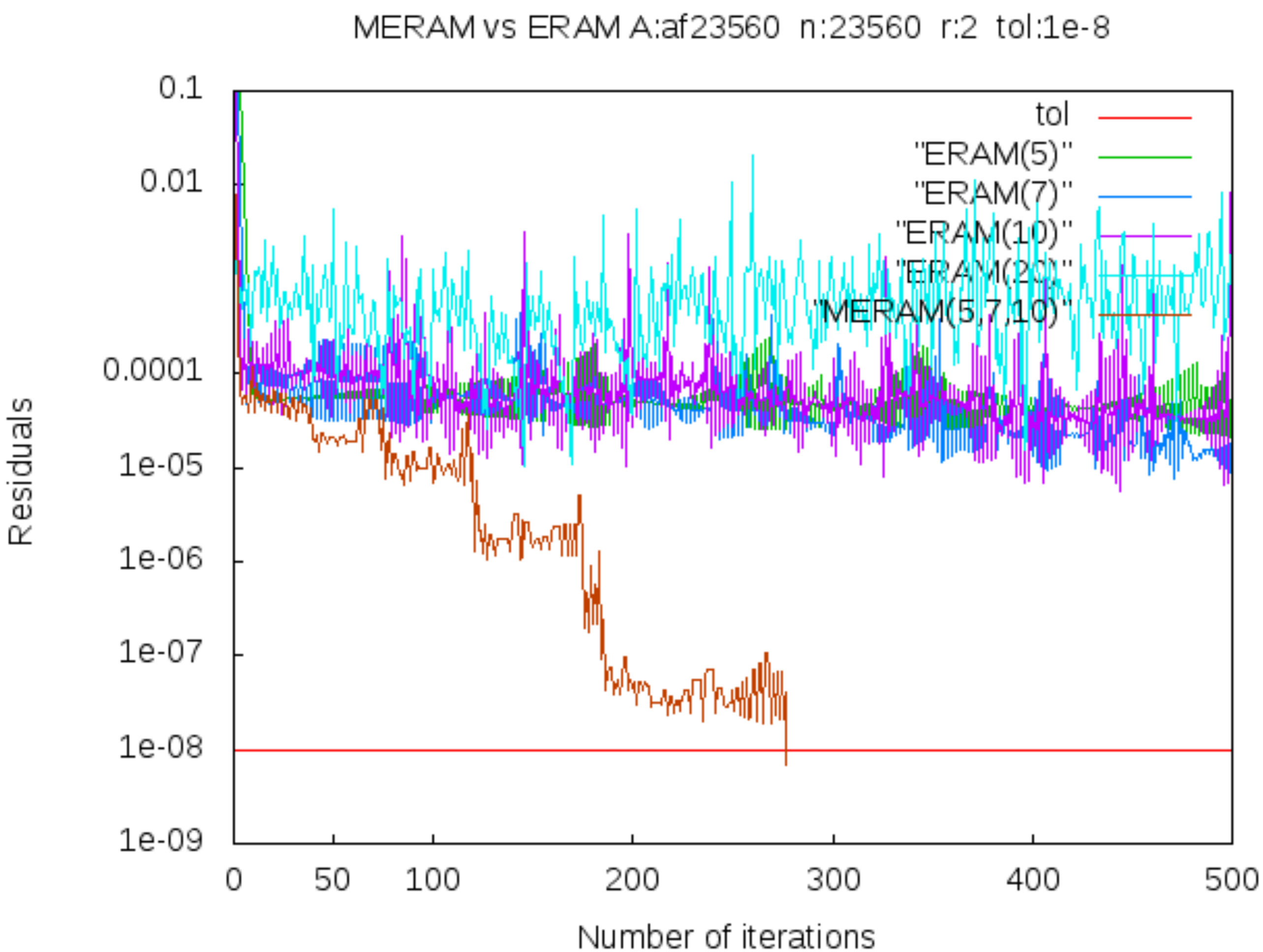
MERAM on NetSolve (France, USA)



MERAM(5,7,10) versus ERAM(10) with af236560 matrix. MERAM converges in 74 cycles, ERAM doesn't converge after 240 restart cycles

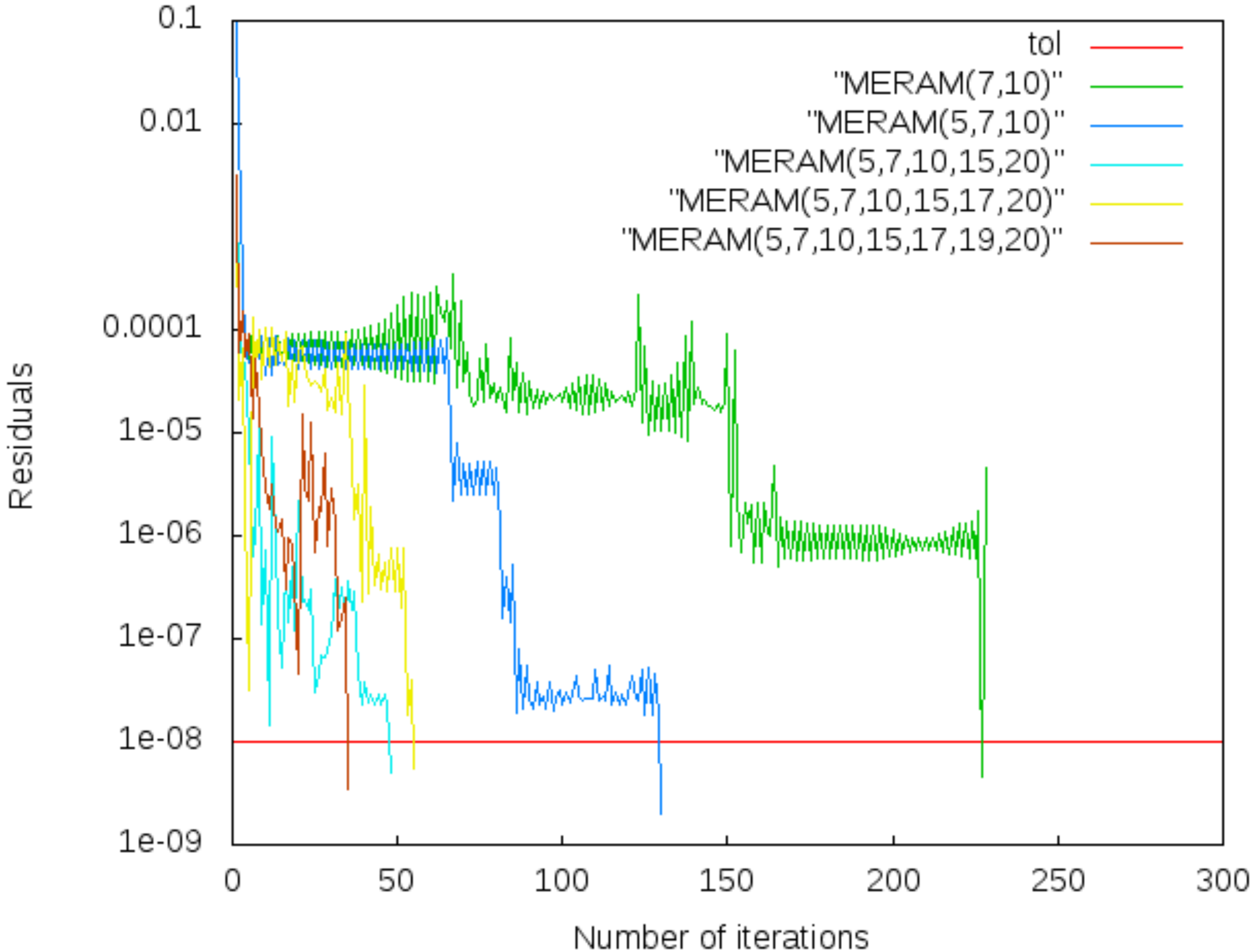
N. Emad, S.-A. Shahzadeh-Fazeli, and J. Dongarra, An Asynchronous Algorithm on NetSolve Global Computing System, FGCS, 22(3):279–290, 2006.

MERAM on Grid'5000 with120 cores



PETSc/SLEPc/YML

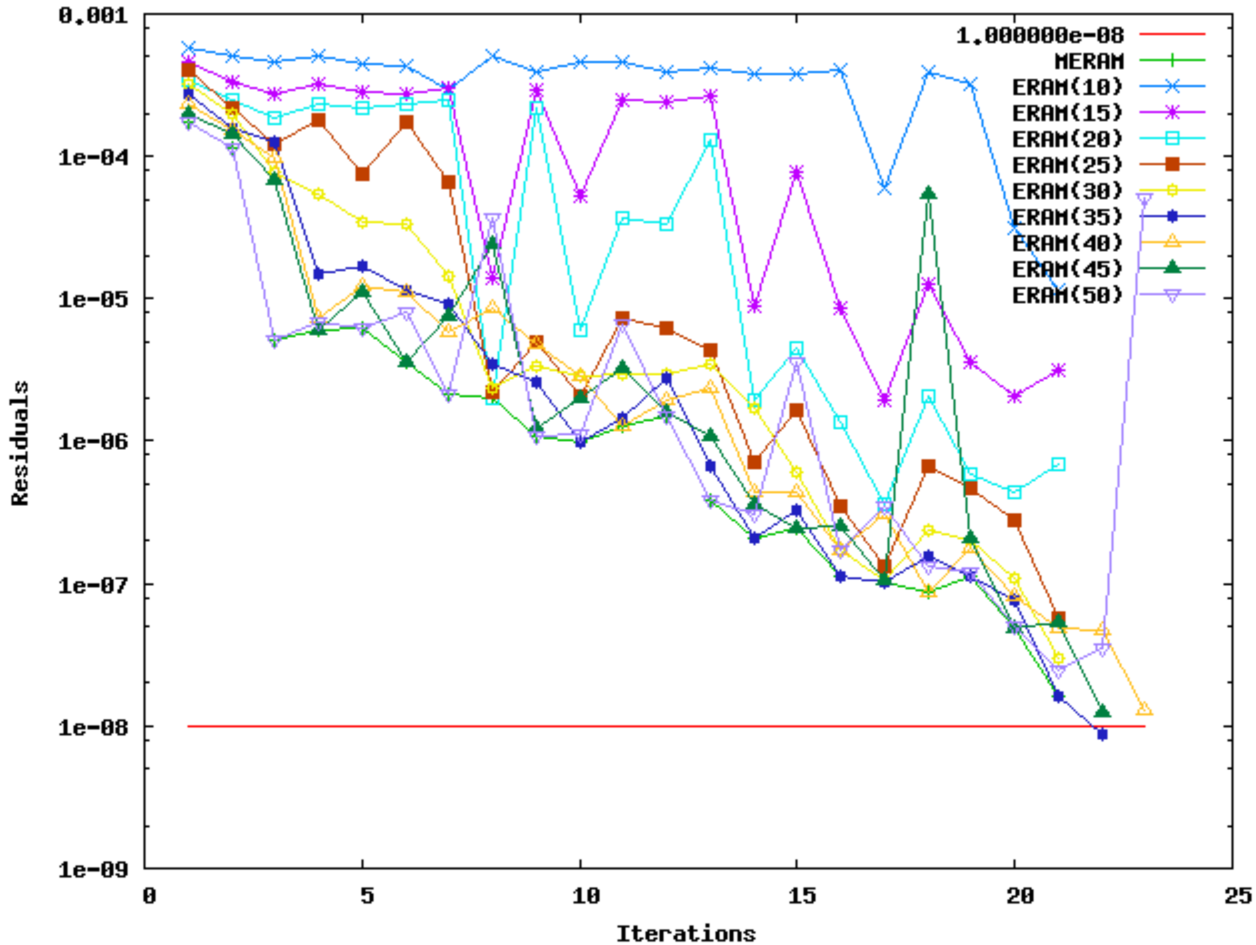
MERAM on Grid'5000 with120 cores



PETSc/ SLEPc/YML

scalability wrt the number of co-methods

MERAM on Grid'5000 platform



Scalability of the solution: number of co-methods/size of A. Convergence of MERAM for matrix pde1000000 with different number of co-methods (4 Ritz values)

Tuning Asynchronous Co-Methods for Large-scale Eigenvalue Calculations

Nahid Emad, University of Versailles, France, Nahid.Emad@prism.uvsq.fr
Leroy Drummond, Lawrence Berkeley National Laboratory, USA, ladrummond@lbl.gov
Miwako Tsuji, RIKEN/AICS, Japan, miwako.tsuji@riken.jp
Makarem Dandouna, University of Versailles, France, makarem.dandouna@prism.uvsq.fr

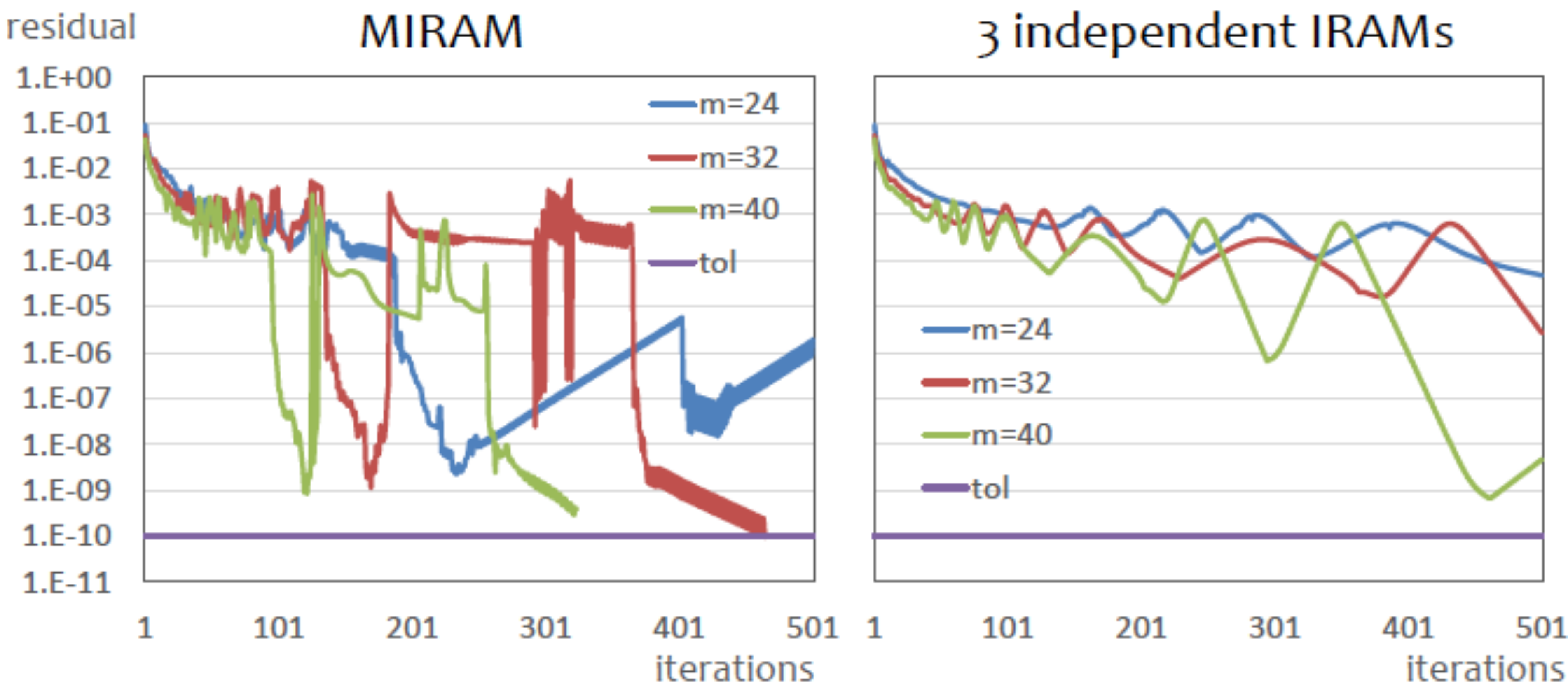
SIAM PP 2014

K computer
Computer simulations create the future

MIRAM with FP2C (XMP/YML) + PETSc/SLEPc/ARPACK: Experiments

- K-Computer
  - Node
    - SPARC64 VIIIfx (8core) CPU 128GFLOPS/node
    - 16 GB DDR3 SDRAM
  - System
    - 864 rack x 102 nodes x 1 CPU = 88,128 CPUs
    - Tofu: six-dimensional torus interconnect
    - FEFS (Fujitsu Exabyte File System) based on Lustre

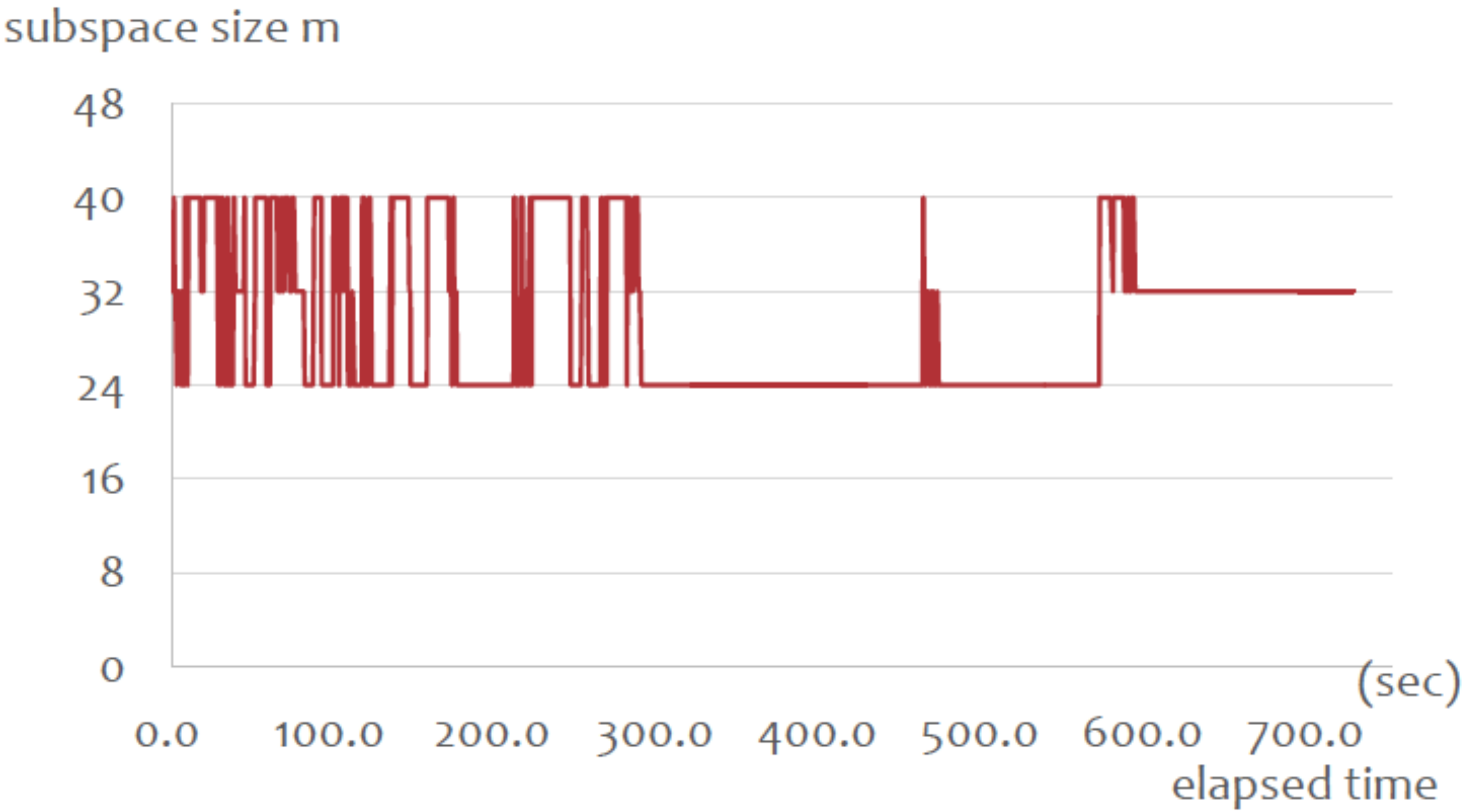
MIRAM: Speedup Convergence



We can reduce the number of iterations!

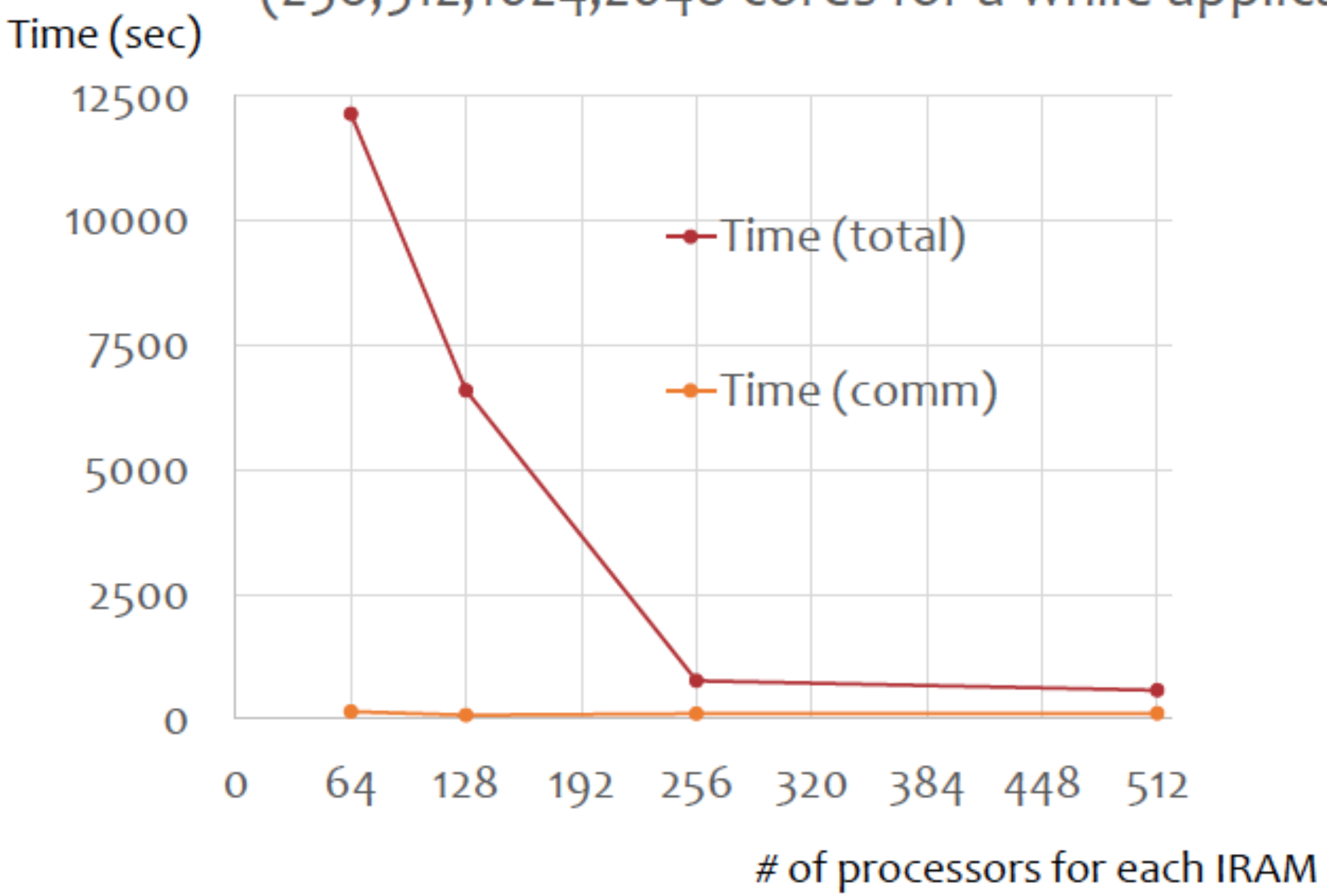
Schenk/nlpkkt240 (the largest matrix in the UF Sparse Matrix Collection):
$n=27993600$, $k=10$, $tol=10^{-10}$,

Tuning Subspace Size in MIRAM



MIRAM: Speedup Execution Time

64, 128, 256, 512 cores for each IRAM
(256,512,1024,2048 cores for a while application)



PLAN DE COURS

1. Introduction
2. Quel paradigmes de programmation?
3. Approche « *unir & conquérir* »
4. Méthodes: Multiple Krylov Subspace (MKS)
5. Quelques résultats expérimentaux
6. Zoom sur certaines méthodes itératives pour des problèmes d'AL de grande taille (TD/TP)
7. Zoom sur l'hybridation de ces méthodes
8. Programmation orientée graphes

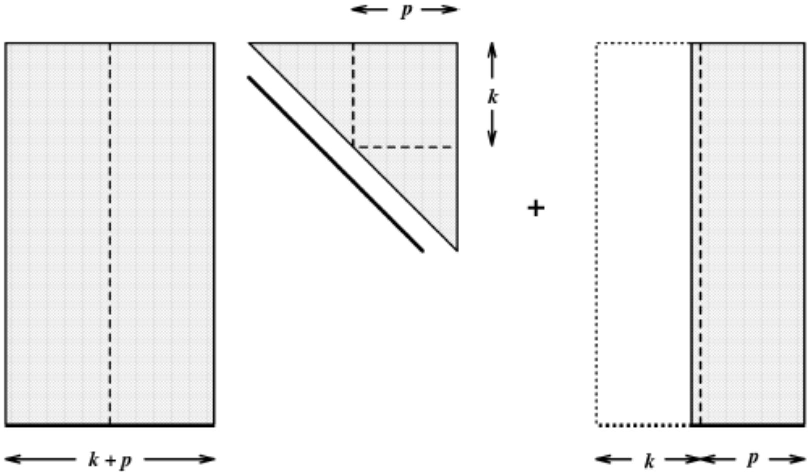
Méthodes numériques pour pb à grande taille

- ERAM
- IRAM
- LANCZOS
- SVD
- Etc.

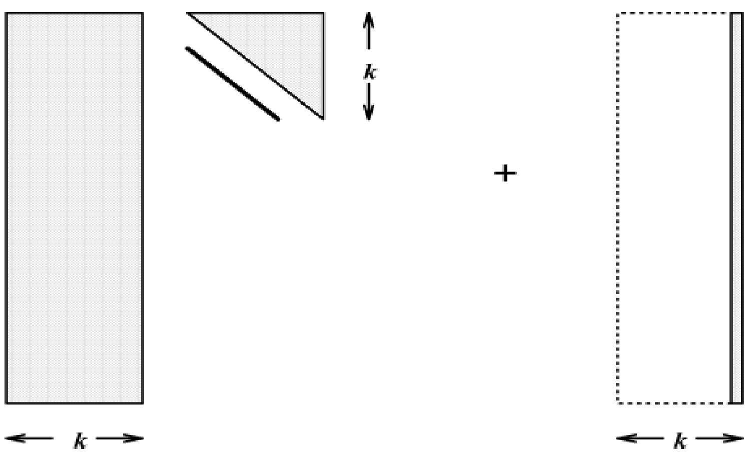
PLAN DE COURS

1. Introduction
2. Quel paradigmes de programmation?
3. Approche « *unir & conquérir* »
4. Méthodes: Multiple Krylov Subspace (MKS)
5. Quelques résultats expérimentaux
6. Zoom sur certaines méthodes itératives pour des problèmes d'AL de grande taille (TD/TP)
7. Zoom sur l'hybridation de ces méthodes
8. Programmation orientée graphes

IRAM (m=k+p)



$V_{k+p}QQ^TH_{k+p}Q+f_{k+p}e^T_{k+p}Q$
after p implicitly shifted QR



$V_kH_k+f_ke^T_k$
from a k-step AF after discarding the last p columns

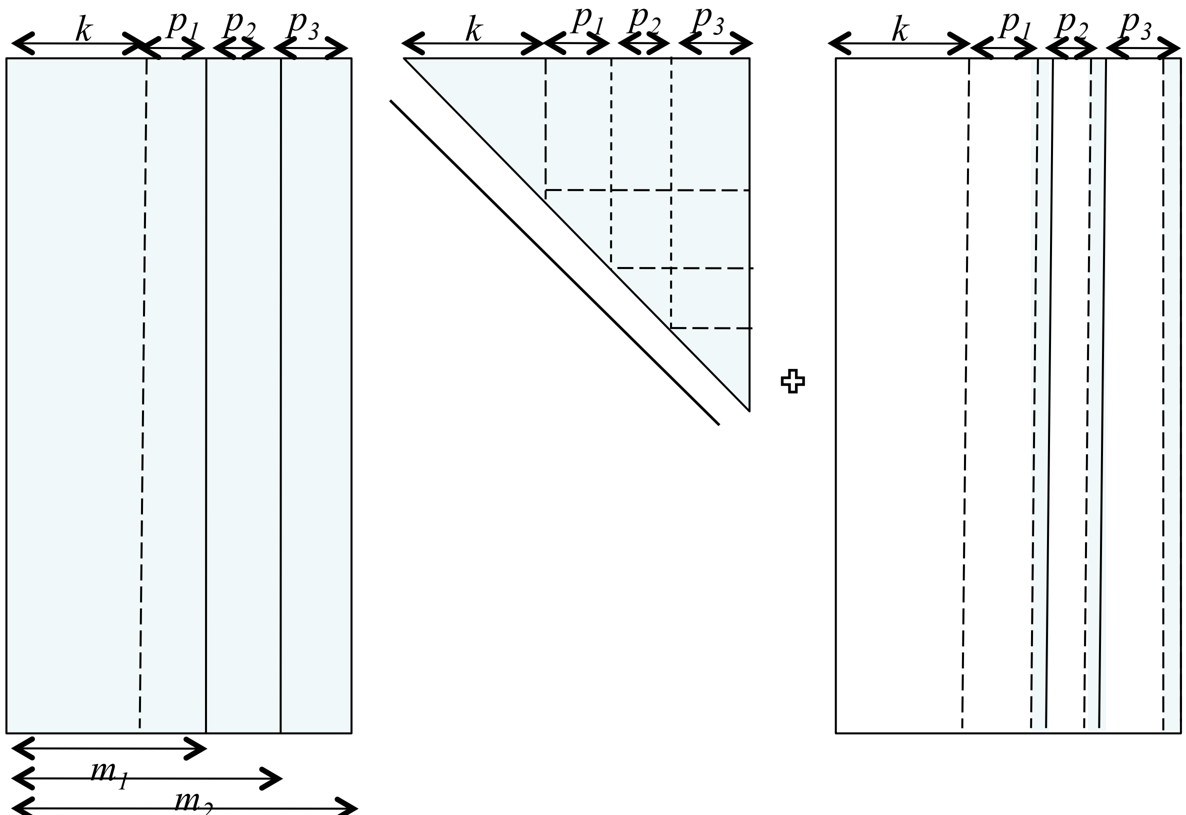
Beginning with $AV^+_k=V^+_kH^+_k+f_k^+e^T_m$, l m_i-step AF

$AV_{mi}=V_{mi}H_{mi}+f_{mi}e^T_{mi}$ for $i=1, \ldots, l$

MIRAM with nested subspaces

$K_{ml} = span(v_1, Av_1, \ldots, A^{m1-1}v_1, A^{m1}v_1, \ldots, A^{m2-1}v_1, A^{m2}v_1, \ldots, A^{m3-1}v_1, \ldots, A^{ml-1}v_1)$

$K_{m1} \subset K_{m2} \subset \ldots \subset K_{ml}$



$V_{mi} \; H_{mi} \; + \; f_{mi} e^T_{mi}$

MIRAM with nested subspaces

$$AV_{m1}=V_{m1}H_{m1}+f_{m1}e^{T}_{m1} \Longrightarrow (\Lambda_{k,}U_{k})_{m1}$$

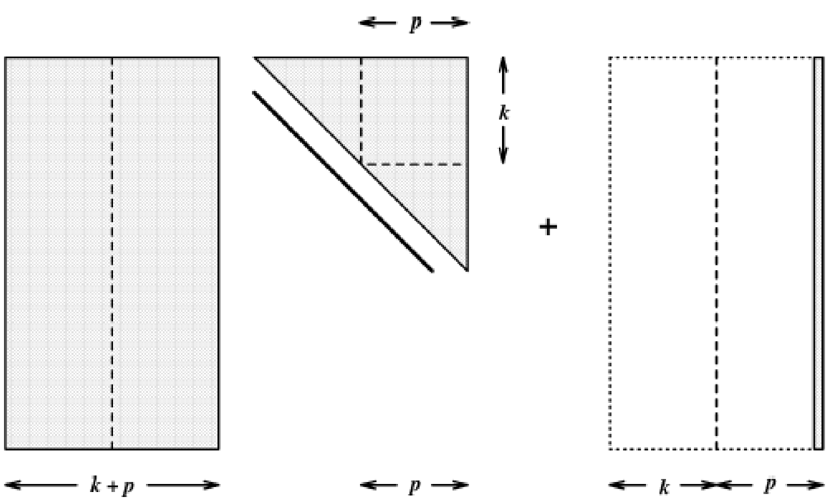
$$AV_{m2}=V_{m2}H_{m2}+f_{m2}e^{T}_{m2} \Longrightarrow (\Lambda_{k,}U_{k})_{m2}$$

$$\cdots\cdot \qquad\qquad \cdots\cdot$$

$$AV_{ml}=V_{ml}H_{ml}+f_{ml}e^{T}_{ml} \Longrightarrow (\Lambda_{k,}U_{k})_{ml}$$

$$(\Lambda_{k},U_{k})_{m\_best}$$

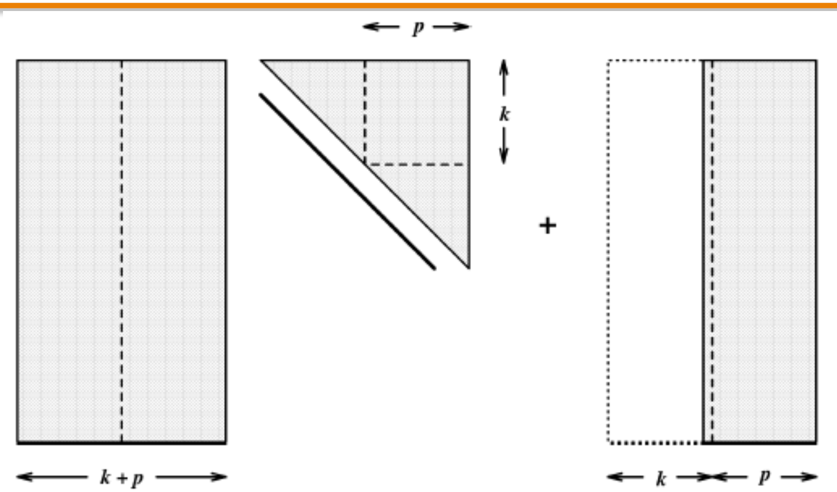
$m_i=k+p_i$ for $i=1, \ldots, l$



$Let\ m=m_{best}\ \ p=p_{best}=m_{best}\text{-}k$

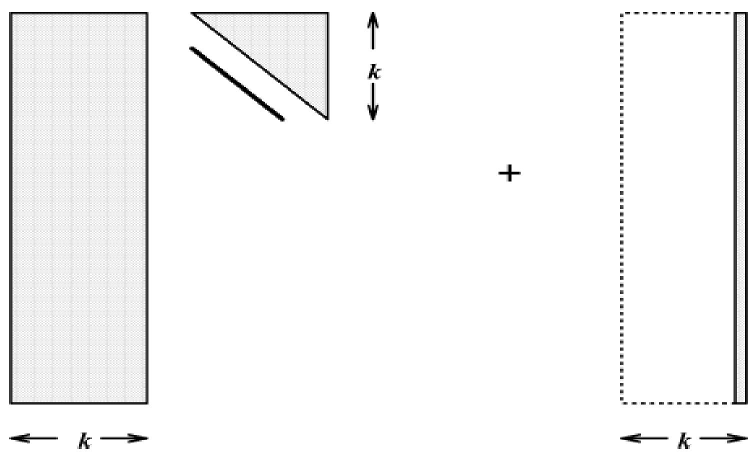
$$AV_{k+p}=V_{k+p}H_{k+p}+f_{k+p}e^{T}_{k+p}$$

MIRAM with nested subspaces



$V_{k+p}QQ^TH_{k+p}Q+f_{k+p}e^T{}_{k+p}Q$

after p implicitly shifted QR



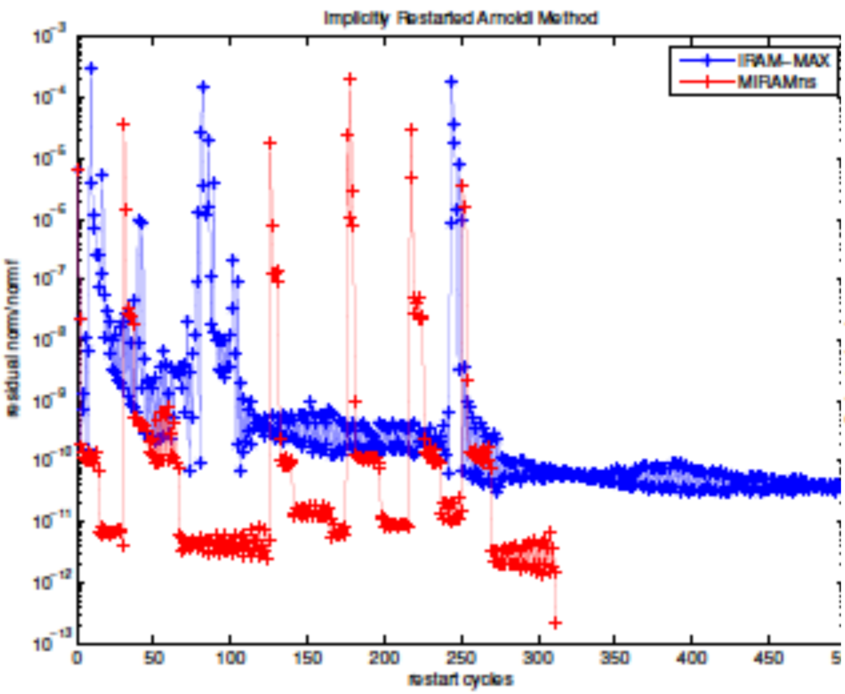
$V_kH_k+f_ke^T{}_k$

from a k-step AF after discarding the last p columns

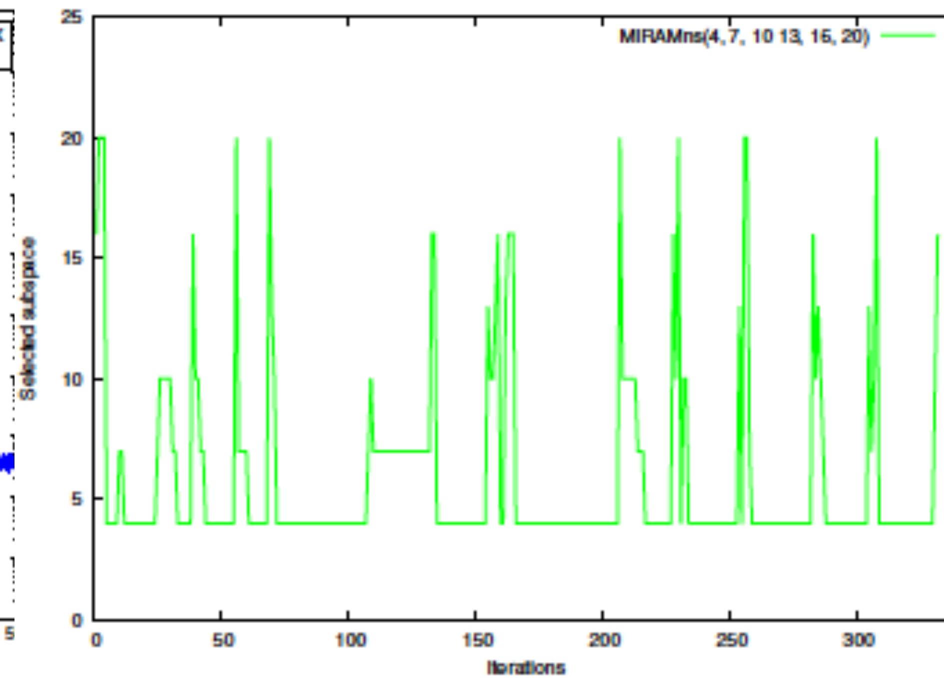
Beginning with $AV^+{}_k=V^+{}_kH^+{}_k+f_k{}^+e^T{}_m$, *l* m_i*-step AF*

$AV_{mi}=V_{mi}H_{mi}+f_{mi}e^T{}_{mi}$ *for* $i=1, \ldots, l$

Evolution of m_{best} in MIRAM



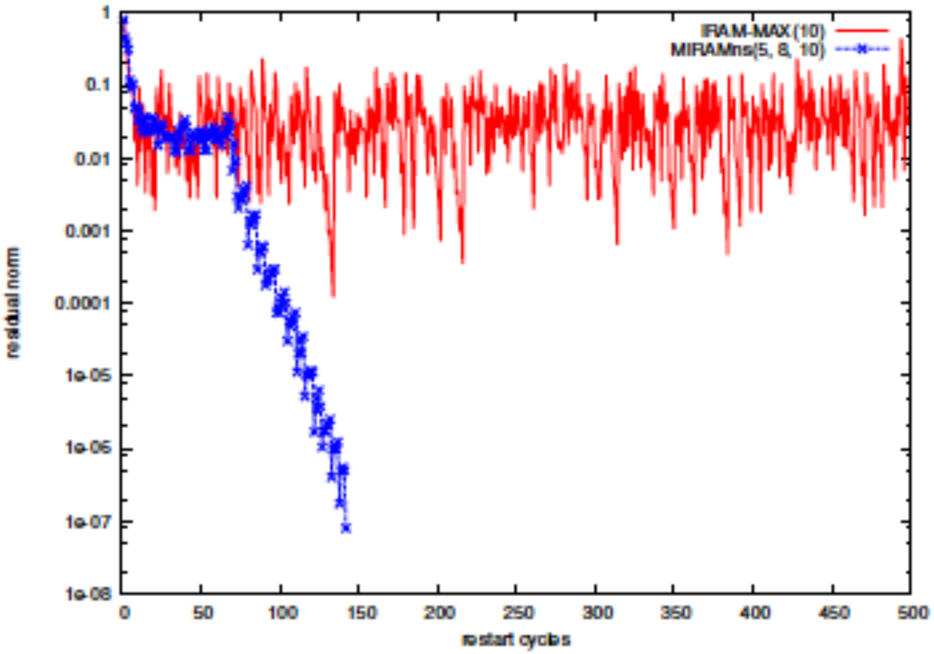
(a) MIRAMns(4, 7, 10, 13, 16, 20), $k = 2$, $tol = 10^{-12}$



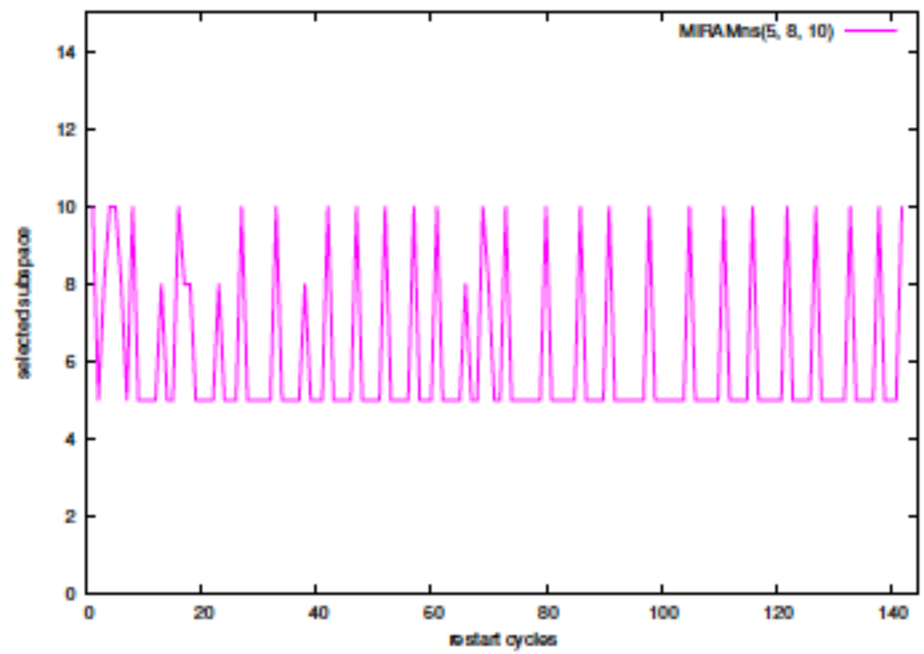
(b) MIRAMns(4, 7, 10, 13, 16, 20) with with $WikiTalk$ matrix

S. A. Sahzadeh Fazeli, N. Emad, Z. Liu. A key to chose subspace size in implicitly restarted Arnoldi method, To be appear in Journal of Numerical Algorithm

Evolution of m_{best} in MIRAM within ARPACK



(a) MIRAMns(5, 8, 10) versus IRAM(10)

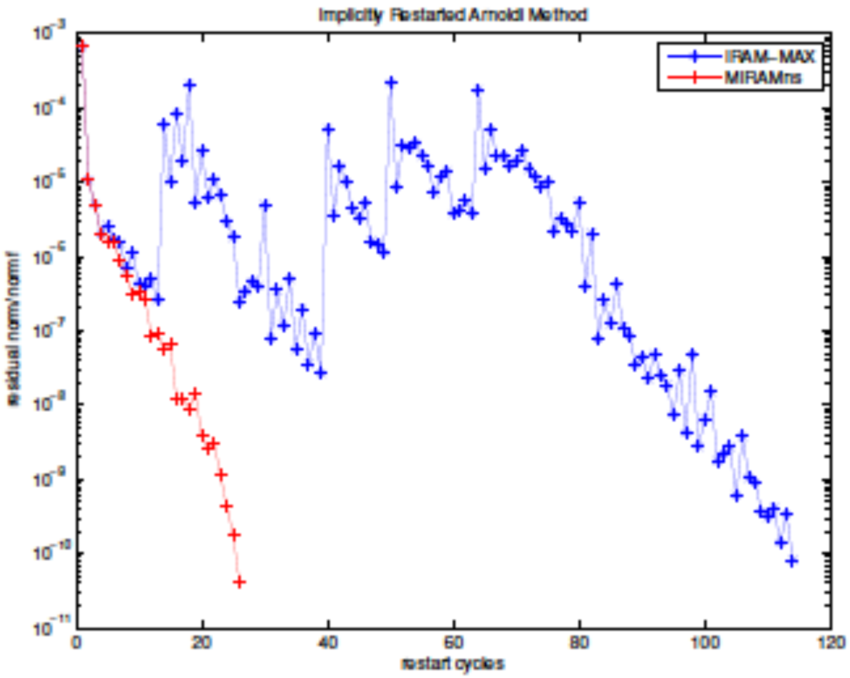


(b) Evolution of m_{best} along cycles

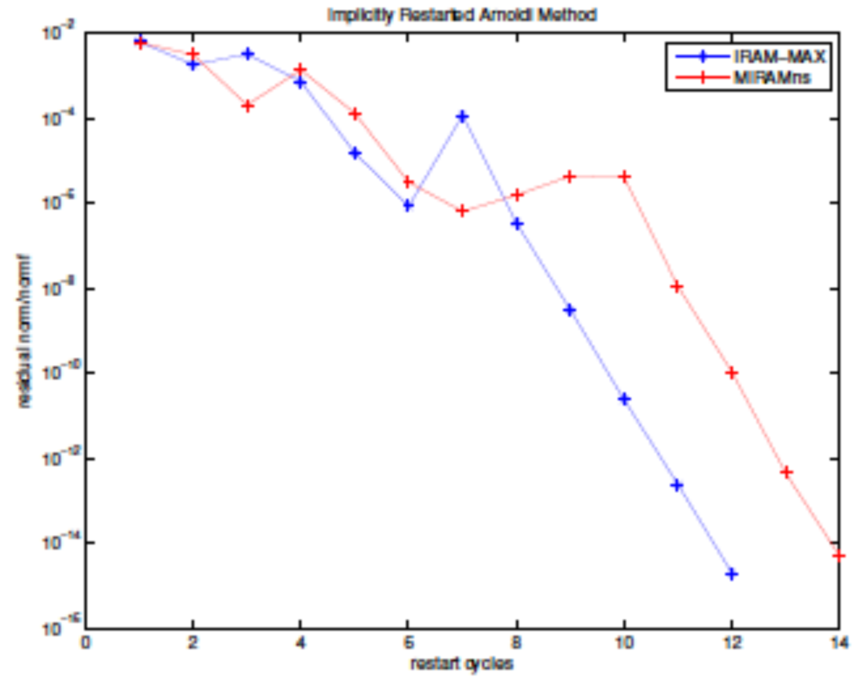
Bfw782a, k=2, tol=10^{-8} and a random initial guess

Tick restarted: MIRAMns vs IRAM

A.Stathopoulos, Y. Saad, K. Wu



(a) MIRAMns$(20, 30, 40)$, $m = 40$, $q = 2$

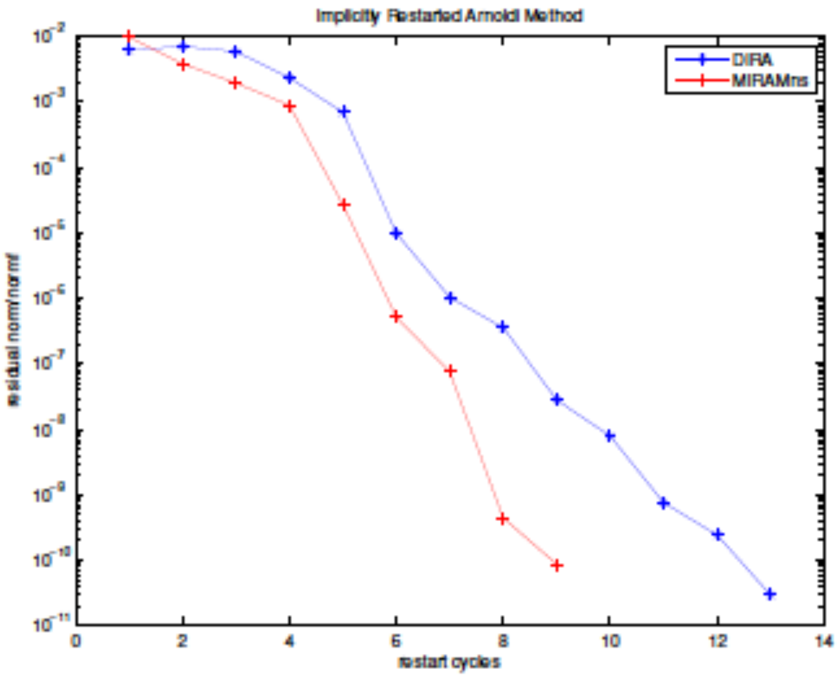


(b) MIRAMns$(20, 25, 30)$, $m = 30$, $q = 5$

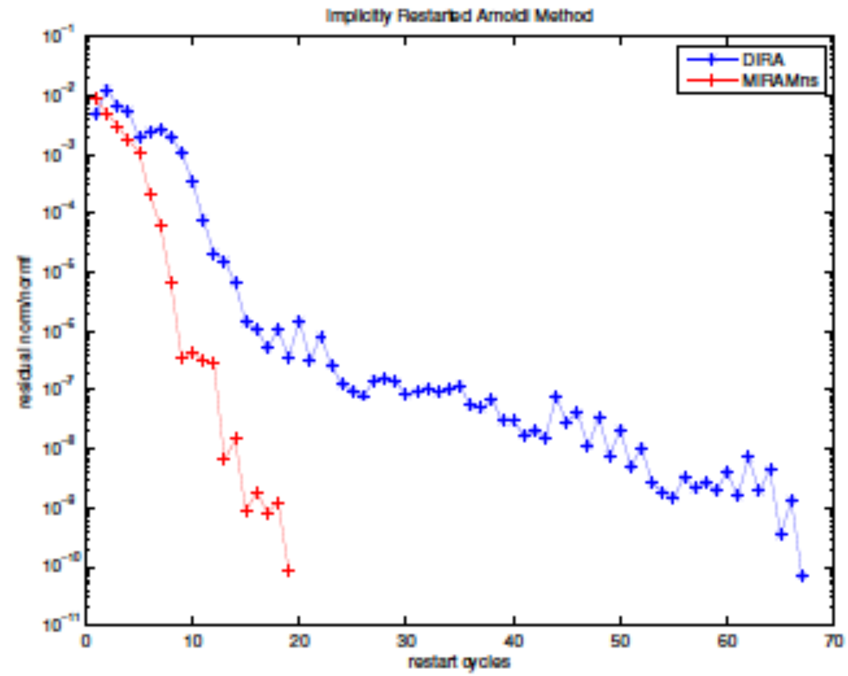
af23562, k=10 with a buffer of q extra vectors (tick parameter q)

Comparison of MIRAM with DIRA (m)

Dookhitram et al.



(a) MIRAMns(15, 17, 19), $m = 19$, $k = 6$



(b) MIRAMns(17, 19, 21, 23, 25), $m = 25$, $k = 10$

Bfw782a, $t_n=(1,1,0,\ \ldots,0)^T$ and 10^{-10}

PLAN DE COURS

1. Introduction
2. Quel paradigmes de programmation?
3. Approche « *unir & conquérir* »
4. Méthodes: Multiple Krylov Subspace (MKS)
5. Quelques résultats expérimentaux
6. Zoom sur certaines méthodes itératives pour des problèmes d'AL de grande taille (TD/TP)
7. Zoom sur l'hybridation de ces méthodes
8. Programmation orientée graphes

PLAN DE COURS

Annexe: Big Data

Big Data and HPC

Nothing tends so much to the advancement of knowledge as the application of a new instrument. The native intellectual powers of men in different times are not so much the causes of different success of their labors, as the peculiar nature of the means and artificial resources in their possession.
Humphrey Davy (1778 – 1829)

Elements of Chemical Philosophy (1812), in J. Davy (ed.), The Collected Works of Sir Humphry Davy(1839-40), Vol. 4, 37.

Goal of many: Having most powerful scientific tools
Computer models allows

Big Data and HPC

Computer modeling is the mirror of life:

- *High energy particle accelerators (discovery of Higgs boson, etc.)*
- *Astronomy instruments (Hubble Space telescope with the insights into the universe expansion and dark energy)*
- *High-throughput DNA sequencers (exploration of metagenomics ecology, etc.)*
- *Etc.*

Depend on computing (sensor control, data processing, access, etc.)

PLAN DE COURS

Méthode QR

Méthodes des puissances & puissances inverses

Méthodes de Lanczos & bi-Lanczos

Méthode des itérations simultanées

Méthodes de moindres carrées

Décomposition en valeurs singulières

Arnoldi (ERAM/IRAM)

Davidson