



**CALCUL
HAUTE
PERFORMANCE
SIMULATION**

ÉTABLISSEMENTS FONDATEURS

UNIVERSITÉ DE
VERSAILLES
ST-QUENTIN-EN-YVELINES



école
normale
supérieure
paris-saclay
UNIVERSITÉ PARIS-SACLAY

TELECOM
SudParis

instn

Méthodes et Programmation Numériques Avancées

Master CHPS, parcours IHPS

chps.uvsq.fr
(2018-2019)



Laboratoire
d'Informatique
Parallélisme
Réseaux
Algorithmique
Distribuée



MAISON DE LA SIMULATION

UNIVERSITÉ DE
VERSAILLES
ST-QUENTIN-EN-YVELINES



université PARIS-SACLAY

PROBLEM TO SOLVE

Let A be a very large and sparse matrix

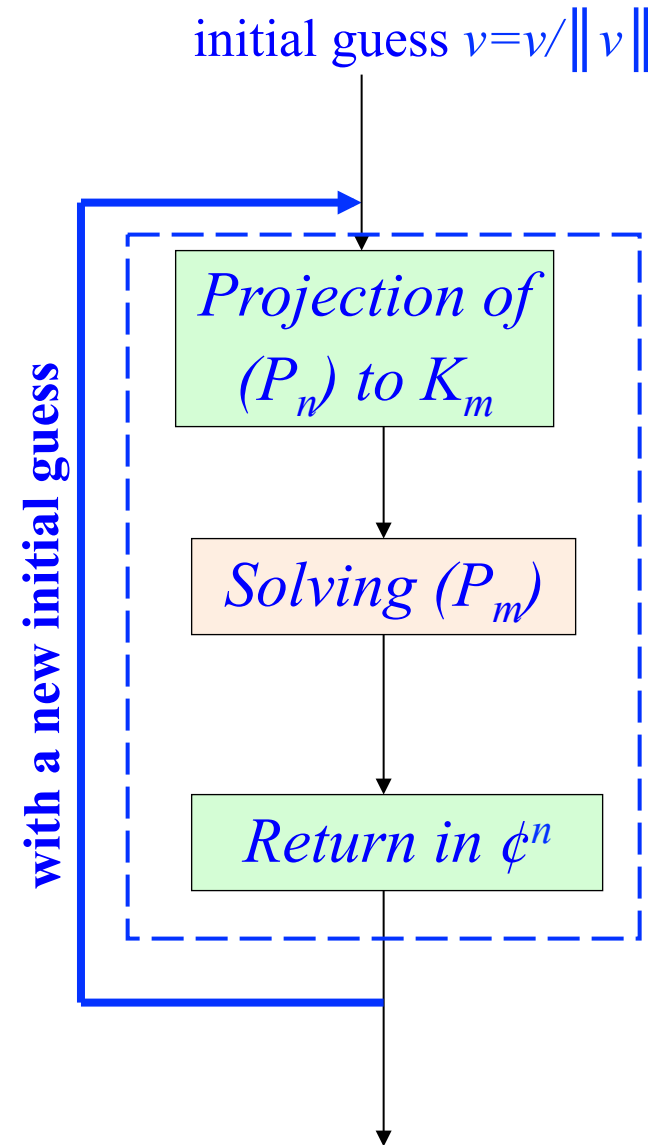
$$A \in \mathbb{C}^{n \times n}, b \in \mathbb{C}^{n \times n} \text{ seek } x \in \mathbb{C}^n: Ax=b$$

(P_n)

$$A \in \mathbb{C}^{n \times n}, \text{ seek a few } k \text{ Ritz pairs } \Lambda_k = (\lambda_1, \dots, \lambda_k) \in \mathbb{C}^k \\ \text{and } U_k = (u_1, \dots, u_k) \in \mathbb{C}^{n \times k}: Au_i = \lambda_i u_i \ (i=1, \dots, k)$$

RESTARTED KRYLOV SUBSPACE METHODS

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



SUJETS PROJETS

1. MÉTHODES DES PUISSANCES & PUISSANCES INVERSES
2. MÉTHODES DE LANCZOS & BI-LANCZOS
3. MÉTHODE DES ITÉRATIONS SIMULTANÉES
4. MÉTHODES DE MOINDRES CARRÉES
5. DÉCOMPOSITION EN VALEURS SINGULIÈRES
6. ARNOLDI (ERAM/IRAM)
7. DAVIDSON
8. MÉTHODE PADÉ-RAYLEIGH-RITZ (PRR)

MÉTHODE DE LA PUISSANCE (P1)

Soit $v^{(0)}$ un vecteur non nul. L'idée de base est de générer une suite de Vecteurs $A^k v^{(0)}$:

ALGORITHM : 1 ■ *The Power Method*

1. *Choose a nonzero initial vector $v^{(0)}$.*
2. *For $k = 1, 2, \dots$, until convergence, Do:*
3. $v^{(k)} = \frac{1}{\alpha_k} A v^{(k-1)}$ *where*
4. $\alpha_k = \operatorname{argmax}_{i=1, \dots, n} |(A v^{(k-1)})_i|$
5. *EndDo*

α_k est la plus grande –en module – des composantes de $A v^{(k-1)}$. Dans la pratique, on normalise le vecteur $v_k = A v^{(k-1)}$.

MÉTHODE DE LA PUISSANCE (ET MP SHIFTÉ)

- Supposons que λ_1 , la plus grande (en module) vp de A , est simple et que $v^{(0)}$ est un vecteur non nul qui n'est pas vecteur propre associé à λ_1 . On peut montrer alors que α_k tend vers λ_1 et v_k tend vers le vecteur propre associé à λ_1 . La vitesse de convergence est définie par le ratio: $|\lambda_2|/|\lambda_1|$.
- Cette méthode ne fonctionne pas pour le cas où 1 et -1 sont les valeurs propres dominantes en module de A . Dans ce cas, il suffit de considérer la matrice $I+A$ qui a une seule vp dominante -2- et les mêmes vecteurs propres que A .
- Plus généralement, on peut appliquer PM à la matrice $B(\sigma)=A+\sigma I$ au lieu de A (PM shifté). Quel est le meilleur shift σ ?

MÉTHODE DE LA PUISSANCE SHIFTÉ

Quand toutes vps sont réelles et vérifient:

$$\lambda_1 > \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_{n-1} \geq \lambda_n$$

Alors, la valeur de donnant la meilleure convergence est:

$$\sigma_{\text{optimal}} = (\lambda_2 + \lambda_n)/2$$

MÉTHODE DES ITÉRATIONS INVERSES (P2)

$$Au_i = \lambda_i u_i \text{ est équivalent à } A^{-1}u_i = (1/\lambda_i)u_i$$

A^{-1} a les mêmes vecteurs propres que A . Les valeurs propres de A^{-1} sont l'inverses de celles de A .

- En appliquant PM à la matrice de A^{-1} , on calcule les vps les plus proches de zéro.
- En appliquant PM à la matrice $(A - \sigma I)^{-1}$, on calcule les valeurs propres les plus proches de σ .

Méthode performante mais nécessite une décomposition de la matrice (LU ou autres).

MÉTHODE DES ITÉRATIONS INVERSES (P2)

L'idée principale est d'appliquer la méthode de la puissance (ou Arnoldi ou Lanczos ou MIS) à la matrice $C=(A-\sigma I)^{-1}$ au lieu de la matrice A . La matrice C ne doit pas être calculée explicitement. À chaque fois qu'on a besoin d'appliquer C à un vecteur, nous résolvons un système linéaire

$$C^{-1}B = (A-\sigma I)x=y.$$

Soit $B=A-\sigma I=LU$, alors pour résoudre $Bx=y$, il suffit de résoudre $Lz=y$ et $Ux=z$.

GÉNÉRALISATION DE PM: MIS

Méthode des itérations de sous-espaces (orthogonalisation des X_k):

- 1. Start:** $Q_0 = [q_1, \dots, q_m]$
- 2. Iterate:** *Until convergence do,*
- 3.** $X := AQ_{k-1}$
- 4.** $X = Q_k R$ (*QR factorization*)
- 5. EndDo**

Méthode des itérations de sous-espaces:

MÉTHODE DES ITÉRATIONS SIMULTANÉES (P3)

Start: Choose $Q_0 = [q_0, \dots, q_m]$

Iterate: For $k = 1, \dots$, **until convergence do:**

Compute $\hat{Z} = AQ_{k-1}$.

$\hat{Z} = ZR_Z$ (**QR factorization**) Les vecteurs de Z orthonormalisent ceux de \hat{Z}

$$B = Z^H A Z$$

Compute the Schur factorization $B = YRY^H$ Y: les vecteurs de Schur de B

$$Q_k = ZY \quad \text{La nouvelle matrice « initiale »}$$

EndDo

Supposons:

$$|\lambda_1| \geq |\lambda_2| \geq \dots |\lambda_m| > |\lambda_{m+1}| \geq \dots \geq |\lambda_n|,$$

Alors, la vitesse de convergence pour λ_1 est: $|\lambda_{m+1}/\lambda_1|$

ARNOLDI PROJECTION/FACTORIZATION

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

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

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

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

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

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

*Gramm-Schmidt
orthogonalization
process*

$$AV_m = V_m H_m + f_m e_m^T$$

with

$$f_m = h_{m+1,m} v_{m+1}$$

Eigenproblem in the subspace

$$H_m y_i = \lambda_i y_i \text{ for } i=1, \dots, m$$

(P_m)

ARNOLDI PROJECTION/FACTORIZATION

ALGORITHM : 6 ■ *Arnoldi's procedure*

For $j = 1, \dots, m$ ***do***

Compute $w := Av_j$

For $i = 1, \dots, j$, ***do***
$$\begin{cases} h_{i,j} := (w, v_i) \\ w := w - h_{i,j}v_i \end{cases}$$

$h_{j+1,j} = \|w\|_2; \quad v_{j+1} = w/h_{j+1,j}$

End

PROBLEM IN KSM: $K_M = K(A, V)$

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

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

Problem with clustered eigenvalues

MÉTHODES D'ARNOLDI (ERAM) – (P4)

ALGORITHM : 7 ■ Restarted Arnoldi (computes rightmost eigenpair)

1. Start: *Choose an initial vector v_1 and a dimension m .*
2. Iterate: *Perform m steps of Arnoldi's algorithm.*
3. Restart: *Compute the approximate eigenvector $u_1^{(m)}$*
4. *associated with the rightmost eigenvalue $\lambda_1^{(m)}$.*
5. *If satisfied stop, else set $v_1 \equiv u_1^{(m)}$ and goto 2.*

MÉTHODES D'ARNOLDI (IRAM) – (P5)

Calculer une étape d'Arnoldi: $AV_m = V_m H_m + f_m e_m^T$

Calculer les valeurs et vecteurs de Ritz recherchés

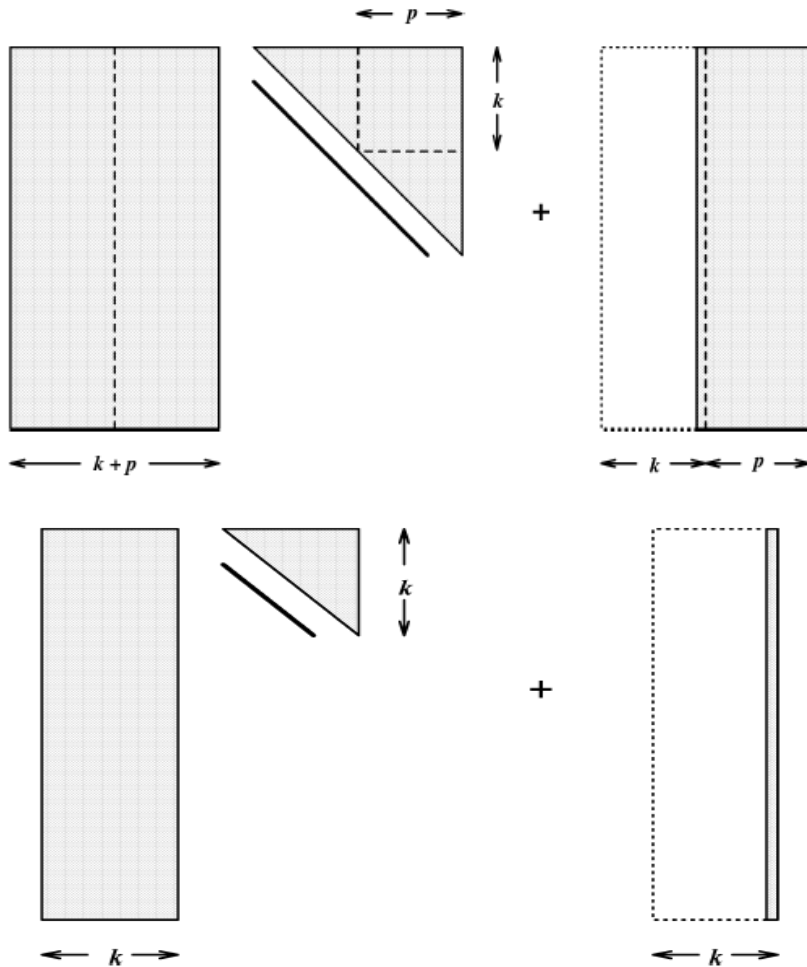
En cas de non-convergence, garder les k premières colonnes de l'équation ci-dessus:

16

$$AV_m = V_m H_m + f_m e_m^T$$

et la compléter pour obtenir la factorization complète.

IRAM (M=K+P)



$$V_{k+p} Q Q^T H_{k+p} Q + f_{k+p} e_{k+p}^T Q$$

after p implicitly shifted QR

$$V_k H_k + f_k e_k^T$$

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

Beginning with $AV_k^+ = V_k^+ H_k^+ + f_k^+ e_m^T$, complete an m -step

$$AV_m = V_m H_m + f_m e_m^T$$

MÉTHODE DE LANCZOS SYMÉTRIQUE (P6)

Méthode d'Arnoldi appliquée à une matrice hermitienne ($A=A^H$) et $V_m^H A V_m = H_m$ donc $H_m = H_m^H$. On a alors:

$$H_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \alpha_3 & \beta_4 & \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & \cdot \\ & & & & \beta_m & \alpha_m \end{pmatrix}$$

Avec une récurrence à 3 termes:

$$\beta_{j+1} v_{j+1} = A v_j - \alpha_j v_j - \beta_j v_{j-1}$$

MÉTHODE DE LANCZOS SYMÉTRIQUE (P6)

ALGORITHM : 8. *Lanczos*

1. **Choose an initial vector v_1 of norm unity. Set $\beta_1 \equiv 0, v_0 \equiv 0$**
2. **For $j = 1, 2, \dots, m$ Do:**
3. $w_j := Av_j - \beta_j v_{j-1}$
4. $\alpha_j := (w_j, v_j)$
5. $w_j := w_j - \alpha_j v_j$
6. $\beta_{j+1} := \|w_j\|_2$. **If $\beta_{j+1} = 0$ then Stop**
7. $v_{j+1} := w_j / \beta_{j+1}$
8. **EndDo**

MÉTHODE PRR- CAS SYMÉTRIQUE (P7)

Iterative PRR algorithm

Step 1. Choice of m .

Step 2. Choice of initial vector x .

Step 3. Normalization of x : $y_0 = x / \|x\|$, $C_0 = \|y_0\|^2 = 1$.

Step 4. Computation of C_1, \dots, C_{2m-1} .

- For $k = 1, m - 1$, do

$$C_{2k-1} = (y_k, y_{k-1})$$

$$C_{2k} = (y_k, y_k)$$

$$y_{k+1} = Ay_k$$

- End for k
- $C_{2m-1} = (y_m, y_{m-1})$

Step 5. Linear system solving (63).

Step 6. Computation of the roots of $Q_m(\lambda)$ polynomial.

Step 7. Computation of the Ritz vectors $u_i^{(m)}$ by (66) and (67).

Step 8. If $(\max_{1 \leq i \leq r} \|(A - \lambda_i^{(m)}I)u_i^{(m)}\| \leq \text{requested precision})$ then stop, otherwise with a new initial vector go to step 3.

MÉTHODE BI-LANZOS (P8)

ALGORITHM : 9. The Lanczos Bi-Orthogonalization Procedure

1. **Choose** v_1, w_1 **such that** $(v_1, w_1) = 1$. **Set** $\beta_1 = \delta_1 \equiv 0, w_0 = v_0 \equiv 0$
2. **For** $j = 1, 2, \dots, m$ **Do:**
3. $\alpha_j = (Av_j, w_j)$ [$\alpha_j = (Av_j - \beta_j v_{j-1}, w_j)$]
4. $\hat{v}_{j+1} = Av_j - \alpha_j v_j - \beta_j v_{j-1}$ [$\hat{v}_{j+1} = (Av_j - \beta_j v_{j-1}) - \alpha_j v_j$]
5. $\hat{w}_{j+1} = A^H w_j - \bar{\alpha}_j w_j - \delta_j w_{j-1}$ [$\hat{w}_{j+1} = (A^H w_j - \delta_j w_{j-1}) - \bar{\alpha}_j w_j$]
6. $\delta_{j+1} = |(\hat{v}_{j+1}, \hat{w}_{j+1})|^{1/2}$. **If** $\delta_{j+1} = 0$ **Stop**
7. $\beta_{j+1} = (\hat{v}_{j+1}, \hat{w}_{j+1}) / \delta_{j+1}$
8. $w_{j+1} = \hat{w}_{j+1} / \bar{\beta}_{j+1}$
9. $v_{j+1} = \hat{v}_{j+1} / \delta_{j+1}$
10. **EndDo**

MÉTHODE BI-LANZOS (P8)

Cette méthode calcule une paire de bases bi-orthogonales pour les deux sous espaces de Krylov :

$$K_m(A, v_1) \text{ et } K_m(A, w_1).$$

Il existe beaucoup de choix pour δ_{j+1} et β_{j+1} des lignes 6 et 7 de l'algorithme précédent. La seule contrainte est que:

$$\delta_{j+1}\beta_{j+1} = (\hat{v}_{j+1}, \hat{u}_{j+1})$$

Soit:

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \delta_2 & \alpha_2 & \beta_3 & & \\ & \cdot & \cdot & \cdot & \\ & & \delta_{m-1} & \alpha_{m-1} & \beta_m \\ & & & \delta_m & \alpha_m \end{pmatrix}$$

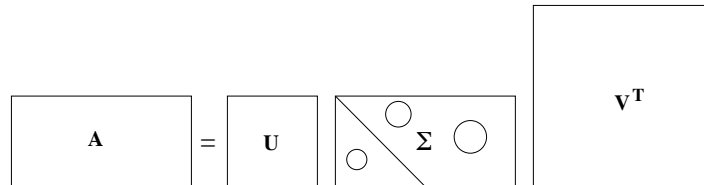
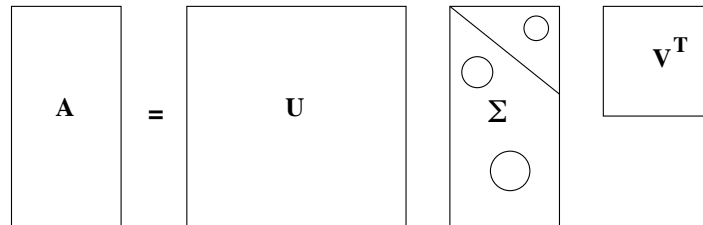
v_i dans $K_m(A, v_1)$ et w_j dans $K_m(A, w_1)$

DÉCOMPOSITION SVD (P9)

Pour toute matrice rectangulaire A de n lignes et de m colonnes, il existent deux matrices orthogonales U d'ordre n et V d'ordre m tel que:

$$A = U \Sigma V^T$$

Où Σ est une matrice diagonale avec des valeurs non-négatives sur la diagonale



MÉTHODE DE DAVIDSON(P9)

ALGORITHM : 11 ■ Davidson's method (Real symmetric case)

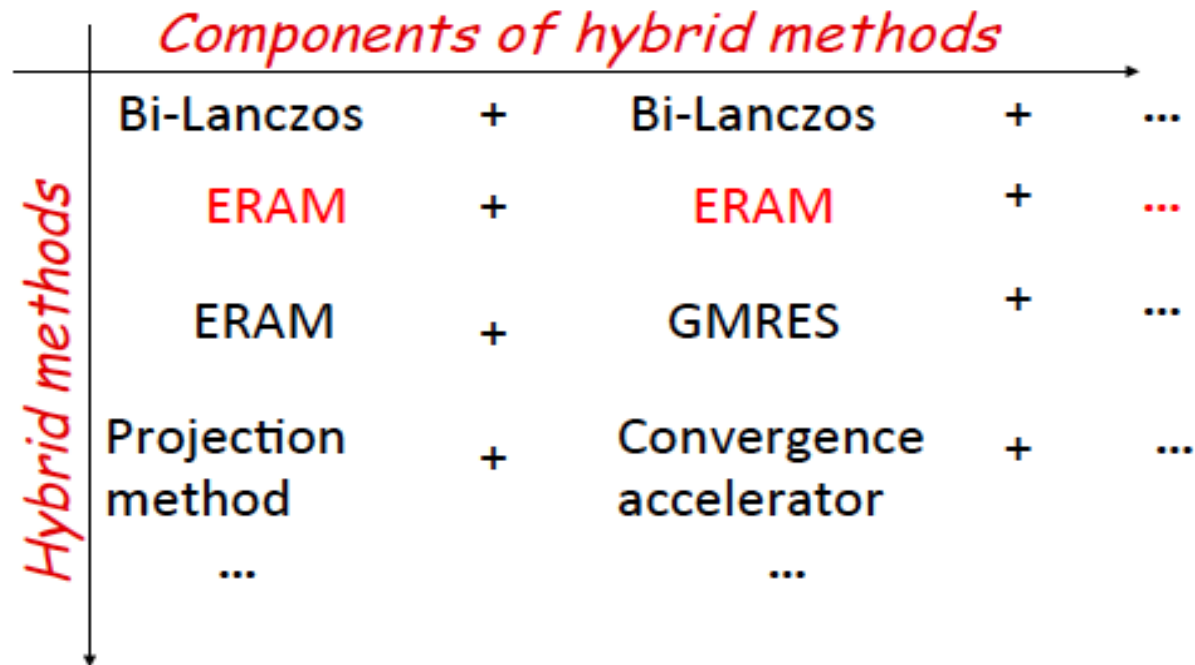
1. **Choose an initial unit vector v_1 . Set $V_1 = [v_1]$.**
2. **Until convergence Do:**
3. **For $j = 1, \dots, m$ Do:**
4. $w := Av_j$.
5. **Update $H_j \equiv V_j^T AV_j$**
6. **Compute the smallest eigenpair μ, y of H_j .**
7. $z := V_j y \quad r := Az - \mu z$
8. **Test for convergence. If satisfied Return**
9. **If $j < m$ compute $t := M_j^{-1} r$**
10. **compute $V_{j+1} := ORTHN([V_j, t])$**
11. **EndIf**
12. **Set $v_1 := z$ and go to 3**
13. **EndDo**
14. **EndDo**

UNITE AND CONQUER APPROACH

Méthodes hybrides:

*Une combinaison des méthodes classiques de
projection avec redémarrages*

UNITE AND CONQUER APPROACH

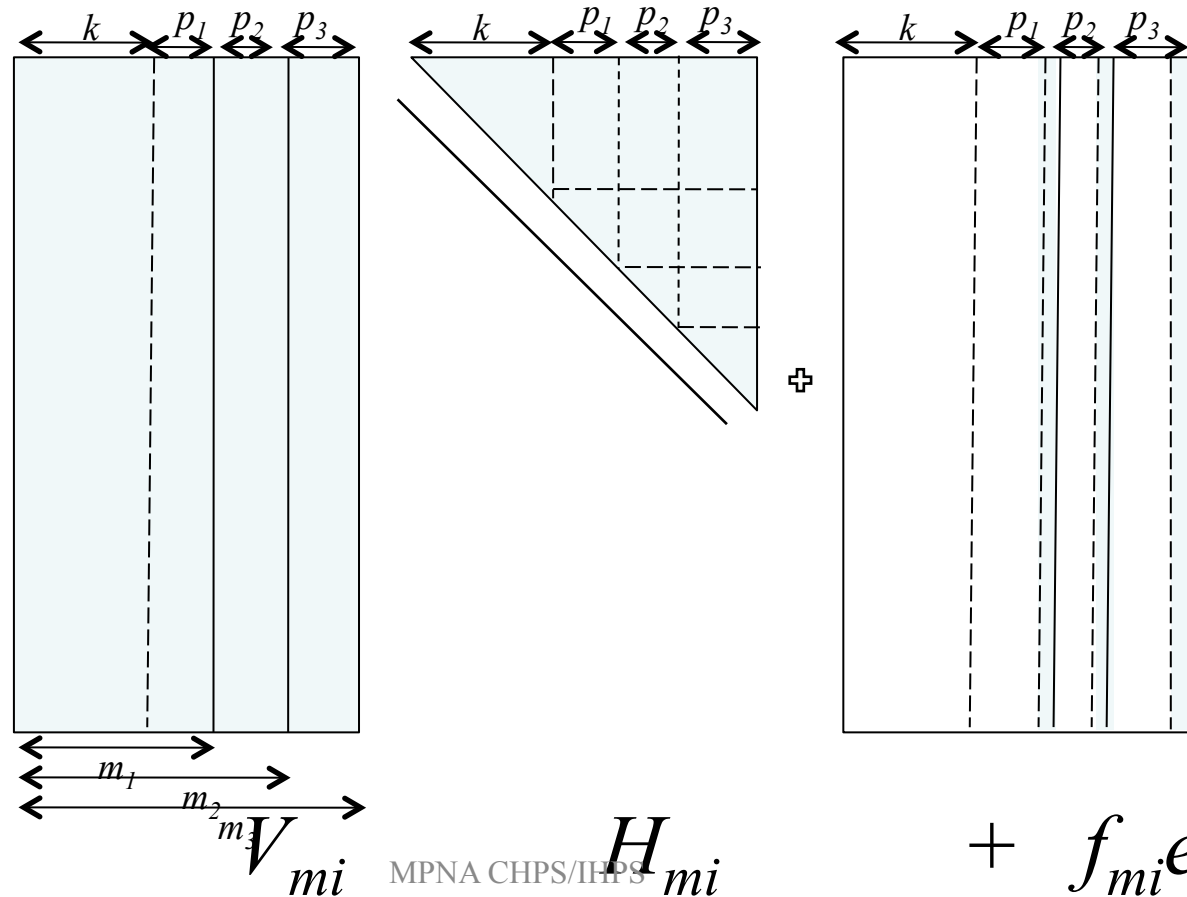


Saad (Chebyshev acceleration techniques for solving nonsymmetric eigenvalue; 1984), *Brezinski* (hybrid procedures for solving linear systems; 1994), Code coupling (in simulation), ...

MIRAM WITH NESTED SUBSPACES

$$K_{ml} = \text{span}(v_l, Av_l, \dots, A^{m_l-1}v_l, A^{m_l}v_l, \dots, A^{m_2-1}v_l, A^{m_2}v_l, \dots, A^{m_3-1}v_l, \dots, A^{m_l-1}v_l)$$

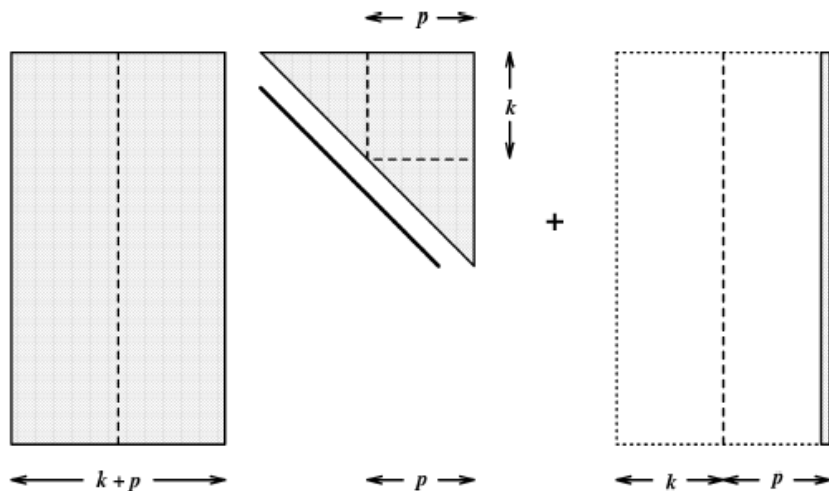
$$K_{m1} \subset K_{m2} \subset \dots \subset K_{ml}$$



MIRAM WITH NESTED SUBSPACES

$$\begin{array}{lcl}
 AV_{m1} = V_{m1}H_{m1} + f_{m1}e^T_{m1} & \Longrightarrow & (\Lambda_k, U_k)_{m1} \\
 AV_{m2} = V_{m2}H_{m2} + f_{m2}e^T_{m2} & \Longrightarrow & (\Lambda_k, U_k)_{m2} \\
 \dots & & \dots \\
 AV_{ml} = V_{ml}H_{ml} + f_{ml}e^T_{ml} & \Longrightarrow & (\Lambda_k, U_k)_{ml}
 \end{array}
 \left. \vphantom{\begin{array}{l} \\ \\ \end{array}} \right\} \Downarrow (\Lambda_k, U_k)_{m_best}$$

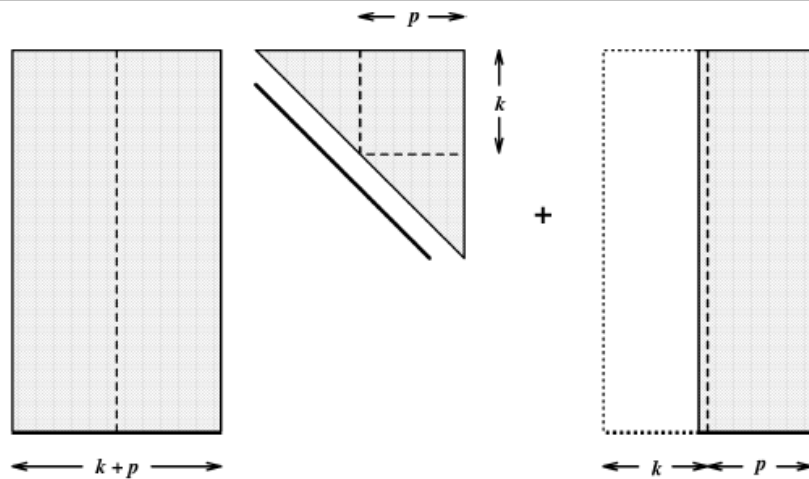
$$m_i = k + p_i \text{ for } i = 1, \dots, l$$



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

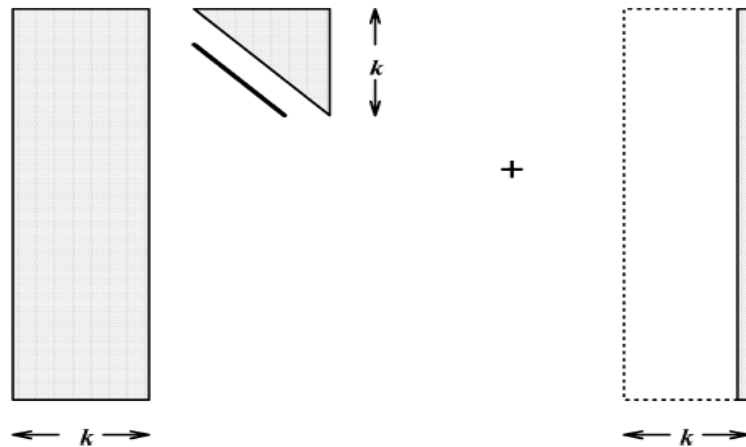
$$AV_{k+p} = V_{k+p}H_{k+p} + f_{k+p}e^T_{k+p}$$

MIRAM WITH NESTED SUBSPACES



$$V_{k+p} Q Q^T H_{k+p} Q + f_{k+p} e_{k+p}^T Q$$

after p implicitly shifted QR



$$V_k H_k + f_k e_k^T$$

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

Beginning with $AV_k^+ = V_k^+ H_k^+ + f_k^+ e_{m_i}^T$, l m_i -step AF

$$AV_{m_i} = V_{m_i} H_{m_i} + f_{m_i} e_{m_i}^T \quad \text{for } i=1, \dots, l$$