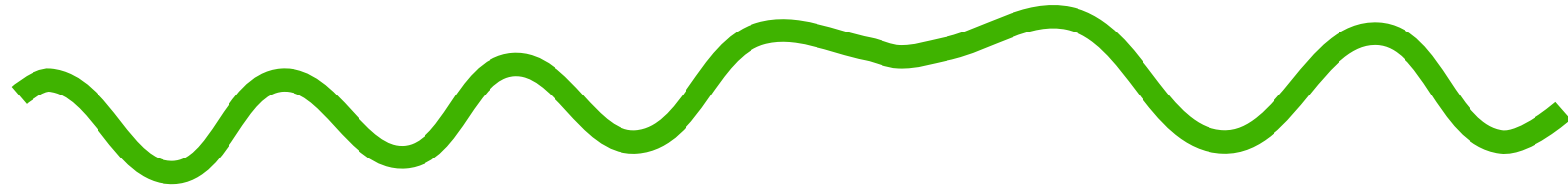


Bibliothèques scientifiques parallèles
Outils pour le calcul scientifique à haute performance
École doctorale sciences pour l'ingénieur
juin 2001

Philippe MARQUET

phm@lifl.fr

Laboratoire d'informatique fondamentale de Lille
Université des sciences et technologies de Lille



- ~ Ce cours est diffusé sous la licence GNU Free Documentation License,
<http://www.gnu.org/copyleft/fdl.html>
- ~ La dernière version de ce cours est accessible à partir de
<http://www.lifl.fr/west/courses/cshp/>
- ~ \$Id: bibsp.tex,v 1.6 2001/11/02 13:08:26 marquet Exp \$

Table des matières



- ~ Motivations
- ~ Bibliothèques monoprocesseur
- ~ Bibliothèques parallèles
- ~ Principes d'utilisation des bibliothèques parallèles
- ~ Conclusion

Remerciements



Cette présentation est essentiellement basée sur

↪ *Les bibliothèques scientifiques parallèles*

J. CHERGUI, D. GIROU, G. GRASSEAU, J.-Ph. PROUX
IDRIS, avril 2000

http://www.idris.fr/data/cours/parallel/para_b/

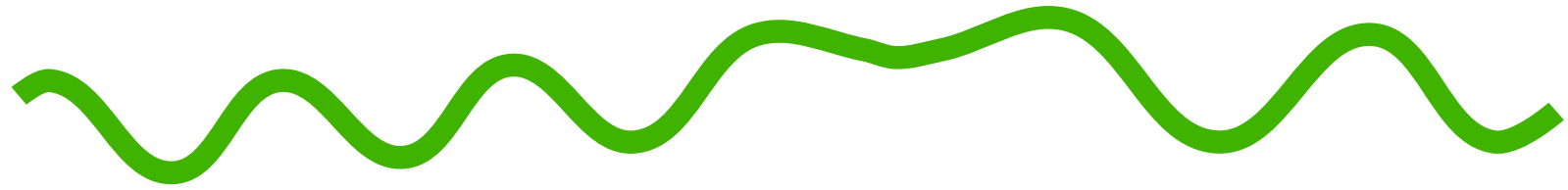
↪ *Ordinateurs et calcul parallèles*

chapitre *Bibliothèques pour le calcul numérique parallèle : synthèse et perspectives*

Serge PETITON et Bernard TOURANCHEAU
OFTA, 1997

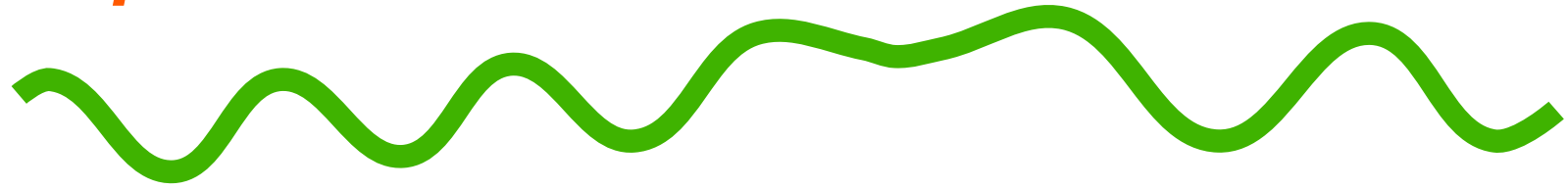
↪ *Initiation au parallélisme*

Marc GENGLER, Stéphane UBÉDA, et Frédéric DESPREZ
Masson, 1997



Motivations

Bibliothèques scientifiques parallèles



✓ Calcul scientifique – calcul numérique

- ✓ algèbre linéaire dense
- ✓ algèbre linéaire creuse
- ✓ résolution linéaire directe
- ✓ résolution linéaire itérative
- ✓ résolution non-linéaire
- ✓ valeurs propres
- ✓ vecteurs propres
- ✓ transformées de Fourier rapides
- ✓ maillages
- ✓ génération de nombres aléatoires
- ✓ etc.

✓ Haut niveau d'abstraction

- ✓ multiplication matrices

✓ Réduction du temps de développement

✓ Indépendance vis-à-vis de l'implantation

- ✓ séquentielle, ou
- ✓ parallèle

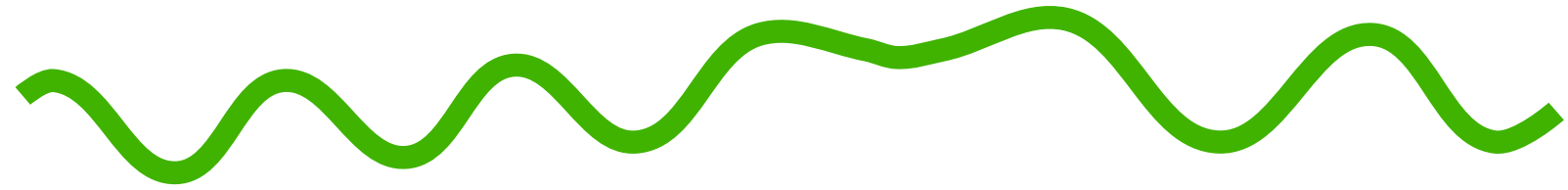
Situation idéale !

Portabilité et efficacité



Portabilité et performances

- ✓ généricité (indépendance vis-à-vis du type des données)
- ✓ indépendance vis-à-vis de la représentation des données
- ✓ choix du meilleur algorithme numérique
- ✓ assurance de la stabilité numérique
- ✓ optimisation fonction de l'architecture de la machine
- ✓ optimisation fonction des paramètres
(nombre de colonnes/lignes)



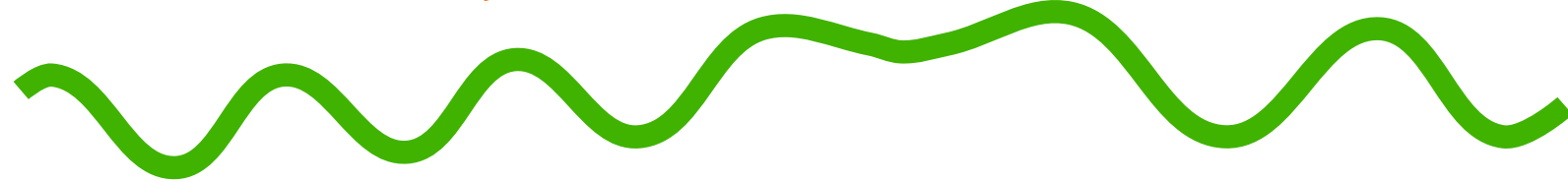
Bibliothèques monoprocesseur

Algèbre linéaire dense



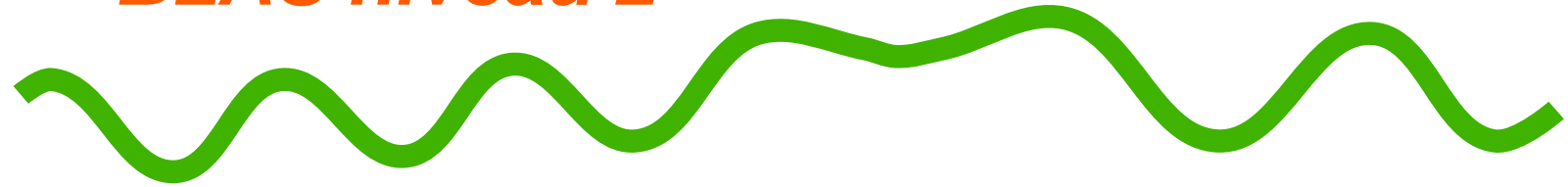
- ~ Bibliothèques séquentielles originelles (1974)
- ~ *LINPACK* : LINear algebra PACKage
- ~ *BLAS* : Basic Linear Algebra Subroutines
- ~ Robustesse
 - ~ largement utilisées, retour des utilisateurs
- ~ Performance
 - ~ implantation en assembleur des parties critiques
 - ~ optimisation au niveau des registres / caches
- ~ Succès

Les BLAS, BLAS niveau 1



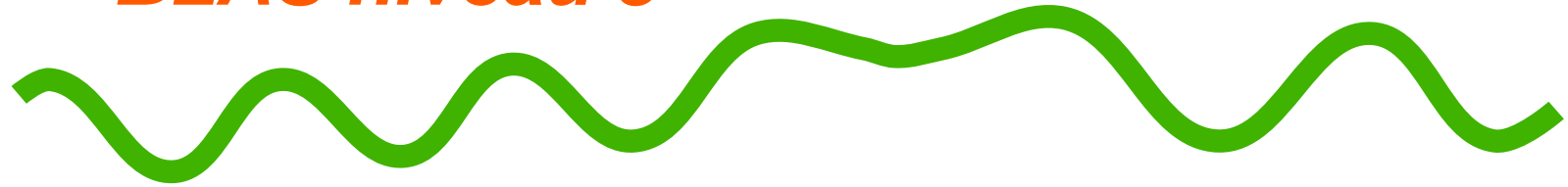
- ~ Bibliothèque *BLAS* divisée en trois sections
 - ~ *BLAS 1* : vecteur / vecteur
 - ~ *BLAS 2* : matrice / vecteur
 - ~ *BLAS 3* : matrice / matrice
- ~ *BLAS* niveau 1
 - ~ première bibliothèque de calcul scientifique
 - ~ blocs de base des algorithmes d'algèbre linéaire
 - ~ opérations de type $z = \alpha x + y$ (α scalaire, x , y , et z vecteurs)
 - ~ traitement élément par élément des éléments de vecteurs
 - ~ grand succès
 - ~ exemple de nommage : SAXPY
 - ~ S : réel simple précision (D double, etc.)
 - ~ AXPY : $\alpha x + y$!
 - ~ $\mathcal{O}(n)$ opérations sur $\mathcal{O}(n)$ éléments

BLAS niveau 2



- ✔ Motivés par
 - ✔ diminution du nombre d'accès mémoire
 - ✔ augmentation du grain de calcul
 - ✔ prise en compte des unités vectorielles pipelines
- ✔ Opération entre matrices et vecteurs
 - ✔ produit matrice-vecteur SGEMV
 - $y = \beta y + \alpha Ax$ (A matrice)
 - ✔ $\mathcal{O}(n^2)$ opérations pour $\mathcal{O}(n^2)$ éléments

BLAS niveau 3



- ✔ Motivés par
 - ✔ utilisation maximale des hiérarchies mémoire
 - ✔ minimisation des mouvements de données entre les mémoires
- ✔ Opération entre matrices et matrices
 - ✔ produit matrice-matrice SGEMM
 - $C = \beta C + \alpha AB$ (A, B, C matrices)
 - ✔ $\mathcal{O}(n^3)$ opérations pour $\mathcal{O}(n^2)$ éléments

Performances des BLAS



BLAS niveau 1

- ✓ $\mathcal{O}(n)$ opérations pour $\mathcal{O}(n)$ éléments
- ✓ nombreux accès aux données par rapport aux calculs
- ✓ pas de bénéfice de la hiérarchie mémoire
- ✓ pas de bénéfice des registres et unités vectoriels

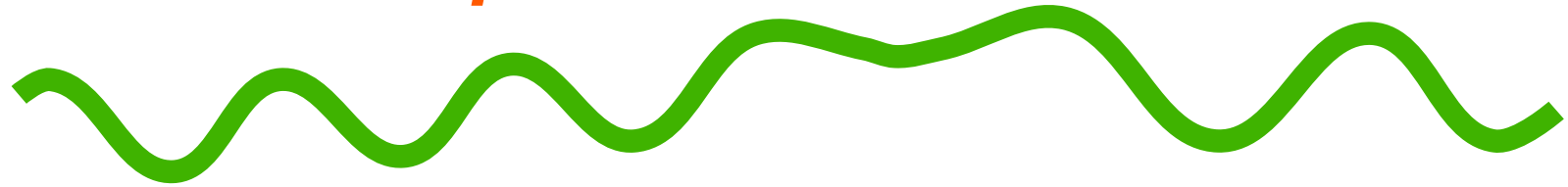
BLAS niveau 2

- ✓ $\mathcal{O}(n^2)$ opérations pour $\mathcal{O}(n^2)$ éléments
- ✓ premiers bénéfiques de registres vectoriels
- ✓ performances limitées dues aux hiérarchies mémoires

BLAS niveau 3

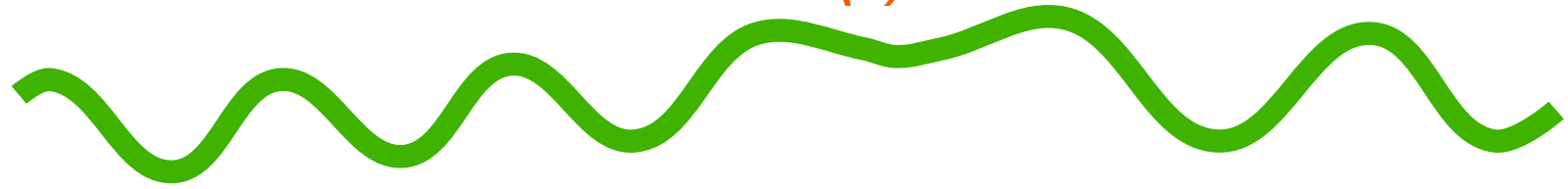
- ✓ $\mathcal{O}(n^3)$ opérations pour $\mathcal{O}(n^2)$ éléments
- ✓ meilleure utilisation des caches
- ✓ performances proches des performances de crête

Bibliothèque LAPACK



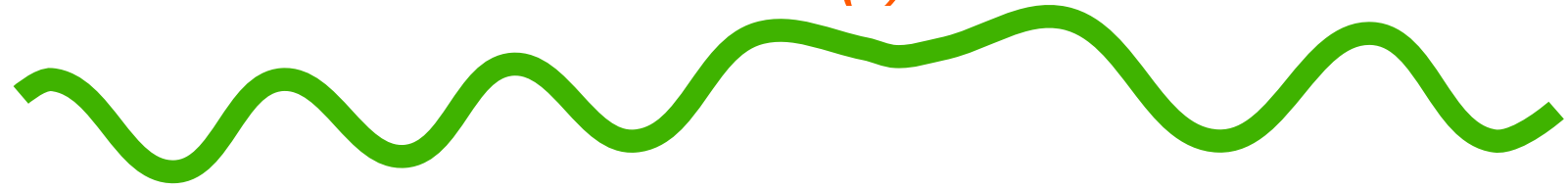
- ✓ Algèbre linéaire de haut niveau
 - ✓ résolution de systèmes linéaires
 - ✓ résolution des moindres carrés
 - ✓ valeurs propres, valeurs singulières
 - ✓ factorisation de matrice LU , Cholesky, QR , Schur, Schur généralisée
- ✓ Travail sur des matrices denses ou des matrices bandes
 - Pas de prise en compte de matrices creuses générales
- ✓ Conçue pour les machines vectorielles
 - Étendue aux machines parallèles à mémoire partagée
- ✓ Utilise des algorithmes par blocs pour utiliser au mieux la hiérarchie mémoire
- ✓ Implantée au dessus des $BLAS$ 3

Structures creuses (1)

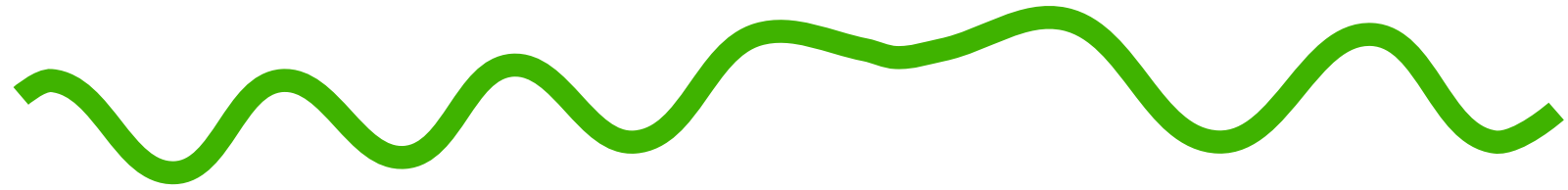


- ✓ Matrices creuses très répandues dans le calcul numérique
 - ✓ matrices issues de maillages liés aux méthodes à éléments finis
- ✓ Prise en compte de la structure creuse
 - ✓ taille mémoire réduite
 - ✓ nombre de calculs réduits
- ✓ Choix d'une représentation des matrices creuses
 - ✓ CSR : *compress sparse row*
 - ✓ éléments non-nuls de la matrice dans un tableau (nnz éléments)
 - ✓ un tableau des indices de colonnes dans la matrice des éléments non-nuls (nnz éléments)
 - ✓ un tableau des indices dans les tableaux précédents des premiers éléments de ligne (n éléments)

Structures creuses (2)

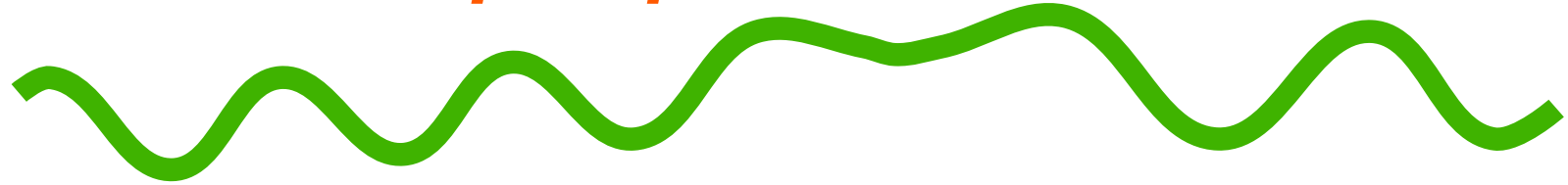


- ✓ Bibliothèque *SPARSKIT*
- ✓ NASA et université de Minneapolis
- ✓ Format CSR en interne
- ✓ Ensemble d'outils de manipulation de matrices creuses
 - ✓ transformation au format CSR
 - ✓ transformation depuis le format CSR
 - ✓ visualisation de matrices
 - ✓ etc.



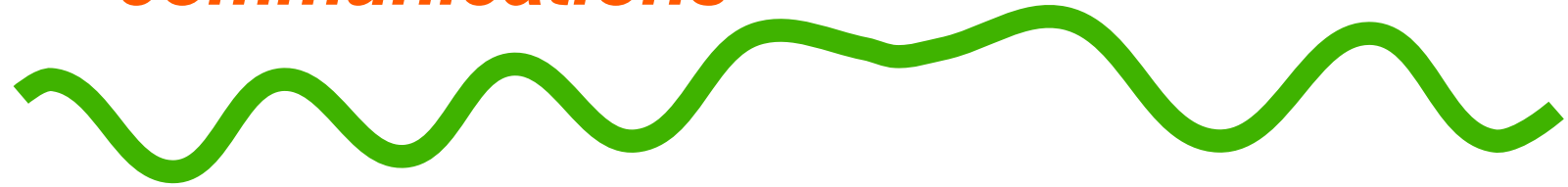
Bibliothèques parallèles

Bibliothèques parallèles



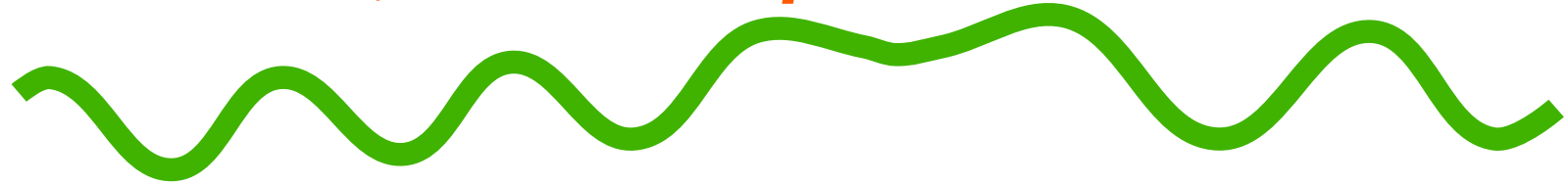
- ✓ Nécessité de versions parallèles des bibliothèques de calcul numériques
- ✓ Prise en compte de la localité des données
- ✓ Bonne répartition des données
 - ✓ équilibre de la charge de calcul
 - ✓ recouvrement des accès aux données distantes par des calculs
- ✓ Versions parallèles des algorithmes
 - ✓ jusqu'à de nouvelles méthodes numériques
- ✓ Proposer une bibliothèque parallèle
 - ✓ paralléliser chacune des fonctions de la bibliothèque
 - ✓ quel algorithme parallèle ?
 - ✓ quelle répartition des données
- ✓ Modèle de programmation SPMD d'utilisation des bibliothèques

BLACS, bibliothèque de communications



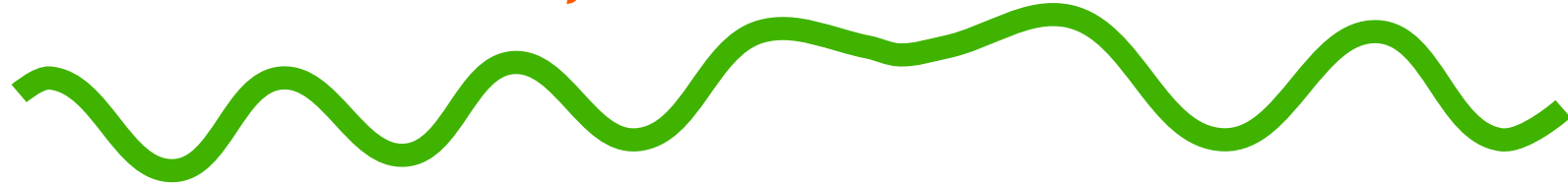
- ✓ Données distribuées sur des processeurs
⇒ communications entre ces processeurs
- ✓ Communications spécialisées
 - ✓ échange de vecteurs/matrices
 - ✓ lignes/colonnes de matrices : blocs de matrices
 - ✓ diffusion et réduction
 - ✓ contexte de communication : isoler les communications de la bibliothèque et d'autres communications
- ✓ Bibliothèque spécialisée de communications : *BLACS*
Basic Linear Algebra Communication Subroutines
 - ✓ portabilité (versions PVM et MPI)
 - ✓ efficacité (versions optimisées pour nombres de machines)
 - ✓ ce n'est pas une bibliothèque générale de communications

PBLAS, les BLAS parallèles



- ✓ *PBLAS* : Parallel Basic Linear Algebra Subroutines
- ✓ Quasi *BLAS* 1 à 3 pour machines à mémoires distribuées
- ✓ Interface aussi proche que possible de celle des *BLAS*
Mais différente !
 - ✓ descripteurs d'allocation des matrices
- ✓ Implantée au dessus de *BLAS* + *BLACS*
 - ✓ *BLAS* : calculs locaux
 - ✓ *BLACS* : communications
- ✓ Considérée comme un noyau pour d'autres bibliothèques numériques parallèles

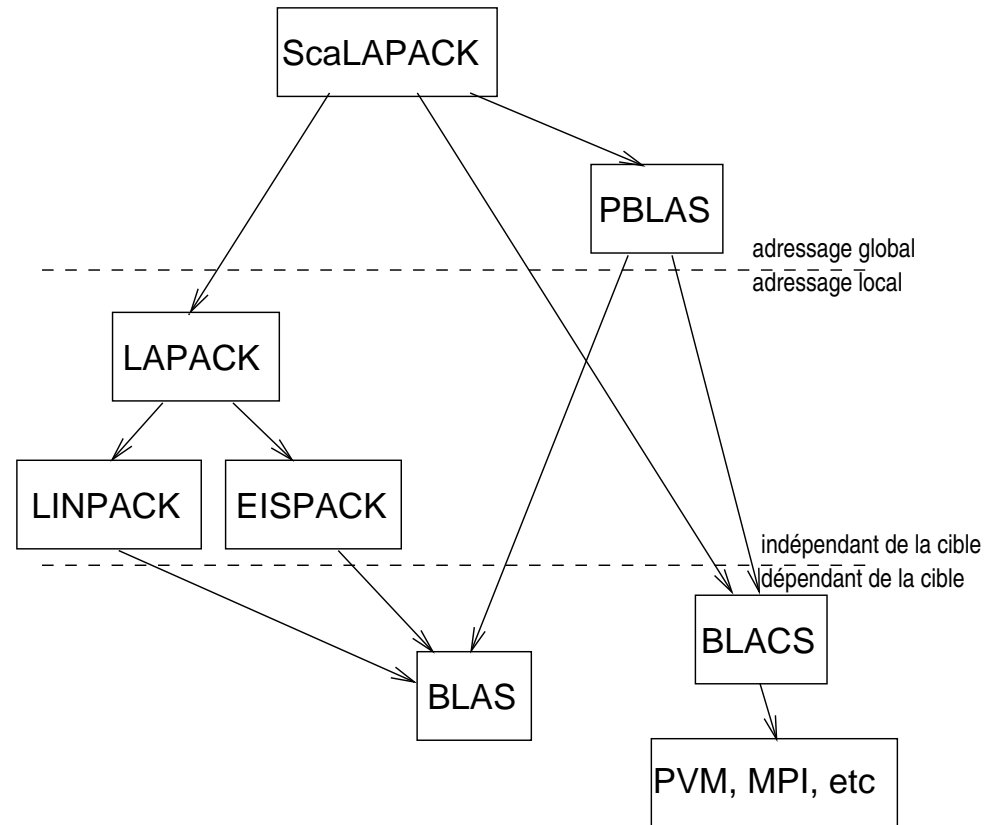
ScaLAPACK, scalable LAPACK



- ✓ La référence en terme d'algèbre linéaire pour machines à mémoires distribuées
- ✓ Projet de Jack DONGARRA (Oak Ridge Nat. Labs), 1989
- ✓ À la base portage de *LAPACK*
- ✓ Aujourd'hui composé de multiples bibliothèques
 - ✓ *ScaLAPACK* : Scalable LAPACK
algèbre linéaire pour les matrices denses
 - ✓ *PARPACK* : Parallel ARPACK
recherche de certaines valeurs propres sur de très grosses matrices
 - ✓ *CAPSS* : Cartesian Parallel Sparse Solver
résolution directe de systèmes linéaires sur matrices creuses définies positives symétriques
 - ✓ *PARPRE* : Parallel Preconditionners
préconditionneurs pour le traitement de matrices creuses quelconques

Structure de ScaLAPACK

- ~ ScaLAPACK est une bibliothèque domaine public
- ~ Implantée au dessus de *LAPACK*, *PBLAS* et *BLACS*
- ~ D'autres bibliothèques sont implantées au dessus de ScaLAPACK : *SciLib* sur SGI/Cray...



Succès de ScaLAPACK



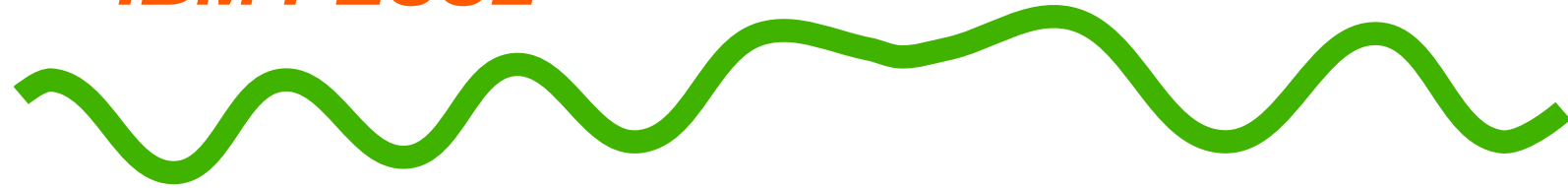
Extensibilité

- ✓ garantir les performances quand le nombre de processeurs augmente
- ✓ équilibre de la charge via des distributions cycliques 2-D
 - ✓ équilibre de charge pour des petits blocs (grain de distribution)
 - ✓ faible coût de communication pour des gros blocs (diminution du nombre de messages)
 - ✓ taille de blocs qui soit le bon compromis entre équilibre de la charge et latence des communications

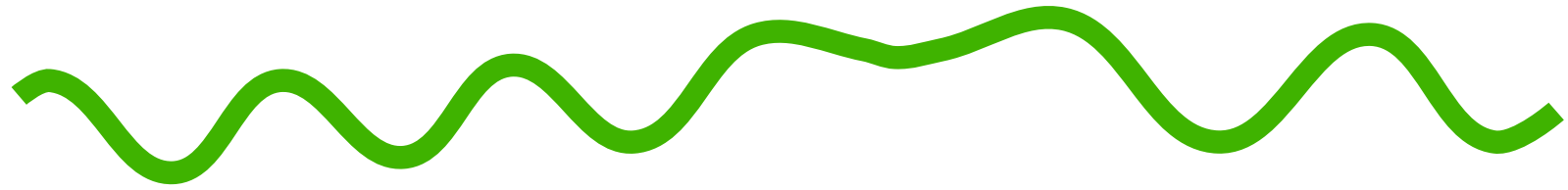
Stratégie d'un succès

- ✓ code source (Fortran 77 et C), donc portabilité
- ✓ structure hiérarchique (*BLAS*, *BLACS*...)
- ✓ évolution continue des bibliothèques de base
 - ✓ meilleurs algorithmes
 - ✓ optimisation pour la majorité des machines (assembleur)
 - ✓ rejaillit de suite sur *ScaLAPACK*

IBM PESSL

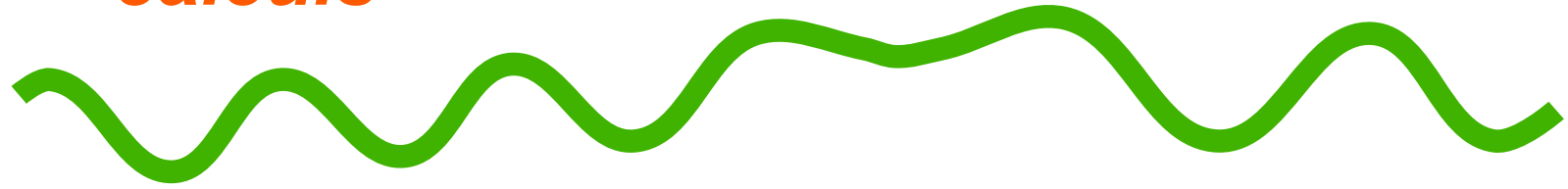


- ✓ *PESSL* : Parallel Engineering and Scientific Subroutines Library
- ✓ Bibliothèque mathématique d'IBM : calcul numérique haute performance sur IBM SP et clusters d'IBM RS/6000
- ✓ Algèbre linéaire, FFT, génération de nombres aléatoires...
 - ✓ *ESSL*
 - ✓ *BLAS* optimisée pour le RS/6000
 - ✓ sous-ensemble de *LAPACK*
 - ✓ *BLACS* versions MPI et version optimisée
 - ✓ sous-ensemble des *PBLAS* niveaux 2 et 3
 - ✓ sous-ensemble de *ScaLAPACK*
- ✓ Disponibilité d'une interface High Performance Fortran
 - ✓ tableaux distribués

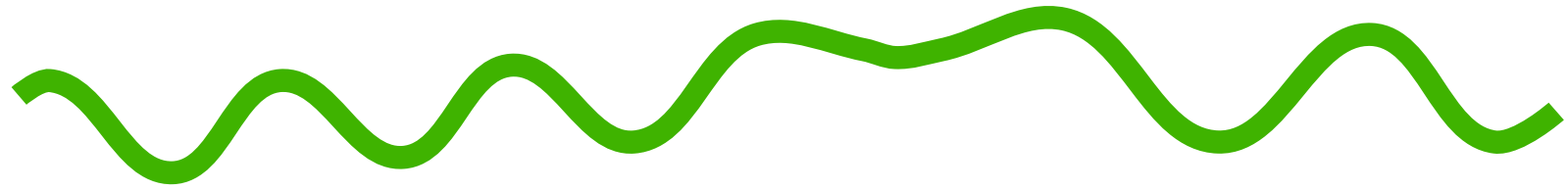


Principes d'utilisation des bibliothèques parallèles

Répartition des données et des calculs

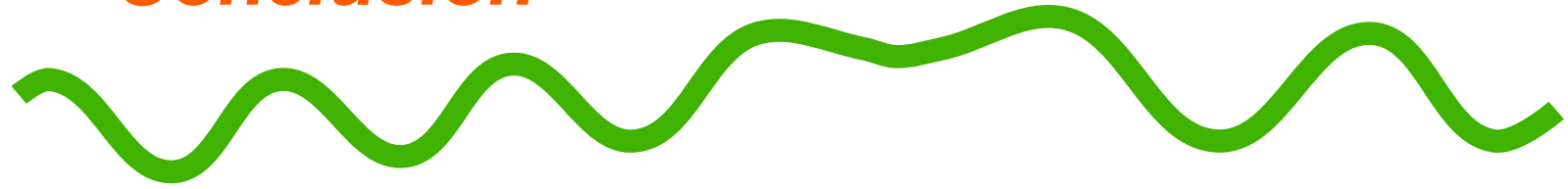


- ✓ Implique une structure de programme
- ✓ Déclarations
 - ✓ processeurs logiques
 - ✓ répartition des données : descripteurs de matrices
 - ✓ distribution des données sur les processeurs logiques
- ✓ Calculs
 - ✓ appels aux bibliothèques parallèles
- ✓ Redistribution des données
 - ✓ pour continuer les calculs
 - ✓ pour sauvegarder les résultats



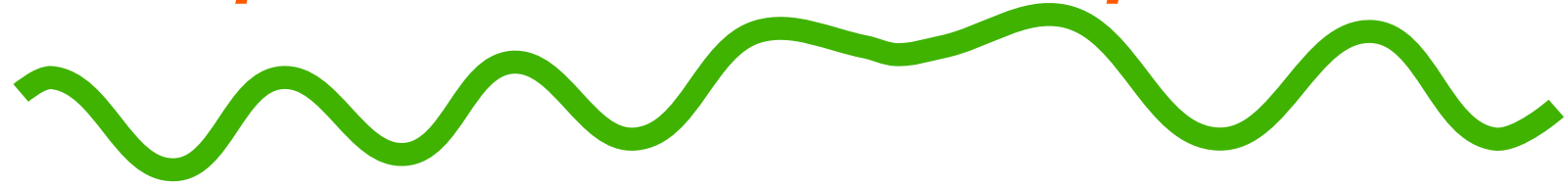
Conclusion

Conclusion



- ~ Parallélisation du cœur calculatoire d'un programme
- ~ Efficacité
- ~ Portabilité
- ~ Distribution des données
- ~ Appropriation de l'interface de la bibliothèque
- ~ Syntaxe, lisibilité !

Disponibilité des bibliothèques



- ✓ Bibliothèques « portables »
 - ✓ domaine public : *BLAS*, *LAPACK*, *ScaLAPACK*...
 - ✓ commerciale : *NAG*...
- ✓ Bibliothèques « constructeurs »
 - ✓ *ESSL* et *PESSEL* chez IBM
 - ✓ *SciLib* chez SGI/Cray
 - ✓ (devrait être) optimisée pour la machine
 - ✓ non portable !

Références



- ~ *BLAS* : <http://www.netlib.org/blas>
- ~ *LAPACK* : <http://www.netlib.org/lapack>
- ~ *BLACS* : <http://www.netlib.org/blacs>
- ~ *ScaLAPACK* : <http://www.netlib.org/scalapack>
- ~ *ESSL* et *PESSL* :
<http://www.rs6000.ibm.com/software/Apps/essl.html>