

**Une introduction au calcul scientifique**

**Version 1**

**Résumé**

Nous présentons une introduction au calcul scientifique et aux diverses sciences et techniques qu'il utilise et contribue à faire développer ; modélisation scientifique, analyse numérique, logiciels en vue de la simulation, matériels informatiques et réseaux, métiers et implication des hommes, contraintes économiques.

**Plan**

1	Avant-propos	2
2	Modèles	5
3	Logiciels	11
4	Matériels	24
5	Réseaux	29
6	Contraintes économiques	32
7	Conclusion	36
8	Remerciements	37

## 1) Avant-propos

- Définition

Le calcul scientifique est l'art de l'ingénieur qui consiste à effectuer des prévisions numériques fondées sur une démarche scientifique et à l'aide d'ordinateurs.

- Un peu d'histoire

Depuis les années 1970, les ordinateurs sont constitués à base de microprocesseurs, mais l'idée de faire des calculs à moindre effort est aussi vieille que les mathématiques. Nous rappelons simplement ici les motivations qui ont conduit à l'invention de la fonction logarithme. Pour ne pas avoir à faire de multiplications lors des longs calculs de prévision de la position des planètes, John Napier (en 1615) les remplace par des additions via la fonction log (qu'il invente) selon la relation maintenant bien connue :

$$\log (a b) = \log a + \log b.$$

Les calculs d'astronomie et d'astrologie (positions des planètes dans le ciel) ont donc été un moteur puissant du développement des méthodes numériques, dès le 17<sup>e</sup> siècle et on peut dire que lorsque Gauss calculait "manuellement" l'orbite de la petite planète Cérès, il faisait déjà du calcul scientifique, même si la mise en œuvre numérique des méthodes se faisait à l'aide de feuilles de calcul.

Babbage (1850) a eu l'idée du ruban perforé et de la programmation, effectuée par la sœur de Lord Byron, Ada et a ainsi pu mener en langage d'assemblage l'intégration d'équations différentielles ordinaires. On peut citer plus récemment le livre d'Émile Durand [Électrostatique et magnétostatique, Dunod, 1953] où l'auteur propose des méthodes de relaxation pour la résolution numérique de l'équation de Laplace ainsi qu'un mode opératoire et des conseils pratiques (par exemple page 450 et suivantes : "on augmente la vitesse des calculs avec deux personnes travaillant de concert, l'une dictant les chiffres et l'autre opérant avec la machine...")

L'arrivée des premiers ordinateurs pour effectuer des calculs numériques de façon automatique date de la seconde guerre mondiale pour développer la bombe atomique. Voir à ce sujet dans le livre de Richard Feynman [Vous voulez rire, Monsieur Feynman !, Interéditions, 1985] le chapitre "Feynman, la bombe et les

militaires" où il raconte comment, lors d'arrêts intempestifs de la machine, les scientifiques récupéraient une partie des informations déjà calculées en tenant compte des propriétés de vitesse de propagation finie de l'information dans un schéma aux différences finies.

- Représentation des nombres en machine

Avant de poursuivre et définir un certain nombre des termes introduits dans ce paragraphe, il convient de rappeler que dans un ordinateur, l'information est codée sous forme très simple d'un alphabet à deux lettres qui portent le nom conventionnel de "0" et "1". Un bit est une telle information élémentaire qui peut prendre l'une des deux valeurs zéro ou un et qu'on peut interpréter de diverses façons. Comme les bits contiennent chacun très peu d'information, on les regroupe en paquets de huit (on forme de cette façon un octet) et on transfère l'information au sein de la machine (à l'aide du "bus", nous y reviendrons) par paquets de quatre octets (on parle alors de "machine 32 bits") ou de huit octets (machine 64 bits). Il est alors commode (mais ce n'est pas une obligation !) de stocker les nombres réels sur 32 bits (dans le cas d'un ordinateur 32 bits) ou de 64 bits (machine 64 bits). C'est la convention que nous adoptons dans la suite de ce document.

Sans entrer ici dans les détails de codage (la règle proposée par l'association américaine IEEE des ingénieurs électriciens et électroniciens est en train de devenir une norme ; il faut savoir qu'en machine, un nombre réel est codé en suivant la convention dite de mantisse et exposant :

$$a = \pm \{ \text{nombre entre 0 et 1} \} 2^{\pm \{ \text{nombre entier} \}}.$$

La mantisse est le nombre compris entre zéro et un qui porte les chiffres significatifs (de l'ordre de vingt si le nombre est stocké sur 32 bits, de l'ordre de cinquante s'il est stocké sur 64 bits) et l'exposant indique sa taille (très grand ou très petit selon le signe de l'exposant).

On constate donc que les règles de codage des nombres imposent deux limitations fondamentales : il y a d'une part un plus petit nombre (non nul)  $\delta$  qu'il est possible de représenter en machine ( $\delta$  est de l'ordre de  $10^{-45}$  pour une machine 32 bits avec une représentation conforme au standard IEEE,  $\delta$  est de l'ordre de  $10^{-324}$  pour une machine 64 bits au standard IEEE), c'est à dire qu'il n'existe pas en machine de nombre  $x$  tel que  $0 < x < \delta$  et de façon duale, il existe un plus grand nombre  $A$  ( $A$  est de l'ordre de  $10^{38}$  pour une machine 32 bits et

$10^{308}$  pour une machine à la norme IEEE où l'exposant est stocké sur onze bits) que la machine peut représenter. D'autre part, un calcul sur ordinateur s'effectue avec des erreurs d'arrondis : il existe un nombre caractéristique de ces erreurs d'arrondis, c'est à dire le plus grand nombre  $\varepsilon$  tel que, en machine, le résultat de l'addition  $1 + \varepsilon$  est stocké sous la forme "1". L'ordinateur donne donc valeur de vérité "vrai" à la relation suivante :  $1 + \varepsilon = 1$  ! Pour un nombre stocké sur 32 bits,  $\varepsilon$  est de l'ordre de  $10^{-6}$  (ce qui est très grand ; les erreurs d'arrondis sont importantes lorsque les nombres sont stockés sur 32 bits) et  $\varepsilon$  est de l'ordre de  $10^{-15}$  pour une machine 64 bits au standard IEEE. Nous retenons que l'emploi d'un processeur "64 bits" est bien adapté au calcul scientifique.

- Stabilité numérique

Nous avons vu que le stockage des nombres réels sous forme d'un ensemble de 32 ou 64 bits d'informations élémentaires entraîne inévitablement des erreurs de calcul. Il importe donc que les méthodes de calcul employées dans les ordinateurs soient compatibles avec cette caractéristique désagréable des machines. Pour cela, il est indispensable que les erreurs d'arrondis ne s'amplifient pas. Cette caractéristique pratique peut être définie de façon mathématiquement rigoureuse sous le nom de stabilité numérique (voir par exemple le livre de P.G. Ciarlet [Analyse numérique matricielle et optimisation, Masson, 1983]). Seules des méthodes stables numériquement peuvent être utilisées en pratique.

Au 19<sup>ième</sup> siècle, Richardson avait proposé un schéma numérique de résolution pour certains problèmes dynamiques et se proposait d'effectuer des calculs pratiques en utilisant tout un ensemble d'opérateurs humains se transmettant les informations les uns aux autres après chaque étape de calcul élémentaire. Mais sa méthode n'était pas stable et les erreurs (humaines) auraient été inévitablement amplifiées, aboutissant à des résultats numériques sans aucun lien avec une solution approchée du problème initialement posé !

## 2) Modèles

- La solution par les nombres

La plupart des problèmes qu'il est possible de mettre sous forme d'équations conduisent en général à des problèmes trop compliqués pour être résolus par des méthodes "élémentaires", c'est à dire à l'aide de la fonction exponentielle et de l'équation du second degré, ou à l'aide des techniques du calcul formel où la solution du problème posé s'explique grâce à une "formule", c'est à dire des "fonctions spéciales" souvent développables sous la forme d'une série de puissances. On sait par exemple depuis Evariste Galois (1830) qu'une simple équation algébrique posée à l'aide d'un polynôme de degré cinq ne peut pas être résolue par une expression algébrique pouvant contenir des radicaux.

Le calcul scientifique se propose modestement de donner des solutions approchées numériques à des problèmes pour lequel il est possible de se donner une équation qui le modélise. C'est à l'ingénieur utilisateur de cet outil qu'il convient de choisir le modèle à résoudre et faire varier des paramètres de dimensionnement de façon à comprendre intuitivement la valeur des nombres que lui fournit en général en très grande quantité, l'ordinateur.

- Un art de l'ingénieur

Notons ici qu'une pratique semi-empirique fondée sur la notion de simulation de processus physiques complexes et qui ignore la mise en équation continue du problème au bénéfice d'une approche discrète heuristique et d'un algorithme de résolution existe aussi parmi les utilisateurs du calcul scientifique (simulation de l'atmosphère ou aérodynamique hypersonique modélisée par les équations de Navier Stokes parabolisées par exemple). Il est alors difficile de démêler, dans un calcul qui ne donne pas de bons résultats, la part due à la façon dont le problème est posé et celle dont il est résolu. De toute façon et dans tous les cas, la pratique et l'expérience sont irremplaçables et font de cette démarche un véritable art de l'ingénieur.

- Une approche mathématique

Un problème qui doit être résolu numériquement à l'aide d'un ordinateur doit être mathématiquement "bien posé". Cette notion, proposée par Jacques

Hadamard (vers 1920), consiste à dire que le problème considéré admet une solution, que celle-ci est unique et de plus, que cette unique solution dépend continûment des paramètres du modèle. Cette dernière propriété est fondamentale pour le calcul scientifique. En effet, la permanence des erreurs d'arrondis dans les calculs pratiques sur machine montrent que la "solution" calculée par l'ordinateur revient à résoudre exactement un problème dont on aurait perturbé les paramètres d'une grandeur a priori inconnue. Cette perturbation n'est "petite" que si le problème est mathématiquement bien posé.

- Introduction au conditionnement

La résolution numérique d'un système linéaire conduit à la notion théorique de conditionnement d'une matrice. En effet, ayant à résoudre le système linéaire associé à la matrice  $A$ , de second membre  $b$  et d'inconnue  $x$ , c'est à dire l'équation

$$A x = b$$

on est capable de contrôler l'erreur  $\delta x$  commise si on perturbe le second membre de la quantité  $\delta b$  selon la relation

$$\delta x \leq \text{cond}(A) \delta b$$

où  $\text{cond}(A)$  est un nombre qui ne dépend que de la matrice  $A$  et qu'on appelle le "conditionnement" de la matrice  $A$ . Ce nombre peut être très grand même si la taille de la matrice  $A$  est très petite ! Dans ce dernier cas, les erreurs d'arrondis de la machine s'amplifient très vite et les résultats numériques obtenus deviennent donc non significatifs. Il appartient donc au modélisateur d'éviter de faire apparaître des matrices "mal conditionnées" ce qui est en général impossible. Toutefois, une bonne connaissance du conditionnement des systèmes linéaires de grande taille (de quelques centaines à plusieurs millions) qui apparaissent dans les problèmes est indispensable pour s'assurer que les chiffres calculés par l'algorithme de résolution ont un sens.

- Hiérarchie de modèles

Les équations que l'ingénieur peut avoir à résoudre dans l'industrie sont, par ordre de complexité croissante, les quatre opérations, les équations algébriques simples à quelques inconnues, les opérations de traitement du signal (transformation de Fourier rapide ou FFT), le calcul matriciel (une partie importante des statistiques pouvant quasiment rentrer dans le calcul des valeurs propres de matrices symétriques), les équations différentielles ordinaires qui

caractérisent les modèles discrets de la mécanique du point et des solides, la discrétisation des modèles continus, les équations aux dérivées partielles, où la recherche des équations à résoudre par l'ordinateur fait partie du processus de résolution et l'optimum design, c'est à dire la recherche de paramètres permettant de minimiser une fonction qui caractérise le coût technico-économique, sous la contrainte de satisfaire les lois de la physique ou des grands équilibres économiques.

- Résolution numérique d'équations aux dérivées partielles

L'art de l'ingénieur consiste in fine à extraire d'un monde essentiellement complexe quelques données simples et dimensionnantes. Si certains logiciels sont avant tout un concentré du savoir faire accumulé par les experts (code Woodward en aérodynamique par exemple), d'autres ont la prétention d'être des simulateurs les plus fidèles possibles de comportements naturels, sans introduction de savoir-faire a priori.

Il s'agit en général de se donner un modèle physique approprié à la description d'un phénomène naturel donné. Le rôle du physicien est de préciser comment tel ou tel type de phénomène peut être pris en compte de la façon la plus simple possible et d'écrire les équations correspondantes. On tombe alors sur ce que les mathématiciens appellent une équation aux dérivées partielles, dont la particularité est de coupler très fortement l'espace et le temps : équations de Maxwell en électromagnétisme, de Schrödinger en physique moléculaire, de Navier Stokes en mécanique des fluides, etc....

Comme pour les autres problèmes complexes, la résolution analytique d'équations aux dérivées partielles est limitée à quelques exemples académiques et les super calculateurs permettent d'envisager une résolution numérique approchée à des coûts raisonnables. Les objectifs industriels sont de nature globale, comme déterminer une signature radar, une portance ou un échauffement de la paroi. Il est donc indispensable de disposer de méthodes simplifiées efficaces comme l'approximation de la couche limite ou la parabolisation des équations de Navier Stokes en aérodynamique par exemple. La construction d'équations aux dérivées partielles dégradées relève d'une intuition très particulière, où l'on fausse la nature "dans certaines directions seulement" comme dans l'approximation de Darwin des équations de Maxwell, par exemple. L'analyse mathématique de tels modèles est

peu populaire mais leur efficacité pratique mérite un effort important pour faire progresser l'état de l'art.

Les outils industriels pour résoudre les équations aux dérivées partielles sont aujourd'hui de portée générale. En mécanique des structures par exemple, on dispose d'abstractions très puissantes (formulation variationnelle) ayant conduit à des outils numériques de portée générale (éléments finis), ou plus récemment en électromagnétisme, où les mathématiques sous-jacentes sont d'abord linéaires, les différences finies permettent une bonne représentation de la propagation des ondes. Les problèmes de modélisation sont résolus et seule la complexité géométrique et la taille des systèmes considérés (dix millions de mailles au moins sont nécessaires pour représenter un missile entier éclairé par un radar typique) limite in fine la capacité des machines. La discrétisation pose des problèmes difficiles et en particulier les effets de taille finie des grilles : comment écrire que rien ne se passe quand une onde sort du domaine de calcul ? Cette question bien naïve est en fait d'une complexité mathématique certaine et y apporter des solutions concrètes constitue un enjeu essentiel pour le succès des codes de calcul ("conditions limites absorbantes").

Le fait de disposer de méthodes mathématiques générales induit fort logiquement la nécessité de développer des codes de calcul généraux. Il importe de respecter une programmation soignée par l'adoption de structures de données appropriées, une validation rigoureuse sur des exemples triviaux (connaître la solution exacte montre les défauts de l'approximation elle-même) et d'autres exemples plus réalistes physiquement (un code doit effectuer des prédictions, sinon il s'agit d'un jeu intellectuel stérile), tout en prenant en compte les caractéristiques physiques de la machine utilisée (vectorisation hier, parallélisation aujourd'hui). Enfin, un bon module de résolution n'a de sens que couplé à un générateur automatique de maillage (étape clé pour le contrôle des coûts de réalisation puisque l'ingénieur d'étude le conçoit de façon interactive) et à un outil de dépouillement approprié (visualisation tridimensionnelle, extraction de petits fichiers représentatifs d'un effet important, etc...).

Pour terminer sur ce sujet, il faut noter la synergie existant entre équipes de mathématiques appliquées développant des modules généraux de résolution d'équations aux dérivées partielles. Ayant défini un modèle physique, on se rend compte par exemple ensuite que "des phénomènes aussi différents que la banque et le refroidissement d'une coulée d'acier sont mathématiquement décrits par des

modèles analogues" [J.L. Lions, Le Monde, 8 mai 1991], ce qui permet d'envisager un module de résolution unique compatible avec les outils de pré et post traitement existants, de réutiliser des idées de schémas et d'algorithmes ayant déjà fait leur preuve ailleurs, de réutiliser une structure de données d'un module informatique, etc...

- Optimum design et boucle de conception

Un industriel a, entre autres missions, la charge de proposer et réaliser des études de conception de systèmes complexes (lanceurs, missiles, satellites, capsules spatiales par exemple pour Aerospatiale Espace et Défense) au bénéfice de ses clients. De telles études ont fondamentalement un caractère pluridisciplinaire, l'imbrication entre technique et économie étant l'une des contraintes majeures de tels travaux, qui s'étalent typiquement sur plusieurs années. Il importe d'abord de définir avec précision la mission demandée au système (on n'optimise pas de la même façon un lanceur porteur de satellites géostationnaires comme Ariane 5 et une fusée chargée de délivrer des charges sur orbite basse par exemple). Cette étape permet de proposer des grands choix de conception compatibles avec la technologie existante (propulsion, repérage inertiel, etc.). Une fois ces choix acquis, une première ébauche permet de définir ce que seront les contraintes physiques qu'aura à affronter le véhicule : échauffement, dynamique des structures, résistance aux impacts de micrométéorites par exemple.

La maîtrise de l'interaction entre les différentes contraintes (guidage-propulsion, fluide-structure, compatibilité électromagnétique, etc.) est le point fondamental qui permet de réussir à proposer des solutions théoriques admissibles qui conduisent à des réalisations concrètes qui fonctionnent effectivement. La mécanique des fluides par exemple ne fait pas exception à la règle et l'ingénieur dans l'industrie doit, tout en maîtrisant les modèles physiques, mathématiques et informatiques spécifiques, garder toujours à l'esprit la question qui lui posée dans le cadre de l'étude du système couplé. Il s'agit parfois de fournir quelques paramètres numériques "élémentaires" : torseur des efforts de pression sur une structure (pour l'aérodynamique), échauffement maximal d'une paroi (thermique), connaissance d'un pic de pression lors d'une évolution instationnaire rapide, etc.

Une étude globale de système complexe peut donc se formaliser dans le langage de l'optimisation : on cherche à minimiser une fonction de coût sous un

ensemble de contraintes. La fonction coût est en général un résultat économique pour l'entreprise mais peut comporter également des termes de maîtrise de l'environnement (pollution, coût du chômage induit par la réduction de main d'œuvre, etc.). Les contraintes sont des modèles différentiels ou des équations aux dérivées partielles. L'approche traditionnellement adoptée de type "minimisation direction par direction" où chaque discipline optimise ce qu'elle voit de la fonction coût devant à terme laisser la place à une approche plus globale, fondée sur une formalisation et une rationalisation des idées qui précèdent.

### 3) Logiciels

- Définition

Un logiciel est un texte écrit dans un langage particulier destiné à être rendu compréhensible par un ordinateur. Un tel langage a sa grammaire et sa syntaxe qui sont en général indépendantes du matériel utilisé.

- Langages informatiques

Divers langages informatiques dits de haut niveau, sont accessibles à l'utilisateur d'un ordinateur. Pour le calcul scientifique, nous retiendrons ici les plus courants, à savoir Fortran (FORMula TRANslator) proposé initialement par la société IBM en 1955 et le langage C (développé en vue d'écrire le système d'exploitation Unix avec un langage de haut niveau et commercialisé en 1978). Un logiciel écrit en Fortran ou en C est "lisible" par un humain, mais on ne peut pas dire que ce genre de littérature soit très populaire. Aussi on a pris l'habitude de regrouper des morceaux indépendants de logiciels (sous-programmes, ou modules) ayant une fonction bien définie et pouvant communiquer avec d'autres logiciels via un protocole (une règle de transfert de données) bien défini, en bibliothèques de programmes. Une bibliothèque spécialisée pour une famille d'applications physiques particulières s'appelle aussi un "code de calcul".

Un ingénieur devant effectuer une simulation complexe utilise un code de calcul commercialisé si l'application est courante ou spécialisé si cette dernière est rare ou stratégique pour l'entreprise. Ayant évoqué cette "sur-couche" qui donne à un langage informatique un sens macroscopique en termes d'applications, nous devons aussi considérer les "sous-couches", c'est à dire les langages informatiques qui sont utilisés par la machine elle-même pour traiter l'information d'un langage de haut niveau.

Le premier sous-niveau est celui du compilateur. Un compilateur est un logiciel qui transforme un texte de haut niveau (Fortran ou C par exemple) en un texte écrit dans un langage binaire directement interprétable par le matériel.

Un second niveau de langage est celui qui permet de commander, de gérer la machine elle-même. Ce logiciel particulier, le système d'exploitation, était traditionnellement spécifique à chaque type de matériel. Pour le calcul scientifique, le système Unix (né aux Bell Laboratories en 1969 pour les besoins internes d'un petit groupe de chercheurs) tend à devenir l'outil de base pour ce

champ d'applications. Le système Unix se veut le plus indépendant possible des constructeurs. Comme c'est une tâche délicate à réaliser en pratique, une norme internationale, Posix, est décernée aux implémentations d'Unix qui respectent rigoureusement les standards.

Enfin, un dernier niveau de langage (au moins pour le non-informaticien) est celui de la communication entre machines. Avec l'évolution des architectures, les ordinateurs consacrés au calcul scientifique ne disposent plus d'une mais de plusieurs unités de calcul et de stockage de l'information en mémoire. Des langages spécialisés dans la communication de données d'un même programme à plusieurs machines différentes chargées des calculs sont en train de voir le jour depuis 1990. L'un des plus utilisés et des plus diffusés est PVM (Parallel Virtual Machine) de Jack Dongarra de l'université du Tennessee aux USA.

- Quelques grands codes commerciaux

Mécanique quantique

Gaussian 92	Gaussian, Inc
MOPAC	QEPE
DMOL	Biosym
GAMESS	Iowa State University
PERMAS	Intes

Dynamique moléculaire

CHARMm	MSI
DISCOVER	Biostym Technology
Macromodel	Columbia University
Amber	UCSF

Mécanique des structures

SAMCEF	Université de Liège
MARC	Marc Analysis Research Group
ABAQUS	Hibbitt, Karlsson & Sorensen, Inc
ANSYS	Swanson Analysis Systems, Inc
MSC/NASTRAN	Mac Neal Schwendler
ASTER	Electricité de France
PERMAS	Intes

Mécanique des fluides

PHOENICS	CHAM
----------	------

FIDAP	
FIRE	AVL (choisi par PSA pour l'aérodynamique)
KIVA	
FLUENT	Fluent, Inc
STAR-CD	Computational dynamics
AIRPLANE, FLO 67	codes de Jameson

#### Dynamique rapide

PAM-CRASH	Engineering System International
RADIOSS	Mécalog
LS-DYNA	Livermore Software Technology Group
DYNA 3D	Los Alamos Laboratory

#### Compatibilité électromagnétique

JMAG-DYN

#### Traitement d'images

KHOROS	Université d'Albuquerque
GIPS vision	Electronique-Informatique-Applications
AVS	Uniras
PV-Wave	Visual Numerics
Megawave	Universté Paris-Dauphine

#### Optimum design

Le logiciel CADOE de "conception optimale par le calcul paramétré", développé par Mohamed Masmoudi (Toulouse) et Michel Rochette, propose divers algorithmes de contrôle optimal d'équations aux dérivées partielles de la mécanique des milieux continus [l'Usine Nouvelle, décembre 1994].

- Quelques applications non aérospatiales.

#### Mécanique quantique

Le logiciel GAMESS (General Atomic and Molecular Electronic Structure System) résout la structure électronique d'un système moléculaire à partir des équations de la mécanique quantique (équation de Schrödinger ? Approximations de Hartree-Fock ? ) [tourne sur T3D, source mai 1995]. [voir aussi les logiciels

TURBOMOLE, MNDO, GROMOS, VAMP, ADF soutenus par le projet Europort 2, mai 1995].

#### Dynamique moléculaire newtonienne

Le logiciel CHARMM (CHemistry at HARvard Macromolecar Mechanics) permet de calculer l'énergie libre d'une protéine dans une solution, problème clef pour la pharmacologie et les polymères. Le code AMBER (Assisted Model Building and Energy Refinement) est un logiciel très populaire de simulation de la dynamique des protéines et des acides nucléiques, et permet de simuler le mouvement de 10 000 à 15 000 atomes durant une millionième de seconde [Quelle est la valeur du pas de temps ? On résout le problème à N corps avec  $N = 10\,000$  ?]. [tourne sur T3D, source mai 1995].

#### Gaz et pétrole

Simulation de réservoir tridimensionnel (logiciels PROCLIPSE et EP-FRONTSIM), explosion d'un gaz tridimensionnel afin d'en prévoir les effets (logiciel EP-MUSIC), détection de pétrole par interprétation de données sismiques (logiciel EP-SEISMIC) sont soutenus dans le cadre de Europort 2 [mai 1995].

#### Traitement d'images

Images SAR (Synthetic Aperture Radars) : le logiciel CAESAR du consortium PULSAR est soutenu par Europort 2 [mai 1995].

Le logiciel EASI/PACE est implémenté sur Param 9000

- Bibliothèques scientifiques de base

BLAS (Basic Linear Algebra Subprograms)

Algèbre linéaire de base ; produit scalaire de deux vecteurs, multiplication de matrices.

LAPACK

Bibliothèque de base pour la résolution de problèmes s'effectuant sur des machines stockées "pleines", c'est à dire comportant essentiellement des éléments non nuls : systèmes d'équations linéaires, calcul de valeurs propres et de vecteurs propres, décomposition en valeurs singulières, moindres carrés.

NAG (Numerical Algorithm Group)

Créateurs de la célèbre "bibliothèque NAG", la société NAG (Wilkinson House à Oxford) offre maintenant toute une gamme de produits : méthodes numériques, statistiques, calcul formel, visualisation et vient d'être portée aux normes du langage fortran 90 [mai 1995].

IMSL, Matlab.

- Programmation objet

La programmation objet consiste à adopter un point de vue relativement abstrait sur les entités manipulées par un programme informatique, ce qui permet une rapidité globale de développement, une maintenance facilitée d'où une rentabilité économique qui explique le succès de cette approche.

Un objet est une donnée informatique qui n'existe que par les opérations (ou "méthodes") qu'on peut effectuer avec ; les actions qu'on peut faire à un objet doivent être décrites par une méthode. Par exemple, un point ou une matrice sont des objets différents sur lesquels on peut faire des opérations analogues (visualisation, rangement sur fichier) ou spécialisées (addition, multiplication, etc...). De façon plus précise, les objets peuvent être organisés en classes permettant d'aller du cas général au cas particulier en transférant alors de façon globale (par héritage) les méthodes déjà définies pour une classe de niveau supérieur. Par exemple, la classe des matrices carrées est une sous-classe de la classe des matrices et le calcul de l'inverse d'une matrice ou des valeurs propres est une méthode (une action, une opération) qui n'est définie que pour cette dernière classe. Du point de vue de la communication du logiciel, l'utilisateur dispose des attributs (caractéristiques générales, structures de données) et des méthodes (opérations de transformation) d'un objet mais la structure d'un langage objet peut permettre aussi à un utilisateur de concevoir de nouvelles classes dérivées des premières et particulariser l'emploi de l'outil si nécessaire.

En conséquence, une grande autonomie de développement est laissée à l'utilisateur d'un module informatique écrit dans un langage objet, sans pour autant remettre en question par une transmission directe du code source, le détail de la démarche intellectuelle ayant abouti à définir un concept cohérent.

De façon concrète, un programme objet comporte d'abord une spécification de classes en en-tête, c'est à dire la liste des attributs et des méthodes. Ensuite on

décrit chaque fonction, chaque méthode pour transformer cet objet. Les erreurs de programmation sont nécessairement des erreurs de méthodes, ce qui réduit considérablement leur recherche et de ce fait rend l'approche objet rentable. Signalons que l'un des problèmes dans la mise en œuvre du langage est la création et la destruction d'un objet, le danger informatique étant de détruire quelque chose qui n'existe pas ! L'apprentissage d'un langage objet est aujourd'hui considéré comme assez long (six mois typiquement pour C++, langage objet le plus populaire, les chiffres comparatifs étant typiquement de une semaine pour Fortran ou un an pour Ada).

Les langages objets sont apparus dans les années 1960 (Simula, Smalltalk). Le succès de C++ tient du fait qu'il s'agit d'une extension du langage C, ce qui en fait un langage ouvert, fonctionnalité que n'a pas par exemple Smalltalk. Les langages objets n'ont pas a priori été conçus pour traiter des problèmes d'informatique en temps réel, mais il faut signaler que le langage Ada, conçu pour ce type d'applications, comporte depuis peu en son sein un concept d'objet.

L'introduction de la programmation par objet dans le domaine du calcul scientifique doit permettre au cours des dix prochaines années de réduire de façon importante les coûts et méthodes de développement. C'est un outil incontournable de l'informatique vivante des années 90.

- Qualité logicielle

Un logiciel est un texte qui est très sensible aux petites perturbations : la modification d'un seul caractère peut éventuellement en changer tout le sens. Aussi devant la nécessité de construire des outils complexes de simulation qui dépassent largement les capacités d'une seule personne des méthodologies très rigoureuses sont apparues dans l'industrie. Citons également les bibliothèques de programmes en calcul scientifique ainsi que la conception de systèmes d'exploitation des ordinateurs.

La première idée est d'accompagner le logiciel de divers documents. Le besoin de l'utilisateur doit tout d'abord être exprimé de façon très claire (Spécification Technique de Besoin) et la solution théorique apportée à celui-ci synthétisée dans un Document de Justification Technique qui expose les idées guidant la conception du logiciel. Cette conception donne lieu à un Document de Définition qui est un véritable guide de développement pour la personne chargée de

coder le logiciel. Ensuite les validations de différents niveaux qui représentent l'accumulation de savoir-faire sont présentées dans un Document de Validation. Enfin, l'utilisateur doit disposer d'un Manuel Utilisateur pour être autonome lors de l'exploitation.

Tous ces documents de formalisation permettent aux clients utilisateurs d'un logiciel d'avoir l'assurance que tout a été fait pour éviter une erreur de conception ou de réalisation. On doit admettre également qu'il tel travail de soutien reste de conception et de réalisation très lourdes, que la documentation peut peser d'un poids financier non négligeable dans le prix initial de l'outil. Mais le gain immédiat que l'on peut obtenir en ne la réalisant pas peut créer à l'industriel des inconvénients par la suite : mauvaise diffusion de l'outil ou maintenance difficile qui à terme peuvent se traduire par des pertes économiques bien plus importantes que le gain immédiat obtenu.

La seconde idée est de découper le problème en fonctions de plus en plus petites, chaque fonction pouvant être elle même à nouveau divisée en fonctions plus élémentaires. De cette façon on définit des modules logiciels élémentaires que l'on peut valider indépendamment les uns des autres avant de les assembler et valider chaque ensemble de taille de plus en plus grosse et de fonctionnalité de plus en plus complexe. Cette démarche de "cycle en V" (on définit les tâches du général au particulier et on les implémente du particulier au général) est bien adaptée pour définir le logiciel. Il faut également adopter une approche pragmatique et savoir qu'une spécification peut évoluer et de ce fait remettre en cause toute la démarche.

La qualité logicielle est une méthodologie générale rigoureuse destinée à donner un maximum d'assurance qu'un logiciel répond effectivement au besoin spécifié et ne comporte pas d'erreur. Mais elle est tellement chère et rigide que des démarches plus souples et pragmatiques sont utilisées dans la pratique et seront très probablement formalisées dans les années à venir. Des réflexions, confortées par la pratique actuelle, peuvent notablement réduire le coût des logiciels comme la production automatique d'une bonne partie de la documentation à partir des commentaires dans le code source, des outils automatiques de tests. La phase de spécification est de plus en plus couplée à la conception par le développement du maquettage qui permet de prendre en compte la difficulté des points durs et de prévoir au mieux les coûts définitifs. De plus, le développement et la validation tendront à devenir de plus en plus automatisés. L'idée de décomposition d'un

problème en sous problèmes plus simples reste importante, même si le modèle en V n'est plus le seul modèle utilisé, au bénéfice de cascades, de spirales, etc.

La qualité logicielle est une méthodologie utile et rigoureuse qui doit être bien dosée en termes de documentation et de formalisme. Il faut toujours garder à l'esprit que la phase amont du développement est primordiale et qu'il faut avoir une grande visibilité à ce stade du développement car le logiciel reste alors essentiellement abstrait.

- Maillage et lien avec la conception assistée par ordinateur

Le traitement géométrique des données s'effectue traditionnellement en deux étapes, à savoir le maillage et la conception des formes à l'aide des outils de conception assistée par ordinateur (CAO). Le maillage consiste à définir concrètement (dans l'ordinateur !) les éléments de géométrie tels que des volumes discrets ou éléments finis avec leurs caractéristiques topologiques (type de forme, liste des sommets, cas particulier du bord du domaine d'étude) et géométriques (coordonnées cartésiennes des points caractéristiques) en nombre souvent important (de quelques milliers à plusieurs millions). Le logiciel de maillage (ou "mailleur") remplit le volume à partir d'un premier maillage de surface, en avançant petit à petit à partir du bord (méthode frontale) ou par reconstruction de volumes duaux puis primaux de Voronoï-Delaunay à partir d'un nuage de points. Nous renvoyons à l'ouvrage de P.L. George chez Masson pour un développement précis de cette question dans le cas des éléments finis.

Pour l'aérodynamique, on emploie traditionnellement des maillages structurés (grilles "ijk" cartésiennes ayant subi une transformation géométrique régulière), ce qui conduit à des techniques "multiblocs" où l'utilisateur doit d'abord imaginer comment remplir son espace géométrique de travail avec des gros cubes avant un découpage très fin.

Dans tous les cas, la description de la surface de contour est donnée par une description informatique à l'aide de surfaces paramétrées standardisées. L'un des standards d'échange de fichiers est la norme SET (Standard d'Echange et de Transfert) proposée au départ par Aerospatiale et Peugeot et développée par la filiale CIMPA d'Aerospatiale. Toutefois, d'autres protocoles d'échanges (comme IGES par exemple) qui prennent davantage en compte tous les paramètres de l'interface homme-machine (maquillages, couleurs, etc.) continuent à être soutenus par les distributeurs de CAO.

L'idée est de rentrer en machine l'équation d'un morceau de surface paramétrée  $M(s,t)$   $0 \leq s, t \leq 1$  (ou "patch" en anglais) en utilisant des polynômes de Bézier (ingénieur chez Renault, connu grâce à sa thèse soutenue en 1975) ou la famille de fractions rationnelles des Non Uniform Rational B Splines (NURBS). L'utilisateur définit des points de contrôle (ou "pôles") et des poids et le logiciel construit une surface régulière qui passe "au plus près" de ces éléments géométriques. Nous renvoyons au livre de Bézier [Courbes et surfaces, mathématiques et CAO, Hermes, Paris, 1987] pour une introduction à cette question et passons sous silence les liens entre la CAO et le monde de la production (machines-outils automatisées).

Les principaux outils de maillage sont donnés dans la liste qui suit. Signalons que la difficulté de transfert d'information entre l'univers informatique de la CAO (carreaux et surfaces paramétrées dont les équations algébriques sont connues explicitement) et celui du maillage (facettes planes repérées par leurs sommets) est due à la contrainte traditionnelle de devoir faire un maillage qui respecte les contours du carreau de surface paramétrée, ainsi que la présence de trous dans les surfaces de CAO, peu gênantes pour le dessinateur mais bloquantes pour la prise en compte correcte de conditions aux limites lors de la résolution numérique d'équations aux dérivées partielles.

La tendance actuelle (septembre 1995) est à la résolution progressive de ces difficultés : une facette élémentaire de maillage surfacique peut s'appuyer sur plusieurs "patches" de CAO. Ainsi, les logiciels de maillage disposent de plus en plus d'un "CAD modeler" ou modeleur de surfaces CAO et réciproquement les logiciels de CAO offrent la possibilité de disposer en sortie d'un maillage de surface puis de volume.

#### Logiciels de maillage

I-DEAS	SDRC
GIBI (Castem2000)	Commissariat à l'énergie atomique
GHS3D	INRIA-Simulog
MEFISTO	Université Paris 6
PATRAN	Mac Neal Schwendler
PV-Wave	Visual Numerics
ICEM	Control Data Corporation
EMC2	INRIA

## Logiciels de conception assistée par ordinateur (CAO)

Catia	IBM-Dassault Data Systemes
Euclid	Matra

- Graphisme

L'obtention de graphiques et d'images sur l'écran est l'outil essentiel de communication entre l'ingénieur et les données qu'il a à manipuler. L'écran de l'ordinateur ne peut proposer qu'une image, c'est à dire une matrice bidimensionnelle de pixels, points élémentaires de couleur. Or l'information à traiter est en général issue d'un modèle tridimensionnel ; les données physiques de base sont stockées dans une "matrice tridimensionnelle", c'est à dire un tableau de nombres repérés par les trois coordonnées de chaque point. Le passage d'un champ complet à une image visible à l'écran est réalisé par des bibliothèques dites "graphiques 3D", logiciels spécialisés pour ce type de traitement.

Depuis les stations de travail Apollo [1980], l'utilisateur de l'informatique scientifique a l'habitude de manipuler des fenêtres et de les gérer lui-même (taille, position, superposition, couleur, etc...) à l'aide d'une souris. Cette gestion de l'écran graphique est contrôlée par le logiciel X11. Cet outil qui est de plus en plus un standard bien qu'encore en développement, comporte à la fois un serveur (qui exécute sur l'écran les ordres du client comme l'ouverture d'une fenêtre ou un déplacement de la souris), un protocole de communication et une bibliothèque de base, Xlib, écrite en langage C, qui en est en juin 1995 à la version 6, laquelle s'appelle X11R6.

Mais X11 est un outil de base du point de vue des objets graphiques manipulés. Un petit ascenseur comme on peut en trouver à droite d'une fenêtre et qui permet de se déplacer dans le fichier généré lors de la session de travail, est lui-même composé d'une multitude de petites fenêtres. Aussi il existe une bibliothèque graphique standardisée de plus haut niveau, Motif, qui fait appel à la bibliothèque Xlib, et qui permet de concevoir rapidement une interface standardisée entre l'homme et la machine. Il suffit pour la personne en charge de la conception d'une interface de définir les objets dont l'utilisateur aura besoin et de les manipuler globalement à l'aide de la bibliothèque Motif.

Toutefois, l'essentiel du problème technique du graphisme réside dans le long traitement mathématique qu'il est nécessaire d'effectuer pour passer des données tridimensionnelles de l'utilisateur à une image proprement dite. Les composants élémentaires (recherche des données utiles dans la base tridimensionnelle, interpolation) de ces traitements sont effectués à l'aide d'une bibliothèque dite "de base". Il y a une lutte actuellement entre Open-GL d'une part, bibliothèque graphique du constructeur Silicon Graphics (SGI), écrite à l'origine dans le langage d'assemblage du microprocesseur (ce qui en fait un outil efficace mais pas transportable d'une machine à l'autre) et qui a été rendue depuis compatible avec les autres constructeurs, et Pexlib d'autre part, extension de la bibliothèque tridimensionnelle Phigs qui utilise à la base les modules Xlib de X11 et qui a été choisie par la société Hewlett Packard (HP) comme bibliothèque graphique de base.

Signalons que l'idée d'utiliser des couches de logiciel de haut niveau (via toute une série de bibliothèques, il s'agit in fine de langage C) pour ces traitements élémentaires gourmands en puissance de calcul compte tenu de la contrainte de temps réel demandée par l'affichage "instantané" dont doit disposer le pupitreur, a été popularisée (entre autres) par la société française G5G qui a développé une version efficace et opérationnelle de Phigs (logiciel Phigure), exploit technique que n'avait pu réaliser auparavant la société américaine TGS (logiciel Phigaro).

- **Compilateurs**

Un compilateur procède par étapes pour transformer un texte Fortran ou C en langage directement exécutable par la machine. Il procède d'abord à diverses vérifications : analyse lexicale (les mots-clefs, opérateurs, séparateurs, etc... sont bien ceux du langage) puis analyse syntaxique (les règles de grammaire sont bien appliquées). Il y a ensuite production d'un code intermédiaire, éventuellement au cours de l'analyse syntaxique, puis une optimisation de ce dernier en évitant par exemple de refaire ce qui vient d'être fait, comme charger une donnée en vue d'un calcul. Cette phase d'optimisation est cruciale en pratique et les ingénieurs savent bien que l'emploi d'un compilateur optimisé peut permettre de gagner typiquement un facteur trois sur les performances du code exécutable.

Le compilateur C est écrit en C ! En fait, seuls quelques ordres de passage de l'information au processeur sont écrits dans un langage propre à la machine. Une

méthode de programmation est la descente récursive [opérationnelle en 1987, concurrente de la remontée récursive ou de la méthode des tables de prédiction] : on part d'un vocabulaire très réduit qui permet de définir peu à peu de nouveaux mots qu'on peut alors employer librement (un mode d'écriture possible d'un dictionnaire ?).

- **Système d'exploitation**

Quelles sont les différentes couches d'un logiciel système ? Un système d'exploitation demande l'écriture d'un programme comportant de cinq à dix millions de lignes d'instructions. Selon la fiabilité de la programmation, il subsiste typiquement de 200 à 10000 (!) erreurs dans un tel programme [J. Printz, leçon inaugurale de la chaire de génie logiciel du Conservatoire National des Arts et Métiers, mai 1995].

Quelle est l'importance d'un compilateur sur les performances d'un microprocesseur ? i.e. comment passe-t-on de la performance crête à la performance pratique pour une application simple d'algèbre linéaire (résolution de système linéaire, FFT, etc. ) ?

- **Micronoyaux**

Il s'agit de programmer le système d'exploitation de façon modulaire, en mettant en évidence les fonctions essentielles et les opérations plus spécialisées (lire, écrire, effacer par exemple) qui peuvent être confiées à un hardware (un matériel) spécialisé. Il s'agit d'une évolution importante de l'informatique d'aujourd'hui (Cray par exemple a adopté ce type d'architecture logicielle à partir de la version 10 de son système d'exploitation Unicos).

DCE (Distributed Computing Environment) : pour l'avenir. Permet de bien dissocier les fonctions. Les nouvelles fonctions des systèmes d'exploitation qui n'étaient même pas prévues dans les systèmes traditionnels (type Unix) : sécurité, service "temps" de synchronisation globale. Dans les systèmes d'exploitation de demain, l'utilisateur appelle une donnée sans savoir où elle est (disque, cartouche, bande magnétique) ; c'est le système qui gère directement la localité physique des données.

Bagarre sur les systèmes d'exploitation : Windows NT (de Microsoft) système d'exploitation des micro-ordinateurs, contre COSE, consortium IBM, HP, Sun, etc. du marché traditionnel du calcul scientifique et des stations de travail. L'idée est d'avoir toujours les mêmes interfaces, quelles que soient les machines qui ne sont pas des micro-ordinateurs.

Unix devient un véritable label, et la société Novell a été chargée de le délivrer aux seuls systèmes d'exploitation qui respectent la "spec 11 70". Notons que cette spécification est un ensemble de règles communes concernant l'appel système qui sont portées sur tous les noyaux des grands constructeurs du marché ("une révolution"), de façon à offrir les mêmes interfaces système sur des plateformes différentes. De cette façon, l'utilisateur dispose de logiciels vraiment portables d'une machine à l'autre. c'est à dire de spécifications d'appel système portés sur tous les noyaux. Le système d'exploitation Unix est écrit en C, son noyau comporte 500 000 lignes de code, ce qui a fait son succès car C est un langage de haut niveau et non pas un langage d'assemblage propre à chaque constructeur, ce qui permet la portabilité des développements.

#### 4) Matériels

- Microprocesseurs

- (i) processeur scalaire traditionnel

le paramètre physiquement important est le "tick" ou "clock période", c'est à dire la période d'horloge qui est le quantum de temps pour les opérations élémentaires et détermine la vitesse globale du processeur. Pour une station de travail classique comme le sparc 5 [coût de l'ordre de 40 kF] le tick est associé à une fréquence de 50 MHz. Pour un processeur comme celui du Cray YMP (vectoriel et non plus scalaire, attention) on est à 160 MHz et celui du T90 est à 500 MHz (période d'horloge de deux nano-secondes) ; il n'y a qu'un ordre de grandeur entre les processeurs "grand public" et les processeurs des machines du plus haut de la gamme ! Le problème crucial à résoudre pour la conception de tels microprocesseurs est la dissipation du flux de chaleur puisque la course à la vitesse se traduit par une miniaturisation accrue, donc une densité d'énergie très élevée dans les circuits.

- (ii) processeur super scalaire

Il est conçu de façon à pouvoir activer simultanément diverses opérations différentes d'un même processeur comme l'addition et la multiplication par exemple (calcul d'un produit scalaire). On accroît de cette façon le nombre d'opérations réalisées par période d'horloge et ceci permet de multiplier par deux [processeur 88110 de Motorola, 7100 d'HP] ou trois [supersparc de Sun] la performance d'un processeur scalaire.

- (iii) processeur vectoriel

On effectue typiquement [cf. Cray YMP] 64 opérations identiques en même temps, ce qui constitue un parallélisme de données. La préparation des données consiste à remplir des caches pour que les données se répartissent sur les différentes unités de calcul. De plus, le chaînage des opérations élémentaires ("pipe line") permet de gagner sur le temps de traitement des longs vecteurs.

- (iv) processeur RISC

Risc signifie Restricted Instruction Set Computer, ordinateur à jeu d'instruction réduit. Un processeur de ce type n'effectue que les traitements

principaux (80%), les traitements restants étant confiés à d'autres processeurs spécialisés, qui peuvent être montés sur la même carte. En plus d'un coût de fabrication plus réduit, ce type de processeur permet une meilleure optimisation des codes lors de la compilation car on manipule finalement des instructions plus simples que celles utilisées par les processeurs "Complex Instruction Set Computers" (CISC) traditionnels.

- Calcul parallèle

Jusque dans les années 1980, le calcul scientifique de hautes performances était un outil associé aux grandes commandes des états en vue de leur programme électronucléaire, spatial ou de défense. Le développement rapide de l'informatique individuelle au cours des années 1980 (IBM PC et compatibles, stations de travail) a permis de développer une industrie du microprocesseur fondée sur un marché de grande échelle allant même jusqu'au grand public. Le calcul massivement parallèle, techniquement nécessaire pour des raisons de limite technologique des processeurs vectoriels, bénéficie de cette évolution : les processeurs de base sont des produits du commerce, diffusés en grande série sur le marché des stations de travail, donc peu coûteux. Cette discipline se développe à grande vitesse depuis 1990.

- Circuits intégrés.

Technologie VLSI du silicium, processeur RISC à jeu d'instructions réduit, introduction du parallélisme à l'échelle du processeur (un calcul d'adresse, de nombres entiers et de nombres réels peuvent être menés en simultané sur les processeurs RISC actuels).

ECL et CMOS (Complementary Metal Oxyde Semiconductor)

- Le problème de la mémoire

Le microprocesseur effectue les calculs à l'aide de données qui sont stockées en mémoire. Les registres de mémoire directement accessibles pour un calcul lors de la période d'horloge considérée doivent être rangés dans le "cache". Celui-ci est

toujours de taille limitée et les données supplémentaires sont rangées en "mémoire vive" (RAM : Random Access Memory). L'accès entre le microprocesseur et la mémoire vive est assuré par le "bus". A chaque période d'horloge, le bus peut transférer 32 bits (64 pour une machine 64 bits) sur les broches correspondantes du microprocesseur. Toutefois, le bus voyage à vitesse finie et plusieurs périodes d'horloge (attente) sont nécessaires pour assurer le transfert complet d'un jeu de données. Par ailleurs des instructions élémentaires (fonctions d'assemblage typiquement) indispensables au bon fonctionnement de la machine et inaccessibles aux utilisateurs sont stockés en "mémoire morte" (ROM : Read Only Memory).

### Quelques machines

- Le "Top 500" [J. Dongarra, juin 1995]

C'est un classement publié tous les six mois des 500 premiers sites au monde pour le calcul scientifique sur des critères très généraux de puissance totale installée ce qui ne donne pas une indication précise de leur utilisation réelle ni des plus grands problèmes résolus. Si l'on prend en compte le nombre de sites installés, les trois premiers matériels sont les suivants :

Power challenge de Silicon Graphics	128 sites
SP2 de IBM	66 sites
C90 de Cray	46 sites.

En termes de puissance totale mesurée en multiples du "flop" (un flop = une opération par seconde avec des nombres réels codés en machine, un mégaflop = un million d'opérations réelles par seconde, un gigaflop =  $10^9$ , un téraflop =  $10^{12}$ , un pétaflop =  $10^{15}$  opérations réelles par seconde), on a le classement suivant :

Cray	1130 gigaflops
Fujitsu	630 gigaflops.

Notons que 130 systèmes utilisent la technologie ECL et 370 la technologie CMOS et que le nombre de machines parallèles (ou MPP : Massively Parallel Processors) atteint 230.

- Serveurs Power Challenge de Silicon Graphics (SGI) [avril 1995]

A base de microprocesseur 64 bits de type RISC MIPS R8000 à 75 MHz ou 90 MHz de période d'horloge, de 1 à 18 processeurs selon les configurations, la puissance crête va de 300 mégaflops à 6,5 Gigaflops, la mémoire de 64 mégaoctets à 16 gigaoctets, la masse de 16 à 180 kgs. Prix d'entrée de la gamme = 360 kF.

- T90 de Cray [HPCN, mai 1995]

Ordinateur vectoriel "classique", successeur du YMP et du C90. De 4 à 32 processeurs (de la société Cray), mémoire centrale de 512 mégaoctets à 8200 mégaoctets selon les configurations, refroidi à l'air ou grâce à un liquide (secret de fabrication ?), bande passante de 8 à 35 gigaoctets par seconde. Caractéristiques du microprocesseur ? 500 MHz ? Prix d'entrée ?

- SP2 de IBM [avril 1995]

Celui du centre d'Ames de la NASA dispose de 160 nœuds de type RS/6000-590 avec une mémoire par nœud allant de 128 à 512 mégaoctets.

- Serveur VPP 300 de Fujitsu [HPCN, mai 1995]

Un lien avec le VPP 500, où chaque processeur élémentaire (technologie CMOS, Large Intégration ?) de performance crête de 1,6 gigaflops et une mémoire de 2 gigaoctets. Refroidissement par air. Distribué en Europe par Siemens.

- SX-4 de NEC [HPCN, mai 1995]

Il existe une série dite "un nœud" où sont regroupés de 1 à 32 processeurs de 2 gigaflops chacun et offrant une puissance crête allant jusqu'à 64 gigaflops et une série dite "multi-nœuds" allant de 16 à 512 processeurs regroupés en nœuds de huit processeurs, soit une puissance maximale théorique de un téraflop (!). La mémoire va de 0,25 à 2 gigaoctets par processeur et le flux de données de 128 gigaoctets par seconde et par nœud. Les plaquettes n'indiquent aucun prix catalogue.

- Exemplar SPP 1200 de Convex [GFUCX, mai 1995]

Le processeur utilisé est le PA-RISC 7200 contenant 1,3 millions de transistors, de la société Hewlett Packard (période d'horloge de 120 MHz, 240 mégaflops de performance crête). Ils sont regroupés par paquets de 8 (hypernœuds) et un maximum de 16 hypernœuds permet une puissance crête théorique de 30 gigaflops. C'est une machine multiprocesseur où la mémoire peut être adressée globalement. On peut aussi la programmer par échange de messages (PVM). Prix = ?

- N-cube3 de N-cube [J. Sauvage, juin 1995]

Processeur 64 bits développé par la société N-cube (laquelle a été créée par des transfuges d'Intel) de période de base 50 MHz et 100 mégaflops de puissance crête. Mémoire physique allant de 1 à 16 gigaoctets par processeur. Système d'exploitation Unix programmé par micronoyaux qui demande (seulement ?) 500 kilo-octets de mémoire sur chaque processeur. Une entrée de gamme à 16 processeurs (2 MF environ) allant jusqu'à une machine téraflop (10 000 nœuds, 200 MF). Noter l'orientation prise vers la gestion (N-cube appartient aujourd'hui au fondateur du système de bases de données relationnelles Oracle).

- Serveur alpha 2100 de DEC [HPCN, mai 1995]

Processeur alpha de 233 à 275 MHz à 64 bits de gigaoctets de mémoire. Jusqu'à 4 microprocesseurs sur une même station.

- Param 9000 de C-DAC [mai 1995]

Le consortium Center for Development for Advanced Computing est l'outil d'indépendance nationale de l'Inde en matière de calcul massivement parallèle. Processeur du Param : supersparc 2, existe en 4 à 200 nœuds, performance de pointe : 240 mégaflops (un processeur) et 12 gigaflops (200 processeurs)

## 5) Réseaux

- Généralités sur les réseaux

Un réseau informatique est une structure logico-matérielle qui permet de faire communiquer entre eux des ordinateurs a priori différents. Un message doit donc se composer d'un début clairement identifiable par toutes les machines interconnectées, un corps de message qui a un sens pour l'utilisateur et son correspondant et une marque de fin de message. Cet ensemble de connexion constitue un protocole de communication. Divers protocoles existent comme TCP/IP, IPI3 ou SCSI-2...

Afin d'unifier une démarche globalement complexe, l'organisme international de standardisation (ISO, International Standard Organism) propose de retenir un modèle de réseau à sept couches, ce qui donne un formalisme abstrait pour concevoir la notion de réseau, même si la pratique peut différer quelque peu (voir aussi le livre de R. Stevens [Unix Network Programming, Prentice Hall, 1990]). La couche de base (numéro un) est le moyen physique de communication : fil de cuivre, fibre optique, faisceau hertzien, et la plus haute (numéro sept) celle que constitue l'utilisateur humain du réseau. Entre ces deux couches extrêmes, on peut distinguer cinq couches logicielles. En allant de l'homme à la physique, on trouve successivement la couche de présentation (numéro 6) qui permet d'aider l'utilisateur dans ses envois de message puis la session proprement dite (couche 5), l'ensemble des couches 5 à 7 du réseau étant géographiquement localisées au même endroit que la station de travail. La couche suivante (numéro 4) est caractéristique du transport proprement dit (TCP/IP par exemple) et la dernière couche logicielle (numéro 3) permet de gérer les communications entre réseaux, ce qui permet de définir la notion d'ensemble de réseaux d'interconnexion (inter-net en anglais, le réseau universitaire Internet n'étant que l'un d'entre eux !). Enfin, la couche 2 du data-link contient le logiciel directement nécessaire pour piloter la liaison physique.

Dans la pratique, les réseaux sont internes pour la conception des machines, et externes pour faire communiquer les machines entre elles : Ethernet, fibre channel, SONET, ATM (Asynchronous Transfer Mode, standard pour les réseaux téléphoniques, le débit peut aller jusqu'à 2,5 gigabits par seconde), FDDI (Fiber Distributed Data Interface), HIPPI (High Performance Parallel Interface : standard ANSI de réseau de hautes performances, issu des réflexions de Los

Alamos pour relier entre elles deux machines voisines par une liaison à haut débit, plus de 2500 sites interconnectés [mai 1995], de 0,8 à 1,6 gigabits par seconde, c'est à dire trois fois plus performant que fibre channel et six à dix fois plus performant que FDDI).

La société Psi'Tech (Fountain Valley, Californie 92708) propose l'interface HFB 360-24 qui est un "frame buffer" (tampon d'image) de hautes performances pour la visualisation d'images en couleur de plus de 4 millions (image 2000 par 2000) de pixels. La connexion par HIPPI permet des échanges de plus de 200 mégaoctets par seconde ce qui permet une animation, même pour des images de haute résolution

- Réseaux d'interconnexion de stations de travail

Il importe de séparer le besoin local de l'utilisateur (puissance calcul et graphisme éventuellement 3D) qui peut avoir à changer de poste de travail (100 kF typiquement) selon le taux d'utilisation des moyens et le lieu de stockage de ses fichiers qui doivent être sur une structure plus centralisée (un serveur à 300 kF par bâtiment typiquement). De plus, le serveur de fichier doit être placé dans un endroit "digne de recevoir de l'informatique" pour éviter le coup de balai malencontreux de la femme de ménage qui débranche une prise et paralyse le réseau. Ainsi, on s'attachera à avoir une alimentation stabilisée et un précablage des installations électriques (prises réseau RG 45, électrique et téléphone à un mètre du sol).

Protocole de communication TCP/IP, IPI-3, SCSI-2

TCP/IP (TCP : Transmission Control Protocole, IP : Internet Protocol]. TCP est un mode de communication où l'on vérifie que les données arrivent bien à destination, dans le même ordre que lors de l'envoi et sans duplication. IP est le protocole de communication de bas niveau du monde Unix. Il est donc propre au logiciel et indépendant du matériel utilisé, ce qui en fait un standard pour le monde des scientifiques. Les machines ont toutes, au niveau mondial, une adresse écrite sur quatre octets (et huit à partir de 1996) et NIC (Network Information Center de la Federal Communication Authority ou ministère des PTT des Etats Unis) est l'organisme qui a la charge de distribuer les adresses IP et a sous-traité le travail à SRI International. Les deux premiers octets sont relatifs à l'adresse du réseau ou du sous-réseau et les deux derniers au numéro de la machine dans le

réseau. Une série d'applicatifs de communications sont en général intégrés à TCP/IP et permettent d'unifier ces modules de plus haut niveau : FTP (File Transfer Protocol, ou transfert de fichiers sans exécution de tâche à distance), telnet (connexion à distance), rlogin (connexion sur une autre machine du réseau que celle dont l'utilisateur dispose), rsh (exécution de commandes sur une autre machine du réseau), rcopy. Le dialogue entre le monde Unix du calcul scientifique et les autres mondes informatiques (IBM, PC, ?) qui utilisent d'autres protocoles de communication s'effectue à l'aide de la couche "TCP" qui permet de passer du protocole propre (SNA par exemple pour les réseaux IBM) au standard IP.

- Internet

Courrier électronique, forum de discussions, pages de présentation des organismes de type World Wide Web (WWW), Mosaic, logiciels Eudora et Netscape de connexion sur micro-ordinateurs, etc. Tout ceci le lecteur le sait déjà ; inutile d'en parler ici !

- Hommes

Métiers : ingénieur physicien, numéricien, mathématicien, informaticien soft, informaticien réseau, technicien spécialisé dans un matériel donné (Cray, Convex,...), spécialiste graphisme, spécialiste maillage, spécialiste CAO. Formation initiale et continue, rémunération, évolution de carrière.

## 6) Contraintes économiques

- Introduction

Pour avoir un résultat de calcul non trivial, quelles doivent être les compétences d'un grand maître d'œuvre ? Que doit-il savoir faire par lui-même ? Où doit-il mettre son argent ? Avec qui doit-il collaborer ? Qui doit-il soutenir ? Ordre de grandeur du marché du calcul scientifique dans le monde. Quelle croissance prévisible ?

- Santé économique des entreprises qui fabriquent des ordinateurs

Cray, IBM, Intel, Fujitsu, NEC... Quelles caractéristiques communes, quelles différences entre ces grandes entreprises qui fabriquent et vendent des ordinateurs ? Les pionniers du calcul massivement parallèle, qui se sont placés sur cet unique marché, se sont cassés la figure (Thinking Machine Corporation, Kendall Square Research). La société de Seymour Cray (Cray inc), fondée en 1990, qui développait un ordinateur fondé sur l'emploi de microprocesseurs à l'arséniure de gallium [AsGa], plus rapide que les processeurs traditionnels au silicium) a déposé son bilan en 1995. Plus récemment, Archipel et Acri ont fait faillite et Cray Research (Cray YMP, T3D, T3E, etc...) a été racheté par Silicon Graphics [février 1996].

- Concepteurs et revendeurs de logiciels scientifiques

ESI, Cisi, Mécalog, Cham, sont des petites et moyennes entreprises qui conçoivent et/ou commercialisent des logiciels de calcul scientifiques pour des applications concernant en général plusieurs secteurs économiques ayant à résoudre des problèmes de même type (mécanique des structures, dynamique rapide (crash), mécanique de fluides, etc...) . Rôle difficile des "vedettes" qui ont développé l'essentiel d'un logiciel puis entrent en conflit avec leur manager, démissionnent et emmènent de ce fait l'essentiel du savoir-faire de l'entreprise (voir à ce sujet le conflit ESI-Mécalog) par exemple.

- Comment faire du calcul scientifique à bas prix

Le marché de la microinformatique et en particulier du Personal Computer ("PC") permet de proposer des prix bas à l'aide de microprocesseurs conçus pour un PC et fabriqués en grande série qui utilisent le système d'exploitation MS/DOS (Micro-Soft Disc Operating System) avec ses sur-couches logicielles Windows (noter toutefois qu'avec Windows 3.11, un compatible "PC" peut être moins efficace d'un facteur 3 environ !). Avec une telle configuration, on dispose d'un matériel de bureautique standard dont le prix de revient est typiquement de 10 à 15 kF.

Le microprocesseur Pentium de la société Intel [32 bits, période d'horloge de 90 MHz] est a priori un très bon outil pour le calcul scientifique et l'on peut estimer qu'il est aussi puissant qu'un HP 9000 série 700. En revanche, il est vendu en standard sous le système d'exploitation MS/DOS-Windows. Il suffit donc de changer le système d'exploitation (qui est, rappelons-le, le logiciel qui pilote l'ordinateur) et de le remplacer par (un clone) d'Unix. Or, grâce au réseau universitaire (Internet), des besoins pédagogiques ont conduit le Professeur Tannenbaum (Amsterdam) à développer Minix, qui est une excellente base d'Unix. Devant la qualité du travail effectué, Linus Tordvalds, étudiant de 3<sup>o</sup> cycle à l'université d'Helsinki, a développé, en collaboration avec d'autres universitaires grâce au réseau Internet (Rémi Carde, de l'Institut Blaise Pascal à l'Université Paris 6, a développé le système de gestion des fichiers) le logiciel Linux, qui est une version d'Unix développée pour le monde du PC, est un utilitaire dont le source est disponible gratuitement sous les auspices de la Free Software Foundation (projet GNU [GNU is Not Unix ; comprend qui peut !]) et est de plus compatible avec la norme internationale Posix. Le document de Shahid Bokhari [The Linux Operating System : an introduction, rapport ICASE 95-49, juin 1995] constitue une excellente introduction à Linux.

Le doublet (Linux-Pentium) permet de se constituer en kit un outil pour le calcul scientifique pour la même somme de 10 à 15 kF environ. Acheter boîtier, mémoire vive RAM, (270 F le mégaoctet), disque dur (540 mégaoctets de disque dur = 1000 F), écran (de 1,5 kF pour un écran "14 pouces" à 5 kF pour un écran "16 pouces"), périphériques (lecteur de disquettes, vidéo disque, etc....) ou le tout assemblé pour 10 kF. Il suffit de recharger la machine avec le système Linux au lieu de MS/DOS. Signalons enfin que pour des raisons de stratégie commerciale,

Intel, qui a développé le i-860 [superscalaire, 64 bits, son successeur en 1996, le P6, co-développé avec HP, sera avec 250 MHz trois fois plus rapide (seulement !) que le Pentium] pour le calcul scientifique et parallèle, ne développera probablement pas une telle gamme de produits.

- Aspects industriels

D'un point de vue industriel, un calcul doit être mené de façon à être compatible avec le cycle de production des études, le temps de l'ingénieur chargé de mener le calcul étant essentiellement consacré à la préparation et la vérification des données du calcul puis une fois le calcul effectué au dépouillement et à l'interprétation des résultats. Dans un contexte de saturation du matériel (qui arrive on le sait peu de temps après la mise en service d'une nouvelle installation !) il faut pouvoir mener les deux opérations ont lieu au cours de la même journée, le calcul doit être mené typiquement pendant l'heure du repas, donc doit durer au maximum une heure compte tenu de la charge des moyens informatiques. Si les deux opérations de préparation des données et de calcul ont lieu au cours de deux journées consécutives, le calcul proprement dit est mené de nuit, donc peut durer typiquement cinq heures si l'on a affaire à une application prioritaire gourmande de ressources. Enfin, pour des calculs de taille exceptionnelle, une partie du week-end peut y être consacrée, la machine travaillant alors en automate pour mener (sans imprévus du système d'exploitation !) un calcul qui peut alors durer jusqu'à vingt à trente heures. Cette contrainte étant posée, il convient de déterminer ce qui peut être mené au cours d'un calcul : application simple (actuellement en journée), reprise d'une simulation complexe demandant des ressources relativement importantes (actuellement de nuit typiquement), optimisation de paramètres (à prévoir pour le futur).

- Nouveaux développements industriels
  - 1) Utilisation du calcul parallèle pour le pré et post traitement (voir par exemple le quatrième Programme Cadre de l'Union Européenne).
  - 2) Calcul scientifique avec consigne de précision (évaluation de l'erreur et maillage adaptatif).
  - 3) Calcul scientifique et contrôle de process. Résolution numérique de modèles simplifiés "en temps réel" pour la commande d'installations industrielles.

## 7) Conclusion

- Grands enjeux et problèmes scientifiquement ouverts

D'un point de vue de la science fondamentale. Par exemple en mécanique des fluides : turbulence et modélisation du climat ; on veut traiter l'échelle macroscopique (de l'ordre du mètre pour la turbulence "industrielle", 10 000 kilomètres pour la météorologie et le climat) à l'aide de modèles où il est nécessaire de faire intervenir des échelles où l'entropie se dissipe (quelques millimètres pour la turbulence industrielle, quelques kilomètres pour les fluides géophysiques). On a donc un paramètre d'échelle grand ( $10^4$  au moins typiquement), et un facteur de complexité important (loi de Kolmogorov du nombre de Reynolds à la puissance  $9/4$  pour la turbulence par exemple) qui entraîne une taille de problèmes (plusieurs milliards d'inconnues incompatible avec les machines actuelles).

- Limitations du calcul scientifique

De deux types : choix des modèles, taille des problèmes.

Choix des modèles. Pauvreté de la science actuelle : on ne dispose par exemple pas d'une équation simple pour décrire macroscopiquement un tourbillon qui satisfait dans le microscopique aux équations de Navier Stokes. On ne dispose pas non plus de "bon" schéma numérique pour la mécanique des fluides parce que les non linéarités des équations de Navier Stokes n'ont pas encore été analysées mathématiquement avec suffisamment de détail pour construire naturellement (comme en mécanique des structures avec la méthode des éléments finis par exemple) un modèle discret.

Taille des problèmes : on ne peut donner aux machines qu'un modèle compatible avec leurs performances physiques (vitesse de calcul, taille mémoire, complexité de programmation).

## **8) Remerciements**

L'auteur remercie Patrick Bizotto, André Cariou, Jean-Pierre Dumont, Dominique Farbos, Jean Pierre Margeot, Sopha Nor, Jean-Bernard Renard, Antonio Rivas, Marc Waknine pour leur aide indispensable lors de l'élaboration de cette synthèse.