

Chapitre 3

Sophemis : superviseur d'optimisation

Sophemis : superviseur d'optimisation

1. Présentation de l'application

1.1. Intérêt de l'outil

Toutes les procédures de calcul présentées dans les chapitres précédents, plans d'expériences et optimisation par PE, reposent sur la réalisation d'*expériences virtuelles*, c'est-à-dire de simulations ou plus généralement de calculs mathématiques effectués par ordinateur.

La grandeur centrale reste toujours la *fonction objectif*. Celle-ci est généralement une représentation mathématique d'une caractéristique d'un dispositif ou d'un processus réel à améliorer ou à concevoir. Les calculs se concentrent donc sur les résultats obtenus à partir de modèles numériques de ces dispositifs ou de ces processus. On peut dire qu'il s'agit d'expérimentation sur des *prototypes virtuels*.

Les valeurs numériques de cette fonction objectif peuvent être obtenues par plusieurs moyens ; les principaux sont :

- les expressions mathématiques analytiques explicites ;
- les simulations éléments finis (ou tout autre méthode du même type) ;
- les calculs utilisant les réseaux de perméances ;
- les simulations temporelles (Simulink, etc.) ;
- les modèles mathématiques obtenus à partir de plans d'expériences.

Lors d'une optimisation ou du calcul d'un plan d'expériences, les expériences à réaliser sont différentes les unes des autres. Chaque expérience résulte de la modification des valeurs d'un ou plusieurs facteurs.

Cela implique par exemple, de modifier les dimensions du modèle éléments finis, de changer les valeurs des intensités des courants dans les conducteurs, etc.

Le modèle nécessite ainsi d'être remaillé, les calculs refaits, et enfin les valeurs de sorties (les réponses) déduites à nouveau des valeurs brutes issues des simulations.

La réalisation de toutes ces tâches s'avère vite fastidieuse et très coûteuse en temps. Elle peut également être source d'erreurs.

Le logiciel Sophemis a donc été conçu pour faciliter au maximum la réalisation de ces opérations.

Il assure l'automatisation complète des calculs des plans d'expériences et des processus d'optimisation.

Présentation de l'application

Plus généralement, Sophemis a été conçu de façon à servir de structure d'accueil aux développements portant sur l'optimisation des dispositifs électrotechniques.

La signification du nom « Sophemis » devient plus claire ; il s'agit en effet d'un acronyme signifiant :

Superviseur
Op'timisation de
Machines
Electriques
Modélisées

Le développement de Sophemis est indispensable à l'application de la méthode des plans d'expériences (chapitre 1) et des techniques d'optimisation par PE (chapitre 2). Tous les résultats présentés dans le quatrième chapitre n'auraient pu être obtenus aussi efficacement sans le recours à cet outil.

Cette application a donc atteint une certaine ampleur [Vivier 01].

L'application informatique est structurée thématiquement, selon ses sous-répertoires, tel que le montre le tableau ci-dessous.

Tableau 1 - Répartition des fichiers dans les sous-répertoires de Sophemis

Ss – répertoires de Sophemis		Nb fichiers	Fonction
Code	Sophemis	126	Cœur de l'application Sophemis (noyau)
	MatOptTools	85	Application annexe assurant la gestion des modules (§5.2.2.)
	Simul	136	Fonctions mathématiques de base
Utilisateur	Exemples	(226)	Fichiers d'exemples - Modèles éléments finis
	Fcts	38	Fichiers décrivant des fonctions analytiques caractéristiques
	Matrices	9	Définitions de matrices d'expériences particulières
	Modules	571	Définitions de tous les modules existants
	Plans	8	Définitions de plans d'expériences particuliers
		973 fichiers (sans les exemples)	

La partie Code, invariante, constitue le cœur de l'application. La partie Utilisateur regroupe les fichiers développés, modifiés et utilisés par l'expérimentateur.

1.2. Présentation fonctionnelle

L'application Sophemis possède des fonctionnalités pouvant être regroupées en 3 parties.

➤ Evaluation des réponses (*pré-processeur* - §2.)

Sophemis permet de réaliser chaque expérience virtuelle de manière totalement automatique.

Les fonctions réponses peuvent être issues de fonctions analytiques ou de simulations éléments finis.

➤ Réalisation des calculs (*solveur* - §3.)

De la même manière, le déroulement des calculs (plans d'expériences, optimisations) est également totalement automatique. Il suffit de fixer les paramètres caractéristiques et de lancer les calculs : l'intervention de l'utilisateur n'est alors plus nécessaire.

➤ Exploitation des résultats (*post-processeur* - §4.)

Une fois les calculs achevés, l'exploitation des résultats peut être menée. On considérera ici :

- l'analyse des données de simulation ;
- la représentation graphique des données de simulation.



Sophemis peut également être caractérisé par d'autres points importants, tels que :

- le lancement de séries de simulations sur plusieurs ordinateurs distants (calcul distribué avec configuration maître-esclave) (paragraphe 5.1.) ;
- la modularité complète de l'application (toutes les fonctionnalités de Sophemis sont vues comme des briques élémentaires ajoutées à un noyau fixe) (paragraphe 5.2.) ;
- l'existence d'une aide HTML renseignant sur le fonctionnement général, les fonctions mathématiques employées et sur la programmation interne permettant son fonctionnement (paragraphe 5.3.).

1.3. Support

Sophemis fonctionne sous Matlab® [Matlab 99]. Il est donc totalement indépendant du système d'exploitation utilisé (Windows®, UNIX, etc).

Sous Windows, il requiert une version de Matlab supérieure ou égale à 5. Sophemis s'adapte à la version de Matlab qu'il détecte : certaines fonctionnalités sont activées ou inhibées suivant la valeur de cette information.

Le choix d'utiliser Matlab comme support à Sophemis repose sur les principales raisons suivantes :

➤ Très grande diffusion du logiciel Matlab

L'application mathématique Matlab est très largement employée par les établissements d'enseignement, et est de plus en plus répandue dans les entreprises concernées par des travaux de recherche notamment. Matlab est devenu une application incontournable ; il constitue un standard parmi les programmes scientifiques.

➤ Simplicité du langage de programmation propre à Matlab et gestion du code

Les diverses fonctions de l'application constituent chacune un fichier script (fichier texte non compilé). En fonctionnement habituel, le code de l'application est interprété. Les mises à jour éventuelles des fonctionnalités sont alors simplifiées. De plus, grâce à cette particularité, il est possible de faire intervenir des applications satellites venant modifier le code source de Sophemis, afin par exemple d'en augmenter les fonctionnalités. C'est de cette manière que la modularité de l'application est gérée (paragraphe 5.2.).

➤ Disponibilité immédiate de fonctions optimisées de calcul

Les fonctions mathématiques de Matlab sont directement accessibles et exploitables. La méthode des plans d'expériences étant essentiellement basée sur des relations matricielles, l'utilisation de Matlab est alors quasi-optimale. Ce logiciel repose en effet sur l'idée que toute donnée est une variable matricielle ; les fonctions mathématiques permettant de traiter ces données ont donc été créées et optimisées dans ce sens.

On notera également l'existence de *toolboxes* (boîtes à outils), c'est-à-dire d'ensembles de fonctions Matlab supplémentaires regroupées thématiquement (Optimization, Statistics, ...). Ces ressources additionnelles sont parfois utilisées par Sophemis, lorsqu'elles sont disponibles.

➤ Disponibilité immédiate de fonctions de représentation graphique

Parmi les fonctions offertes par Matlab, il faut citer les fonctions de représentation graphique qui sont largement utilisées par Sophemis. Etant donné la grande place de la modélisation et de l'interpolation en général dans la MPE, il est appréciable d'utiliser des fonctions éprouvées à ces fins. Cela évite ainsi de récrire des fonctions existantes qui parfois, nécessitent des connaissances issues de domaines différents de ceux des études menées.

➤ Gestion automatique de la mémoire

Un atout majeur de l'utilisation de Matlab est la transparence totale quant à la gestion de la mémoire. Le nombre important de variables utilisées ainsi que les grandes dimensions de certaines matrices peuvent rendre cette tâche difficile dans un environnement de programmation « conventionnel ».

1.4. Principe de fonctionnement

On peut représenter comme suit la démarche typique d'optimisation, réalisée lors de toute utilisation de Sophemis.

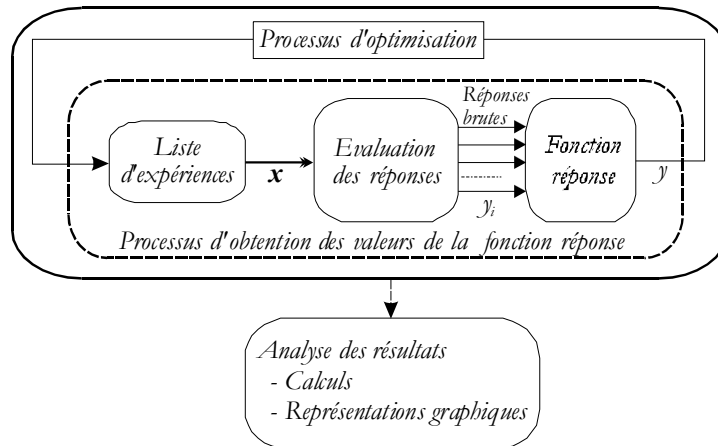


Figure 1 - Processus calculatoire général suivi par Sophemis

L'utilisateur choisit le type de démarche qu'il veut réaliser (plan d'expériences ou optimisation).

Quelle que soit cette démarche, il est nécessaire de connaître la valeur des réponses (y_i) pour des points d'expériences précis (x).

Pour les processus d'optimisation, le recours à la notion de *fonction réponse* est indispensable, afin d'identifier de façon unique la grandeur à optimiser. Cette fonction peut être une combinaison des réponses définies par l'utilisateur. Afin de faire aisément la distinction, ces dernières seront appelées *réponses brutes* (ou *réponses de base*), puisque résultant directement de l'achèvement des simulations.

La valeur de la fonction réponse ainsi calculée est réutilisée par l'algorithme, afin de réinitialiser un nouveau cycle.

L'utilisateur doit préciser le moyen par lequel les réponses brutes sont calculées (prototype virtuel, fonction analytique, etc.).

Les calculs terminés, on accède alors à l'étape du *post-processing*, c'est-à-dire de l'analyse des résultats. Les opérations réalisables à ce stade dépendent de l'étape précédente.

1.5. Interface – aspect visuel

L'application Sophemis se présente comme la réunion de 2 fenêtres.

La *fenêtre principale* possède une barre de menu et une autre de boutons. Sa fonction est de donner accès aux différentes commandes de l'application. Cette fenêtre ne sert pas à l'affichage des données, il s'agit uniquement d'une console.

La seconde fenêtre (ou *fenêtre secondaire*) permet d'afficher les informations données par l'expérimentateur pour la réalisation de ses calculs.

1.5.1. Fenêtre principale

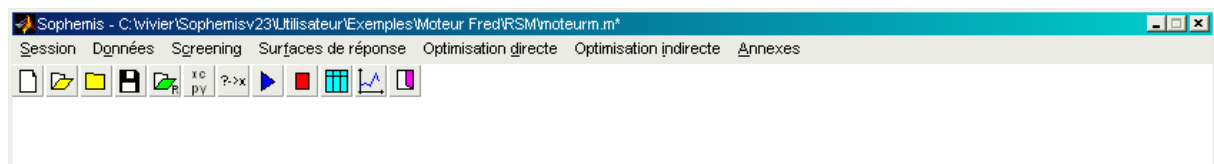


Figure 2 - Vue de la fenêtre principale de Sophemis (exemple)

La barre de menu est l'accès privilégié aux commandes de l'application.

Le contenu de ces menus peut être modifié par l'ajout ou la suppression de modules (paragraphe 5.2. « Modularité »).

Présentation de l'application

Le tableau suivant donne le contenu de tous les menus et sous-menus. Les items indiqués en italique sont des fonctionnalités apportées par des modules. Certains modules sont cités dans plusieurs sous-menus, car ils peuvent être utilisés dans différentes configurations.

Des items de menu peuvent devenir inaccessibles (ils apparaissent alors en grisé). Cela tient au fait que les fonctionnalités sous-jacentes ne peuvent être appliquées dans ces situations. Ainsi, par exemple, il sera impossible de calculer et d'afficher les effets des facteurs principaux si le dernier calcul réalisé a été celui d'un plan de RSM.

Tableau 2 - Liste des items des menus et sous-menus de Sophemis

Session	Nouvelle		
	Ouvrir		
	Enregistrer		
	Enregistrer sous ...		
	Fermer		
	Charger un fichier de résultats		
	<i>Calculs multi-ordinateurs</i>		
	Outils	Répertoire par défaut	
		Fenêtre des paramètres	
		Editer un fichier de session	
Editer le fichier de session courant			
Editer un fichier de résultats			
Editer le fichier de résultats courant			
	Sauvegarder les résultats en mémoire		
	Quitter		
Données	Origine des données		
	Variables		
	Contraintes		
	Modèles		
	Fonction réponse		
	Sélectionner plan		
Screening	Plans fractionnaires		
	Calculs	<i>Résidus</i>	
		<i>Effets</i>	
		<i>ANOVA</i>	
	Représentations graphiques	<i>Effets principaux et Interactions (Hist.)</i>	
		<i>Représentation de Daniel</i>	
		<i>Effets</i>	
	Surfaces de réponse	Sélectionner plan	
		Etudier l'optimalité	
<i>Créer une matrice D-optimale</i>			
<i>Créer une matrice D-optimale approximative</i>			
Calculs		<i>Résidus / Observations</i>	
		<i>Analyse statistiques</i>	
		<i>Meilleurs modèles</i>	
		<i>Calculer la réponse</i>	
		<i>Calculer la fonction de variance</i>	
		<i>ANOVA sur modèles</i>	
		<i>Analyse canonique</i>	
Représentations graphiques		<i>Réponse calculée – réponse mesurée</i>	
		<i>3 facteurs – o=2</i>	
		<i>2 facteurs</i>	
		<i>3 facteurs par interpolations</i>	
		<i>2 facteurs par interpolations</i>	
	<i>Représenter la fonction de variance</i>		
Optimisation directe	Sélectionner méthode		
	Convertir en surfaces de réponse		
	Calculs	<i>Calculer la réponse (interpolation)</i>	
	Représentations graphiques	<i>Toutes les réponses</i>	
		<i>3 facteurs par interpolations</i>	
Optimisation indirecte	Sélectionner méthode		
	Fonction d'optimisation		
	<i>Représenter la fonction d'optimisation</i>		
	Calculs	<i>Calculer la réponse (interpolation)</i>	
	Représentations graphiques	<i>Toutes les réponses</i>	
		<i>3 facteurs par interpolations</i>	
<i>2 facteurs par interpolations</i>			
Annexes	Options ...		
	MatOptTools		
	Aide générale HTML		
	A Propos ...		

La majorité des items de menus servent à activer des modules. Ces derniers sont détaillés plus loin au paragraphe 5.2.2. de ce chapitre.

Présentation de l'application

➤ Menu Session

Une *session* regroupe l'ensemble des travaux et des résultats obtenus par l'utilisateur au cours de l'étude d'un dispositif.

Toute session est matérialisée par un *fichier de session* mémorisant les paramètres indiqués par l'utilisateur pour la configuration de ses procédures de calcul.

Par exemple, pour l'évaluation d'un plan d'expériences, les informations suivantes sont enregistrées dans le fichier de session :

- définitions des variables (facteurs, réponses, etc.) de l'étude ;
- type de plan d'expériences utilisé ;
- source des réponses utilisée (fonction analytique, simulations OPERA, etc.) ;
- définition de la fonction réponse ;
- indication du fichier des résultats ;
- etc.

Le menu **Session** donne accès aux outils gérant de tels fichiers (création, ouverture, fermeture, enregistrement).

Il permet également de charger les fichiers de résultats.

Dans le cas où le module de gestion des calculs distribués sur ordinateurs distants est installé, le menu « Session » permet alors d'accéder à la commande **Calcul multi-ordinateurs**.

Divers autres outils y sont également accessibles.

➤ Menu Données

Ce menu donne accès aux commandes permettant de :

- déterminer l'origine des réponses ;
- définir les caractéristiques des variables du problème (facteurs, réponses, etc.) ;
- entrer les relations définissant les contraintes actives ;
- créer des modèles polynomiaux particuliers ;
- définir la fonction réponse à partir des réponses brutes.

C'est un des plus importants menus de l'application.

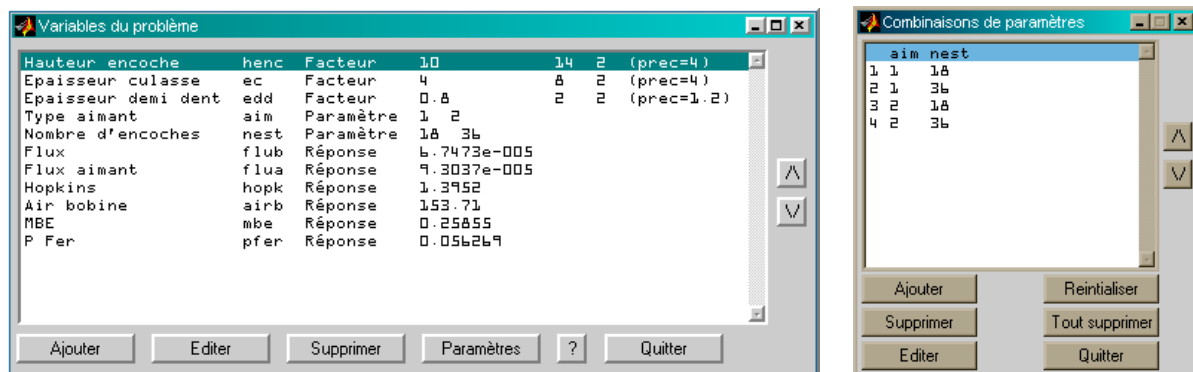


Figure 3 - Boîte de dialogue pour la gestion des variables (gauche)
Boîte de dialogue pour la détermination des combinaisons de paramètres (droite)

- L'illustration de gauche de la figure précédente représente la boîte de dialogue principale pour la gestion des variables définies pour la session en cours. Dans cet exemple particulier, ont été définis 3 facteurs, 2 paramètres et 6 réponses.
- Pour chaque facteur, l'utilisateur doit spécifier les bornes inférieure et supérieure et le nombre de valeurs intermédiaires. La valeur correspondante de la précision est automatiquement calculée et affichée en fin de ligne.
- Pour chaque paramètre, les modalités doivent être indiquées (18 et 36 pour le « Nombre d'encoches » par exemple).
- Pour chaque réponse, l'utilisateur peut spécifier une valeur (dite *de référence*) qui sera utilisée pour diviser toutes les valeurs correspondantes issues des simulations : cela permet d'opérer des normalisations de valeurs.
- Lorsque au moins 2 paramètres sont définis, il existe obligatoirement des combinaisons entre leurs valeurs respectives. Il est possible d'éditer ces combinaisons (tri, suppression, édition ou même création). Ces tâches sont

Présentation de l'application

assurées par la boîte de dialogue représentée dans la figure 51, à droite. Cette fenêtre liste les combinaisons des valeurs des paramètres, qui seront prises en compte dans les calculs à venir.

➤ Menu Screening

Dans ce menu sont regroupées toutes les commandes relatives à l'utilisation des plans de screening. Dans un premier temps, il y a la possibilité de choisir le plan à calculer.

Les autres fonctionnalités dépendent des modules installés (paragraphe 5.2.2.). Elles concernent presque toutes l'exploitation des résultats (post-processing).

➤ Menu Surfaces de réponse

Le principe est identique au précédent.

Dans un premier temps, ce menu donne la possibilité de choisir le plan d'expériences de RSM parmi une liste.

➤ Menu Optimisation directe

L'utilisateur doit avant tout sélectionner la méthode d'optimisation qu'il va utiliser.

Le choix réalisé, le bouton « Configurer » permet de fixer les paramètres propres à la méthode choisie.

➤ Menu Optimisation indirecte

Le déroulement général des commandes pour ce menu est identique au précédent.

Les différences résident dans les possibilités de :

- définir la fonction d'optimisation ;
- représenter graphiquement la fonction d'optimisation.

Le paragraphe 3.4. décrit le principe de l'optimisation indirecte et sa gestion sous Sophemis.

➤ Menu Annexes

Dans ce menu sont incluses les commandes permettant de configurer de manière générale le fonctionnement de Sophemis (éditeur texte utilisé, répertoire d'installation de l'aide HTML de Sophemis, etc.).

Ce menu donne également accès à l'aide HTML de l'application Sophemis.













Il est enfin possible de lancer l'application satellite MatOptTools, assurant la gestion des modules au sein de Sophemis.



La barre de boutons permet d'accéder rapidement à certaines commandes particulières du menu. Son apparence visuelle ne change jamais ; cependant, suivant les configurations rencontrées certains boutons sont rendus inopérants (au même titre que les commandes correspondantes du menu).

Le tableau suivant donne la représentation de chaque bouton, ainsi que des commandes correspondantes.

Tableau 3 - Liste des composantes de la barre de boutons dans la fenêtre principale de Sophemis

Bouton	Commande correspondante du menu	Bouton	Commande correspondante du menu	Bouton	Commande correspondante du menu
	Nouvelle (Session)		Ouvrir (Session)		Fermer (Session)
	Enregistrer (Session)		Charger un fichier de résultats		Variation
	Origine des données		Lancer les calculs (*)		Arrêter les calculs (*)
	Afficher les résultats (*) (**)		Représentation graphique (**)		Quitter

Présentation de l'application

(*) : Ces commandes ne sont pas accessibles par le menu.

(**) : Ces boutons activent des commandes différentes suivant le type de calcul réalisé (plan d'expériences, optimisation, ...).

1.5.2. Fenêtre secondaire

Cette fenêtre affiche de manière condensée les principaux paramètres de calcul fixés par l'utilisateur pour la session courante.

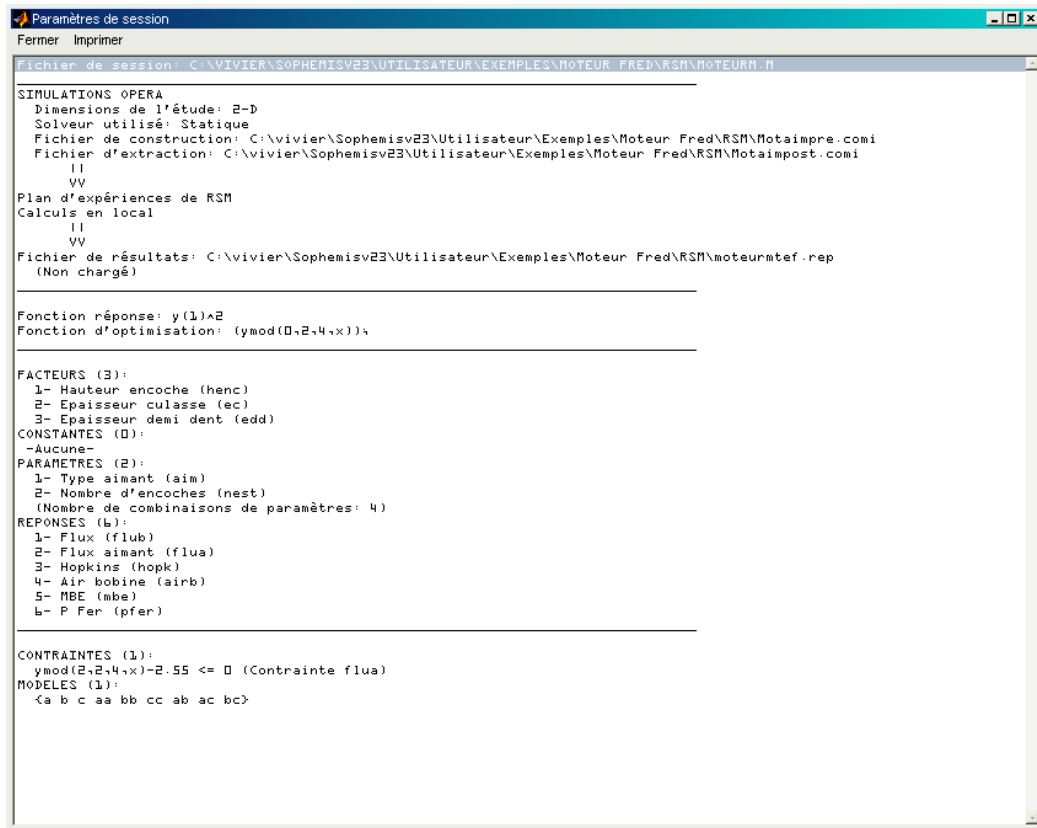


Figure 4 - Vue de la fenêtre des paramètres de session (exemple)

Les principales données affichées dans cette fenêtre sont aussi celles recopiées dans le fichier de session. Cela comprend entre autres :

- le type d'origine des réponses et informations connexes ;
- le type de calcul choisi (plan d'expériences, processus d'optimisation, ...) ;
- le mode de calcul (local ou par machines distantes) ;
- le nom du fichier de résultats dans lesquels les données expérimentales seront enregistrées ;
- l'expression de la fonction réponse, ou nom du fichier externe contenant sa définition ;
- l'expression de la fonction d'optimisation, ou nom du fichier externe contenant sa définition ;
- la liste des variables regroupées par type ;
- la liste des expressions des contraintes avec leur nom ;
- la liste des expressions des modèles polynomiaux supplémentaires définis par l'utilisateur.

2. Évaluation des réponses

Par « évaluation des réponses », on désigne les procédures permettant de calculer les valeurs des réponses, à partir des définitions des points d'expériences.

Toutes les expériences étant virtuelles, savoir calculer une réponse revient donc à maîtriser la façon dont est modélisé le système étudié.

Sophemis gère actuellement 3 moyens d'évaluation :

- les fonctions analytiques ;
- les modèles éléments finis ;
- les modèles (polynomiaux ou non) de comportement déjà connu.

Ces différentes possibilités sont vues plus en détail dans la suite.

2.1. Fonction analytique

Il s'agit de la solution la plus simple.

Elle permet de définir des fonctions mathématiques ayant des caractéristiques complètement connues. Cette option est aussi utilisée pour le test des algorithmes d'optimisation relativement à des fonctions analytiques au comportement piégeur mais déterminé.

De telles fonctions analytiques sont donc de la forme :

$$y_i = f_i(x_1, x_2, \dots, x_k)$$

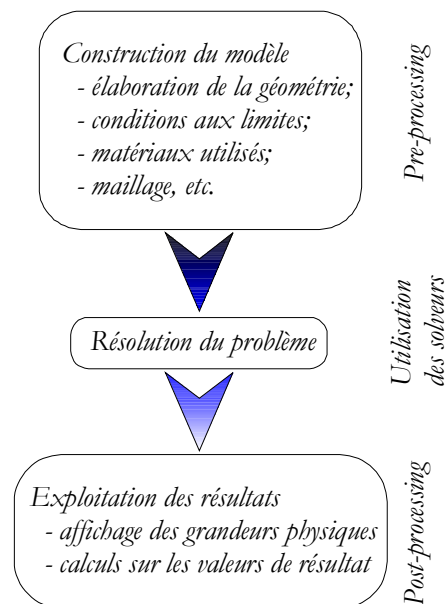
La variable y_i est un scalaire désignant la valeur de la $i^{\text{ème}}$ réponse définie par l'utilisateur. f_i est donc la fonction analytique associée à cette réponse en particulier.

En principe, les arguments d'entrée sont des valeurs de facteurs ; cependant Sophemis donne la possibilité de faire intervenir également les paramètres et les constantes éventuellement définis.

C'est ainsi qu'a été traitée dans le deuxième chapitre, la fonction analytique de test servant à l'évaluation des performances des algorithmes d'optimisation.

L'utilisateur peut ainsi définir une bibliothèque de fonctions analytiques.

La définition complète de l'origine des réponses est donnée par l'indication du nom du fichier dans lequel est écrite la fonction analytique en langage Matlab.



2.2. Simulations par éléments finis

Déduire la valeur des réponses brutes de simulations éléments finis est en quelque sorte la raison de l'existence de Sophemis. Cela en constitue l'option fondamentale et essentielle.

La réalisation des calculs éléments finis est assurée par le logiciel OPERA® [Opera 01].

2.2.1. Principes généraux des simulations OPERA

La figure ci-contre indique le principe général de fonctionnement d'OPERA.

Il compte 3 étapes majeures.

❶ La première correspond au pre-processing, c'est-à-dire aux opérations situées avant les tâches de calcul proprement dites.

C'est lors de cette phase que le modèle éléments finis est construit. Il y a création de la géométrie de l'objet virtuel. Les

Evaluation des réponses

matériaux utilisés sont caractérisés (perméabilité, courbe B-H, etc.). Les conditions aux limites sont également indiquées à ce stade. Après quoi, il y a création du maillage.

② Le problème étant pleinement défini par les opérations du pre-processing, il peut alors être calculé (des relations mathématiques sont déduites des manipulations précédentes ; elles sont ici résolues). C'est la fonction des solveurs.

Seul est utilisé le solveur correspondant au type de problème physique traité. Pour son choix, il faut ainsi considérer (entre autres) :

➤ le nombre de dimensions du problème

Il s'agit du nombre de dimensions de l'espace de modélisation du prototype virtuel : 2 ou 3 (2D ou 3D respectivement).

➤ si le modèle prend en compte des parties mobiles

La prise en compte des parties en mouvement nécessite l'emploi de techniques de résolution différentes. C'est par exemple le cas avec les machines électriques tournantes pour lesquelles la rotation du rotor doit être considérée.

➤ le type de mouvement des parties mobiles (s'il y en a)

On fait ici la distinction entre les déplacements linéaires et les mouvements de rotation des parties mobiles.

Les solveurs adéquats sont fournis par OPERA.

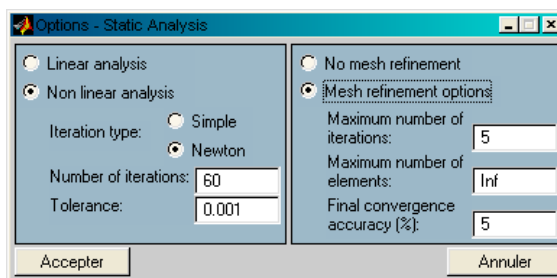


Figure 5 - Boîte de dialogue pour le paramétrage du solveur statique (2D) de OPERA

- La figure ci-dessus donne la représentation de la boîte de dialogue permettant de fixer les paramètres de fonctionnement du solveur statique 2D de OPERA.
- Lorsque les simulations sont réalisées directement sous OPERA (i.e. sans passer par Sophemis), ces paramètres doivent être donnés en utilisant les boîtes de dialogue de OPERA.
- Sous Sophemis, les valeurs spécifiées par l'utilisateur dans ces boîtes de dialogue, sont traduites dans le langage de programmation de OPERA, afin qu'elles soient prises en compte lors de l'exécution du solveur.

③ La troisième étape est celle du post-processing, pour laquelle il y a exploitation des résultats renvoyés par le solveur. En effet, celui-ci ne calcule que des grandeurs physiques de base (champ électrique \vec{E} , induction B , potentiel vecteur \vec{A} , etc.). Il est donc nécessaire d'opérer un traitement supplémentaire afin de connaître les valeurs des entités qui nous intéressent, c'est-à-dire des réponses, avant d'être renvoyées.

Accessoirement, dans cette même étape, on peut demander à représenter graphiquement la répartition spatiale de certaines grandeurs physiques.

De manière simplifiée, on peut considérer qu'il existe un programme distinct pour chacune des 3 étapes précédentes.



Habituellement, toutes les opérations décrites précédemment sont effectuées à partir de l'interface utilisateur graphique de OPERA, manuellement, par l'expérimentateur lui-même.

Cependant, la réalisation de ces mêmes tâches est également possible par l'utilisation d'un langage de programmation propre à OPERA.

Evaluation des réponses

En réalité, les opérations de l'interface utilisateur sont traduites par des commandes du langage OPERA, et ensuite exécutées.

Cela conduit naturellement à la possibilité de créer des fichiers scripts particuliers (fichiers texte interprétés, d'extension *.comi [*command in*]) servant de fait à l'automatisation des tâches.

On peut notamment retrouver une utilisation équivalente des fichiers paramétrés dans le cadre d'optimisations de dispositifs électrotechniques dans [Vong 01].

2.2.2. Utilisation de OPERA par Sophemis

Sophemis exploite alors l'énorme intérêt de ce langage de commande, pour diriger OPERA à sa guise. Il respecte en outre la séquence à 3 volets présentée au paragraphe précédent.

Ainsi, pour chacune de ces étapes, Sophemis lance le programme (exécutable) correspondant, tout en contrôlant les données qui y seront utilisées.

Pour le pre-processing, l'utilisateur doit fournir un fichier COMI contenant toutes les instructions appropriées (construction de la géométrie, maillage, ...). Ce fichier doit être paramétré, c'est-à-dire prendre en compte les différentes valeurs des facteurs calculées et imposées par Sophemis.

L'utilisateur doit également fixer les paramètres qui seront utilisés par le solveur préalablement choisi.

Enfin, il doit donner les commandes (dans un second fichier COMI externe) permettant d'extraire les données intéressantes.

Sophemis lance ainsi le pre-processeur et fait exécuter les commandes du premier fichier COMI de construction de modèle. Les fichiers décrivant la modélisation éléments finis sont alors créés par OPERA.

Ils sont ensuite utilisés par le solveur (lancé en second lieu par Sophemis) pour le calcul des grandeurs physiques fondamentales sur l'ensemble du maillage.

De là, les valeurs renvoyées sont utilisées pour l'étape suivante, lors de laquelle le post-processeur est lancé. Les grandeurs intéressantes sont calculées et écrites dans un fichier externe temporaire, afin d'être récupérées puis chargées par Sophemis, à la fin du processus.

On notera enfin, que la communication entre Sophemis et le logiciel de calcul électromagnétique n'aurait pu exister si OPERA ne possédait pas de langage de commande, ou du moins d'une fonctionnalité autorisant un contrôle externe.

Il est évident que tout programme externe, contrôlable par un langage de commandes, est également susceptible d'être utilisé par Sophemis. Il en est ainsi par exemple des codes de calculs thermique, mécanique ou plus simplement d'Excel®.

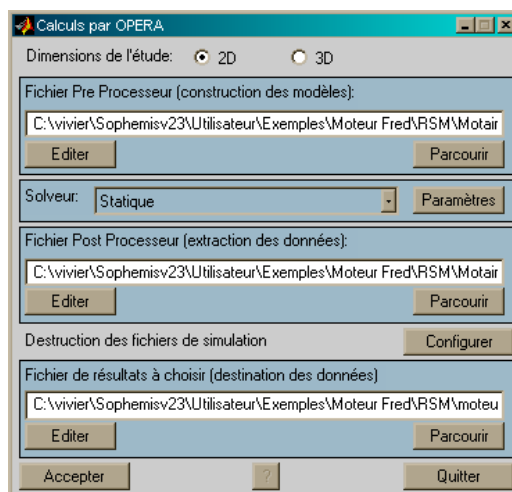


Figure 6 - Boîte de dialogue pour le paramétrage des calculs externes (par le logiciel OPERA)

Evaluation des réponses

- La figure précédente permet de retrouver les éléments nécessaires pour le paramétrage des calculs externes, c'est-à-dire utilisant des logiciels autres que Sophemis. Cela concerne en particulier les simulations réalisées grâce à OPERA.
- Dans un premier temps, il faut sélectionner le nombre de dimensions du problème à traiter (2D ou 3D).
- Ensuite, le fichier *.comi de construction et maillage du modèle éléments finis doit être sélectionné. Cette étape correspond au pre-processing d'OPERA.
- Après cela, le solveur doit être choisi dans la liste déroulante. Le contenu de cette liste est dépendant des dimensions de l'étude (2D ou 3D) et bien entendu des modules installés.
- Enfin, dans le cadre du post-processing d'OPERA, il faut donner à Sophemis le nom du fichier de commande *.comi calculant et extrayant les valeurs des réponses des données issues du solveur.
- Etant donné que chaque simulation réalisée par OPERA donne le jour à de nombreux fichiers, l'utilisateur peut vouloir ne pas les garder. Il peut alors configurer la destruction automatique de ces fichiers de simulation.
- On notera enfin qu'il est nécessaire de spécifier le nom d'un fichier de résultats, dans lequel les valeurs numériques des réponses brutes et de la fonction réponse seront enregistrées.



La réalisation d'expériences par des programmes externes tels que OPERA induit des difficultés supplémentaires notables. Celles-ci se concentrent en particulier autour de la construction des modèles éléments finis servant au calcul des réponses.

Les facteurs considérées dans de telles études sont très souvent purement géométriques. C'est ainsi que le changement simultané de leurs valeurs, d'une expérience à une autre, implique inévitablement des modifications de géométrie du modèle. De ce fait, il faut veiller aux points suivants :

- le modèle peut devenir non valide ;
Du fait de la modification de plusieurs côtes, certaines parties du modèles peuvent venir se chevaucher ou se déformer : le modèle éléments finis ne décrit alors plus une configuration physiquement envisageable ou techniquement possible. Cela définit de fait des contraintes en position qu'il faut identifier dans une étape préalable.
- le maillage dépend du modèle ;
Le maillage du modèle est construit à partir de ses dimensions. Si celles-ci changent, le maillage est amené à évoluer également. Cet aspect est important car la pertinence des résultats donnée par la méthode des éléments finis est essentiellement due à la qualité du maillage utilisé.

Les remarques précédentes font apparaître la difficulté de concevoir des fichiers COMI de construction de modèles sûrs et fiables, quelles que soient les valeurs des facteurs.

C'est ainsi qu'il pourra parfois être nécessaire de :

- concevoir des modèles comme la juxtaposition de (sous-)modèles utilisables uniquement pour certaines configurations de valeurs des facteurs ;
- paramétrer le maillage ;
Principalement, la densité de nœuds de la tessellation du modèle peut être maintenue quasiment constante en faisant dépendre le maillage des dimensions propres du modèle. Il peut également être fait usage de la fonction de raffinement automatique du maillage proposée par OPERA, afin par exemple, d'obtenir une erreur de maillage inférieure à un seuil donné.

Dans tous les cas, c'est à l'utilisateur de s'assurer de la validité des fichiers de construction de modèles qu'il donne à Sophemis. Il doit pour cela les tester et les éprouver dans les configurations susceptibles d'être problématiques.

2.3. Modèles connus

Le troisième type de source de données possible – modèles polynomiaux résultant de plans d'expériences – est fourni par des fichiers de résultats Sophemis existants.

De tels fichiers ne sont en fait que des enregistrements de séries de points. Ils décrivent les variations de la fonction réponse (ou d'autres réponses) sur une partie ou la totalité du DEP. Ils sont alors utilisés comme support à une approximation de la réponse considérée.

Cette approximation est utilisée comme « fonction analytique » ; elle est à même de donner les valeurs de la réponse en n'importe quel point du DEP.

Evaluation des réponses

L'utilisation de telles sources de données s'accompagne toujours d'une phase de reconstruction de la réponse. Afin d'aboutir à des résultats les plus précis possibles, l'approximation choisie doit donc être de bonne qualité.

Si le fichier de résultats initial décrit les données issues d'un plan d'expériences de RSM, l'outil employé peut être le modèle polynomial déduit de ce plan, dans l'hypothèse d'une qualité satisfaisante.

Dans les autres cas, l'interpolation issue de la tessellation du domaine est d'un usage simple et courant.



Le recours à ce principe est le fondement même de l'*optimisation indirecte* (paragraphe 3.4.). Celle-ci vise en effet à trouver les conditions optimales par l'utilisation d'une approximation de la réponse étudiée.

3. Réalisation des calculs

3.1. Plans d'expériences de screening

Sophemis offre la possibilité de calculer et de paramétrer de nombreux plans de screening.

Cela comprend principalement :

- les plans factoriels complets à 2 niveaux par facteur ;
- les plans factoriels fractionnaires à 2 niveaux par facteur ;
- les plans de Plackett-Burman ;
- les plans de Taguchi à 2 niveaux par facteur.

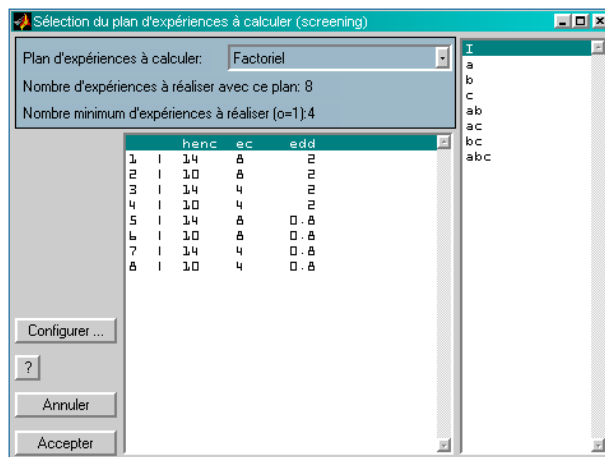


Figure 7 - Boîte de dialogue pour la sélection des plans de screening

- La liste des plans d'expériences de screening disponibles est donnée par la liste déroulante située en haut de la boîte de dialogue.
- Pour chaque plan sélectionné, on peut lire la liste des coordonnées des expériences ainsi que la définition du modèle calculé (liste de droite). Afin de simplifier les écritures, la référence au $i^{\text{ème}}$ facteur a été remplacée par la $i^{\text{ème}}$ lettre de l'alphabet (a pour le 1^{er} facteur, b pour le 2^{ème}, etc.). Ainsi, pour la figure ci-dessus, le modèle calculé par le plan factoriel s'écrit plus naturellement comme :

$$y = b_0 + b_1.x_1 + b_2.x_2 + b_3.x_3 + b_{12}.x_1.x_2 + b_{13}.x_1.x_3 + b_{23}.x_2.x_3 + b_{123}.x_1.x_2.x_3.$$
- Certains plans nécessitent la spécification de paramètres pour être pleinement définis. Le bouton « Configurer » permet alors de faire apparaître les boîtes de dialogues correspondantes, nécessaires à cette tâche.

3.2. Plans d'expériences de RSM

De la même façon, Sophemis propose un choix de plans d'expériences de RSM pouvant être calculés automatiquement.

Cela concerne :

- les plans de Doehlert ;
- les plans de Box-Behnken ;
- les plans centraux-composites ;
- les plans grilles ;
- les plans treillis ;
- les plans de Koshal ;
- les plans de Hoke ;
- les plans simplexes ;
- les plans hybrides ;
- les plans équiradiaux.

Les plans grilles et treillis sont des plans originaux, à la base des méthodes exhaustives présentées au chapitre 2.

Réalisation des calculs

Le placement des points peut également être imposé par l'utilisateur. Celui-ci peut ainsi créer ses propres plans d'expériences de RSM, ou bien modifier des séries de points existantes.

L'annexe 1 donne une description de tous ces plans et le premier paragraphe de la partie Tables en donne les principales propriétés.

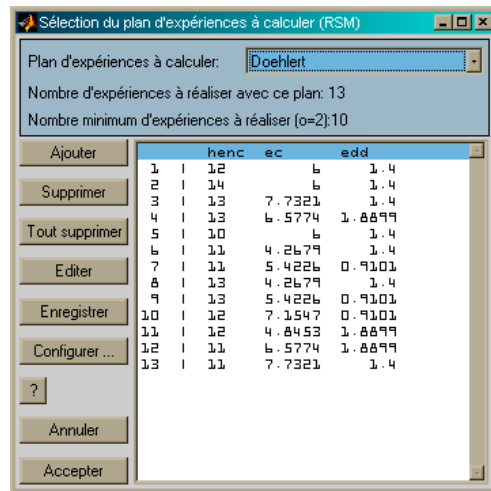


Figure 8 - Boîte de dialogue pour la sélection des plans de RSM

- Le choix du plan d'expériences de RSM se fait par la liste déroulante située en haut de la boîte de dialogue représentée ci-dessus.
- Les coordonnées des points d'expériences sont celles du plan choisi ; elles prennent en compte les bornes des facteurs. Lorsque des contraintes sont définies, Sophemis s'assure que toutes les expériences sont calculables ; dans le cas contraire, l'utilisateur est averti.
- Pour tout plan, il est possible d'éditer les coordonnées des points d'expériences ainsi calculés. Des expériences peuvent être ajoutées ou supprimées.
- Certains plans d'expériences demandent à être paramétrés ; dans ces cas, le bouton « Configurer ... » donne accès à une autre boîte de dialogue correspondant au plan choisi et permettant de fixer les données manquantes.

3.3. Méthodes d'optimisation directe

Sous Sophemis, les méthodes d'optimisation sont classées en 3 groupes :

- les méthodes d'optimisation déterministes « classiques » ;
- les méthodes d'optimisation par plans d'expériences ;
- les autres méthodes.

3.3.1. Méthodes déterministes classiques

Par « méthodes classiques », on désigne ici certains algorithmes d'optimisation déterministes.

On distinguera de façon naturelle les méthodes unidimensionnelles (Tableau 4) des méthodes multidimensionnelles (utilisant les premières dans leur fonctionnement - Tableau 5) [Minoux 83].

Tableau 4 - Méthodes classiques unidimensionnelles implémentées dans Sophemis

	Nature du problème	
	Sans contrainte	Avec contraintes
Suite de Fibonacci	✓	✓
Nombre d'Or	✓	
Interpolation Quadratique	✓	

Tableau 5 - Méthodes classiques multidimensionnelles implémentées dans Sophemis

	Nature du problème	
	Sans contrainte	Avec contraintes
Gradient Conjugué	✓	✓
Plus Grade Pente	✓	✓
BFGS	✓	

Réalisation des calculs

Les méthodes multidimensionnelles citées ci-dessus appartiennent à la classe des *méthodes indirectes*, ou *méthodes d'ordre 1*, car elles nécessitent le calcul du gradient de la fonction réponse en chaque point considéré.

Le respect des contraintes se fait ici par l'application de la méthode des Pénalités Extérieures.

3.3.2. Méthodes d'optimisation par plans d'expériences

Les méthodes d'optimisation par plans d'expériences ont été largement discutées lors du second chapitre.

Toutes les techniques qui y sont décrites sont gérées par Sophemis, ce qui permet :

- de fixer les paramètres de configuration de ces algorithmes ;
- de lancer et de gérer leur déroulement ;
- d'extraire les données finales obtenues.

3.3.3. Autres méthodes d'optimisation

Les « autres méthodes » sont celles qui ne fonctionnent pas selon l'un des 2 principes présentés ci-dessus.

Principalement, il s'agit des algorithmes stochastiques, c'est-à-dire notamment :

- des algorithmes génétiques [Goldberg 94];
- de l'algorithme du recuit simulé [Pardalos 02].

Il y a ici réalisation de séries de plusieurs expériences lors de chaque itération, mais aucun modèle n'est calculé. Seules sont prises en compte les valeurs de la fonction réponse en ces points. De plus, le placement relatif des points n'est pas imposé par la connaissance d'aucun plan d'expériences ; il se fait principalement par considération de facteurs aléatoires.

Ces méthodes ont été programmées pour fonctionner sous Sophemis. Elles sont étudiées dans le cadre d'autres travaux [Hajji 01].

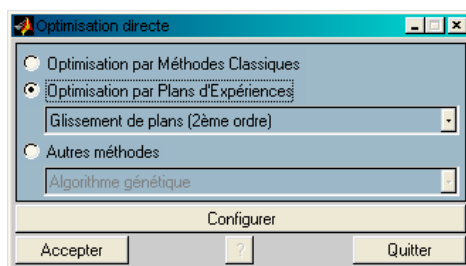


Figure 9 - Boîte de dialogue pour la sélection des méthodes d'optimisation

- Trois groupes de méthodes apparaissent sur la représentation de la boîte de dialogue, faite ci-dessus.
- L'optimisation par méthodes classique désigne tous les algorithmes du type Gradient Conjugué, BFGS, etc.
- La liste déroulante sous la mention « Optimisation par Plans d'Expériences » permet le choix parmi les méthodes d'optimisation présentées dans le chapitre 2.
- Enfin, les autres méthodes (les algorithmes stochastiques ou les techniques heuristiques expérimentales) sont réunies dans le dernier groupe.
- Toutes ces méthodes nécessitent d'être paramétrées. Le bouton « Configurer » permet alors de faire apparaître la boîte de dialogue correspondant à la méthode choisie.

3.4. Méthodes d'optimisation indirecte

Les méthodes d'optimisation indirecte utilisent une approximation préalablement réalisée de la fonction réponse ; ils recherchent le point optimum de ce modèle appelé *fonction d'optimisation*. La solution trouvée est supposée correspondre, dans une certaine mesure, à celle qui aurait été obtenue par la réalisation effective d'une simulation en ce point optimal. La qualité de la fonction d'optimisation est donc essentielle.

Ce principe d'optimisation est par exemple utilisé dans [Caldora 01] [Dyck 99] [Alotto 97].

Il est à la base de travaux notamment sur la construction de la fonction d'optimisation ([Ebner 99] par exemple).

Réalisation des calculs

Sophemis permet d'étendre les possibilités quant à la construction de la fonction d'optimisation. Il est en effet possible de la définir comme étant le résultat d'opérations mathématiques sur des « modélisations simples » calculables, c'est-à-dire à partir :

- de modèles polynomiaux ;
- d'interpolations par tessellation du domaine.

Sophemis utilise les notations suivantes :

- $y_{\text{mod}}(\text{numrep}, o, \text{numcombparam}, \mathbf{x})$: fonction d'optimisation définie par un modèle de la réponse brute n°*numrep*, d'ordre *o* et construit à partir des données numériques obtenues avec la combinaison de paramètres n°*numcombparam*.
- $y_{\text{interp}}(\text{numrep}, \text{numcombparam}, \text{paraminterp}, \mathbf{x})$: fonction d'optimisation définie par une interpolation (par tessellation) de la réponse brute n°*numrep*, construite à partir des données numériques obtenues avec la combinaison de paramètres n°*numcombparam* et utilisant les paramètres d'interpolation *paraminterp* (interpolation linéaire, au plus proche, etc.).

Par exemple, la fonction d'optimisation peut être définie comme valant :

$$y_{\text{mod}}(1,1,1,\mathbf{x}) \times y_{\text{mod}}(1,2,1,\mathbf{x})$$

Dans ce cas, les modèles polynomiaux de la première réponse brute, d'ordre 1 et 2 sont multipliés entre eux, point à point.

Cette technique de construction de la fonction d'optimisation offre donc des degrés de liberté supplémentaires pour la résolution des problèmes d'optimisation.



Il est bien évident que toutes les méthodes d'optimisation directes peuvent être transposées dans le registre actuel : seul l'origine des réponses change.

On peut noter toutefois que certains algorithmes sont ici mieux appropriés, étant donné le faible coût de chaque expérience. Ainsi, toutes les méthodes coûteuses par nature peuvent être favorisées (algorithme génétique, recuit simulé, ...).

4. Exploitation des résultats

Nous sommes ici à l'étape où les calculs demandés (plans d'expériences, optimisations) sont terminés. Les résultats numériques obtenus peuvent alors être analysés, notamment en vue de la détermination d'analyses ultérieures.

Cette exploitation des résultats peut être vue selon 2 angles distincts :

- l'étude numérique des données chiffrées obtenues ;
- les représentations graphiques de ces résultats.

4.1. Etude numérique

Nous voyons ici les processus calculatoires pouvant être menés sur les résultats obtenus pour faciliter leur interprétation.

Une liste simplifiée de ces outils est donnée au paragraphe 5.2.2. .

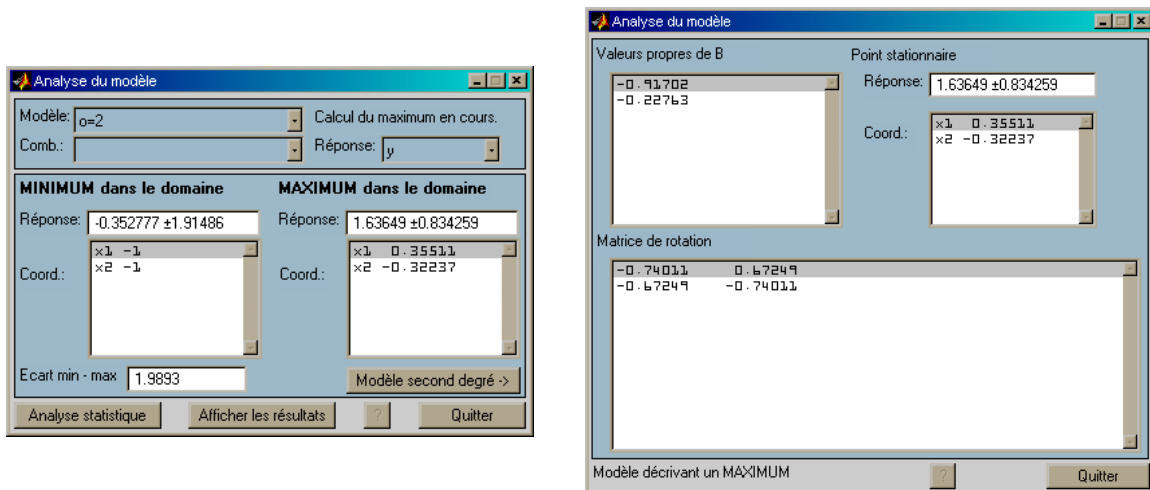


Figure 10 - Boîte de dialogue donnant les caractéristiques des extremums dans le DEP (gauche)
Boîte de dialogue affichant le résultat de l'analyse canonique sur les modèles d'ordre 2 (droite)

- La figure précédente donne l'apparence des boîtes de dialogue utilisées pour l'analyse canonique. Il s'agit donc d'un exemple parmi tous les outils d'étude numérique.
- La boîte de dialogue de gauche est la première à s'afficher : elle renseigne sur le placement et la valeur des extremums des modélisations polynomiales, dans le DEP.
- Lorsque le modèle considéré est du second degré, une analyse canonique peut alors être menée et le bouton « Modèle second degré -> » apparaît alors. Celui-ci permet d'afficher la boîte de dialogue de droite.
- Dans cette fenêtre sont réunies les principales données calculées par l'analyse canonique (position et valeur du point stationnaire, valeurs propres de la matrice B et matrice de rotation).

4.2. Représentation des résultats

Sophemis utilise pleinement les fonctions graphiques fournies par Matlab.

Celles-ci peuvent être différentes suivant la version de Matlab considérée.

4.2.1. Représentation des fonctions

Le terme « fonction » désigne ici toute relation connue entre une grandeur donnée et plusieurs facteurs.

Il peut ainsi s'agir notamment :

- de fonctions analytiques ;
- d'approximations de la fonction réponse ;
- de la fonction d'optimisation ;
- de la fonction de variance ;

Exploitation des résultats

- des fonctions de contraintes.

► 1 facteur – autres facteurs fixés

Parmi les k facteurs, on représente la modélisation de la fonction réponse en fonction d'un seul facteur ; les $k-1$ autres facteurs sont fixés à des valeurs constantes.

Une illustration de ce type de représentation est donné ci-après.

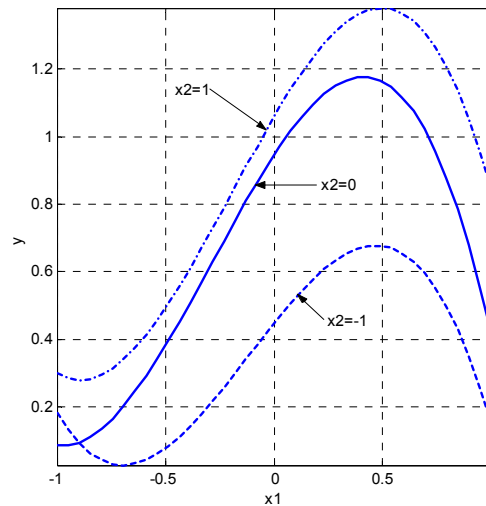


Figure 11 - Exemple de représentation des variations de la réponse (y) selon un facteur (x_1) pour 3 valeurs fixées du second facteur (x_2)

► 2 facteur – autres facteurs fixés

Parmi les k facteurs, on représente la modélisation de la fonction réponse en fonction de 2 facteurs uniquement ; les $k-2$ autres facteurs sont fixés à des valeurs constantes.

Ce type de représentation graphique est le plus courant.

Les variations de la fonction réponse sont figurées par une dimension donnée ; les 2 autres étant prises par les 2 facteurs.

Les figures suivantes donnent des exemples d'utilisation de représentation de la fonction réponse en fonction de 2 facteurs. Matlab offre de nombreuses autres alternatives à ce niveau.

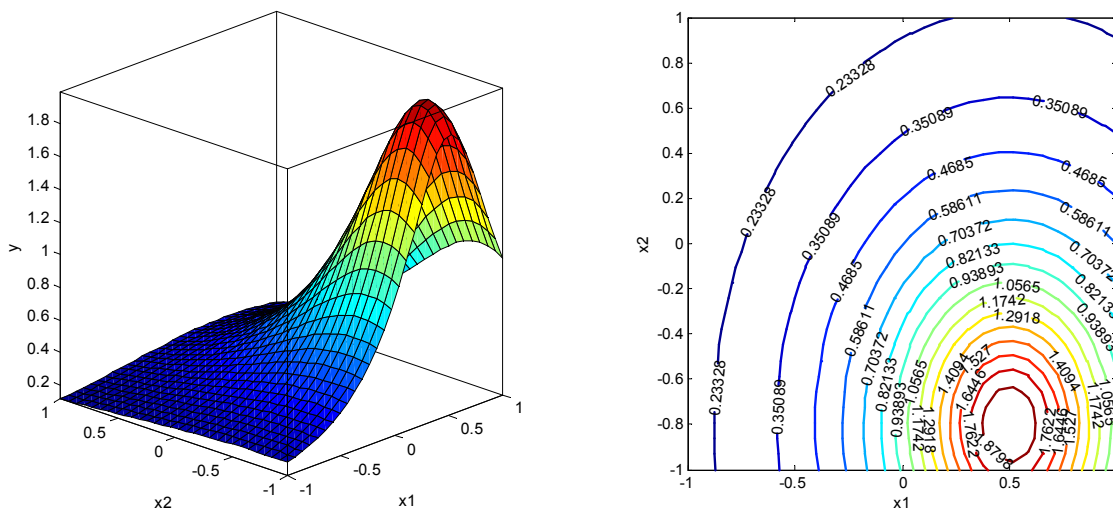


Figure 12 - Exemples de représentations des variations de la réponse (y) selon 2 facteurs (x_1 et x_2)

Gauche : représentation en surface

Droite : représentation en lignes de niveaux

Exploitation des résultats

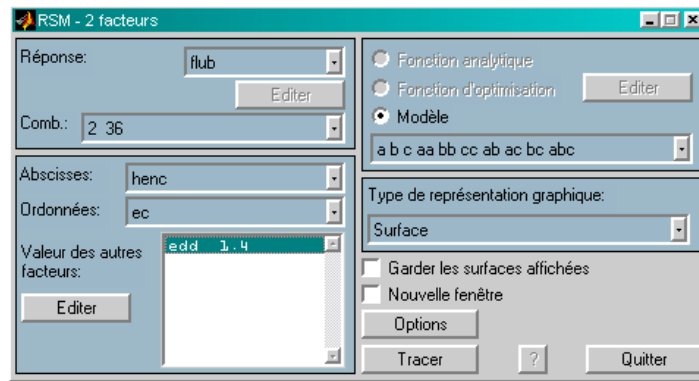


Figure 13 - Boîte de dialogue pour l'affichage des modèles polynomiaux (1 réponse tracée en fonction de 2 facteurs)

- La boîte de dialogue représentée ci-dessus permet de paramétrer et de représenter l'évolution d'une réponse donnée (« flub » ici) en fonction de 2 facteurs sélectionnés (« henc » et « ec »). Les facteurs supplémentaires éventuels sont fixés à des valeurs constantes (c'est ici le cas du facteur « edd » qui prend la valeur constante 1.4). Le bouton « Editer » donne la possibilité de changer ces valeurs calculées par défaut.
- On remarquera ici la présence de combinaisons de paramètres : les données servant à la représentation graphique dépendent de cette information. La valeur affichée « 2 36 » signifie que le premier paramètre prend la valeur 2 et que le second devient égal à 36.
- Les objets représentés par cet outil sont des modèles polynomiaux. On a vu que Sophemis permettait à l'utilisateur de définir ses propres modèles. L'objet à représenter ici correspond à un modèle d'ordre 2 augmenté d'un monôme du troisième ordre ($b_{123} \cdot x_1 \cdot x_2 \cdot x_3$).
- A partir de cette fenêtre, le type de représentation graphique peut être changé. Le bouton « Options » donne accès à une boîte de dialogue où l'aspect visuel du graphique peut être modifié.

➤ 3 facteur – autres facteurs fixés

3 facteurs sont choisis parmi les k pour définir un espace dans lequel sont représentées les variations de la fonction réponse. Les $k-3$ autres facteurs sont fixés à des valeurs constantes.

Selon ces hypothèses, il n'est plus possible de représenter ces variations (puisque'il s'agirait alors d'une hyper-surface à 3 dimensions), mais uniquement plusieurs surfaces d'iso-valeurs. Cela correspond donc aux points de l'espace tri-dimensionnel pour lesquels la fonction réponse prend des valeurs définies.

Ce mode de représentation peut être appréhendé des 2 façons suivantes.

➤ Représentation de modèles du second degré

Comme il a été déjà vu, les modèles polynomiaux du second degré décrivent des iso-surfaces particulières, totalement identifiées suivant les valeurs des valeurs propres λ_i et de $\zeta = y - y_s$ (c'est-à-dire de y).

Exploitation des résultats

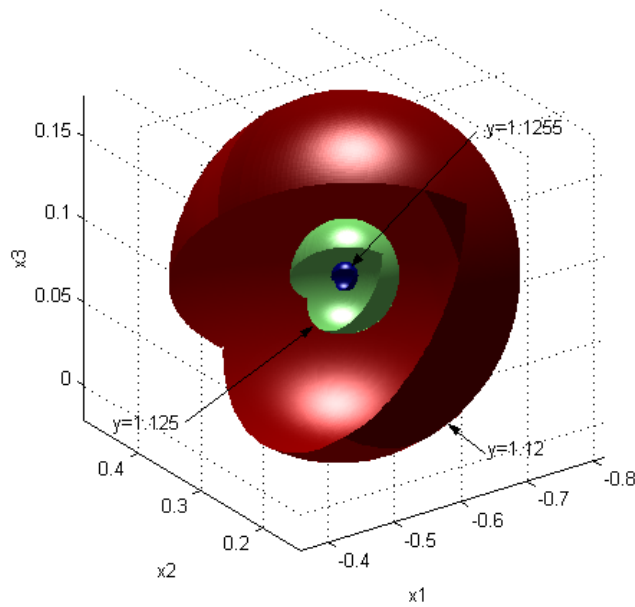


Figure 14 - Exemple de représentation des surfaces d'iso-valeurs de modèles du second ordre
Le point stationnaire est un maximum (où $y_s=1.12553$) ; les iso-surfaces sont des ellipsoïdes
(les surfaces d'iso-valeurs sont ici représentées au $\frac{1}{4}$ pour faciliter la compréhension de leurs placements relatifs)

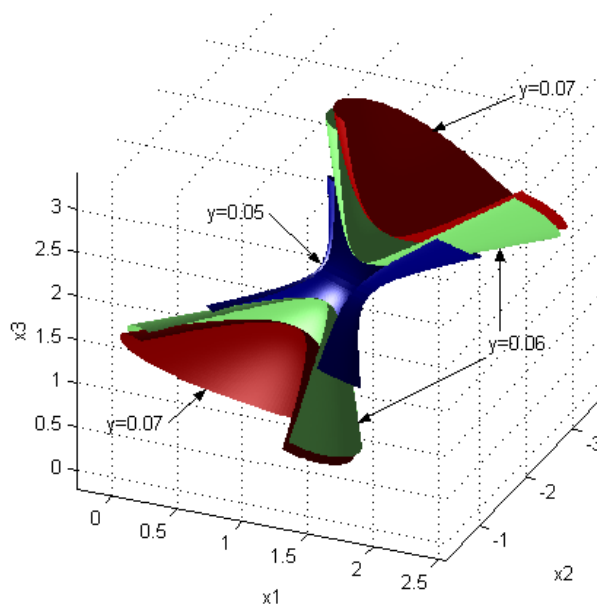


Figure 15 - Exemple de représentation des surfaces d'iso-valeurs de modèles du second ordre
Le point stationnaire est un minimax ($y_s=0.055747$) ; les iso-surfaces sont des hyperboloïdes
(les surfaces d'iso-valeurs sont ici représentées au $\frac{1}{4}$ pour faciliter la compréhension de leurs placements relatifs)

Exploitation des résultats

➤ Maillage de l'espace tri-dimensionnel et interpolation

Les points dispersés dans le DEP peuvent servir de support au maillage de cet espace. A partir des éléments tétraédriques ainsi formés, il est possible de calculer la valeur de la fonction réponse en tout point de ce domaine tri-dimensionnel, par interpolations.

Les éléments créés sont dans la quasi-majorité du temps *linéaires*, c'est-à-dire que la réponse est interpolée linéairement entre leurs nœuds, pris 2 à 2.

Pour utiliser cette méthode, il faut disposer d'un maillageur 3D. Matlab possède cet outil. De plus, les versions 6.x de ce logiciel permettent de calculer aisément les surfaces d'iso-valeurs à partir des volumes pré-maillés.

Ce mode de représentation graphique n'est pas disponible sous Matlab 5.x.

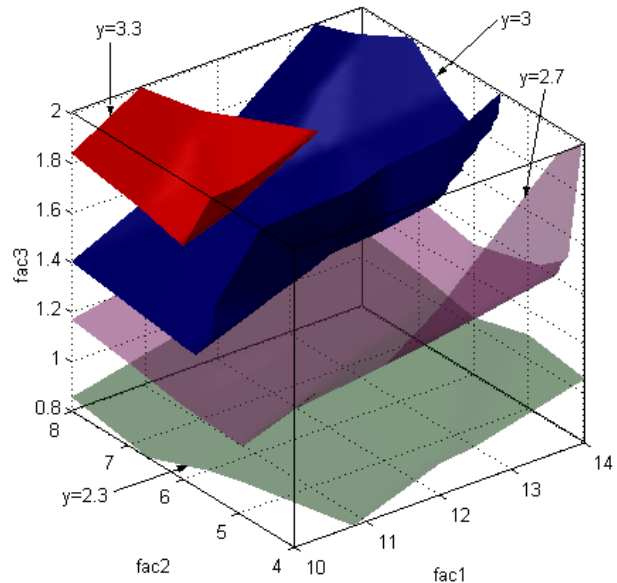


Figure 16 - Exemple de représentation en surfaces d'iso-valeurs

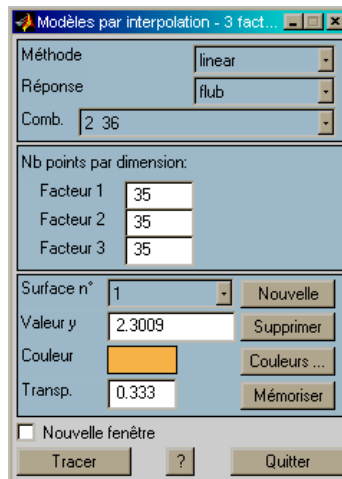


Figure 17 - Boîte de dialogue pour la représentation des réponses en surfaces d'iso-valeurs

- La boîte de dialogue représentée dans la figure 15 n'est accessible que si la version de Matlab est supérieure ou égale à 6. Dans sa configuration actuelle, cette représentation graphique ne peut se faire que lorsqu'il y a exactement $k=3$.
- Les surfaces d'iso-valeurs sont construites à partir d'interpolations reposant sur les points de réponse connue. Dans tous les cas, un maillage d'éléments tétraédriques est construit et les iso-surfaces déduites. Le type d'interpolation utilisé alors peut être choisi dans la liste déroulante en haut de la boîte de dialogue.
- Il faut bien entendu faire également le choix de la réponse à représenter et de la combinaison de paramètre à considérer (s'il y a lieu).
- Cette boîte de dialogue permet alors de configurer les iso-surfaces, c'est-à-dire de préciser la valeur de la réponse qu'elles représentent, leur couleur et leur transparence. L'utilisateur peut en définir autant qu'il le souhaite.

4.2.2. *Autres représentations*

Outre les modèles, Sophemis est parfois amené à représenter graphiquement des données de type différent.

➤ *Succession des simulations*

Les réponses obtenues successivement en chacun des points d'expériences sont reportées en ordonnées. Les numéros des simulations forment l'échelle des abscisses.

Ce type de représentation est typiquement utilisé pour visualiser l'évolution des réponses au cours des processus d'optimisation.

Exploitation des résultats

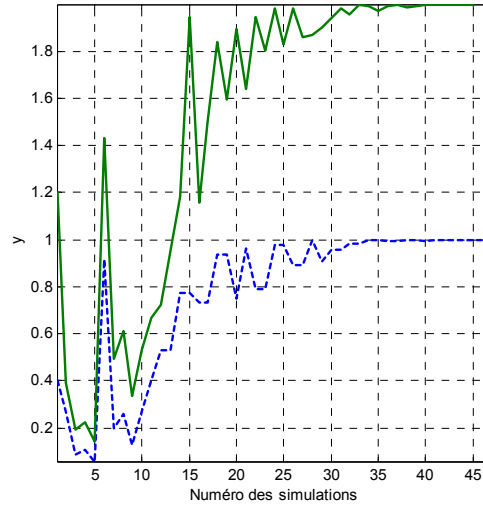


Figure 18 - Valeurs successives de la fonction d'optimisation pour 2 valeurs de combinaisons de paramètres - (exemple de processus d'optimisation)

➔ Représentations propres au screening

L'analyse de screening utilise un certain nombre d'outils graphiques propres.

Le paragraphe 4.4.2. du premier chapitre a montré les principales représentations des effets principaux et des interactions, c'est-à-dire :

- la représentation des effets principaux par un segment de droite ;
- la représentation des interactions entre 2 facteurs par 2 segments de droite ;
- la représentation des effets (des facteurs principaux et des interactions) par des histogrammes ;
- la représentation de la normalité de répartition de la valeur de effets, par le graphe de Daniel.

Toutes ces solutions sont gérées par Sophemis.

On notera également la possibilité de représenter en une seule fois les effets (par segments de droite) de tous les facteurs et interactions d'ordre 2. La figure suivante en donne un exemple.

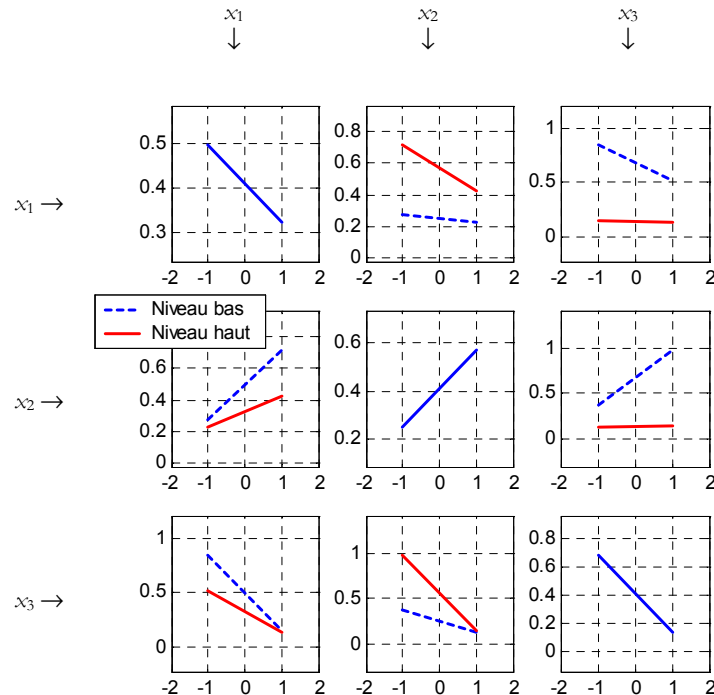


Figure 19 – Exemple de représentation des effets des facteurs principaux et des interaction d'ordre 2 – $k=3$ (exemple)

5. Autres caractéristiques

5.1. Calculs multi-ordinateurs

5.1.1. Description

La majorité des calculs assurés par Sophemis sont relatifs aux plans d'expériences.

De manière générale, il faut réaliser N expériences ($N > 1$) avant de pouvoir déduire une information (le modèle de la réponse étudiée). Ces N expériences sont indépendantes entre elles.

Les méthodes d'optimisation par PE utilisent cette même méthode, à chacune de leur étape de calcul.

L'idée naturelle est donc de répartir le calcul de ces N expériences sur plusieurs machines différentes, et de réaliser ces simulations de façon simultanée.

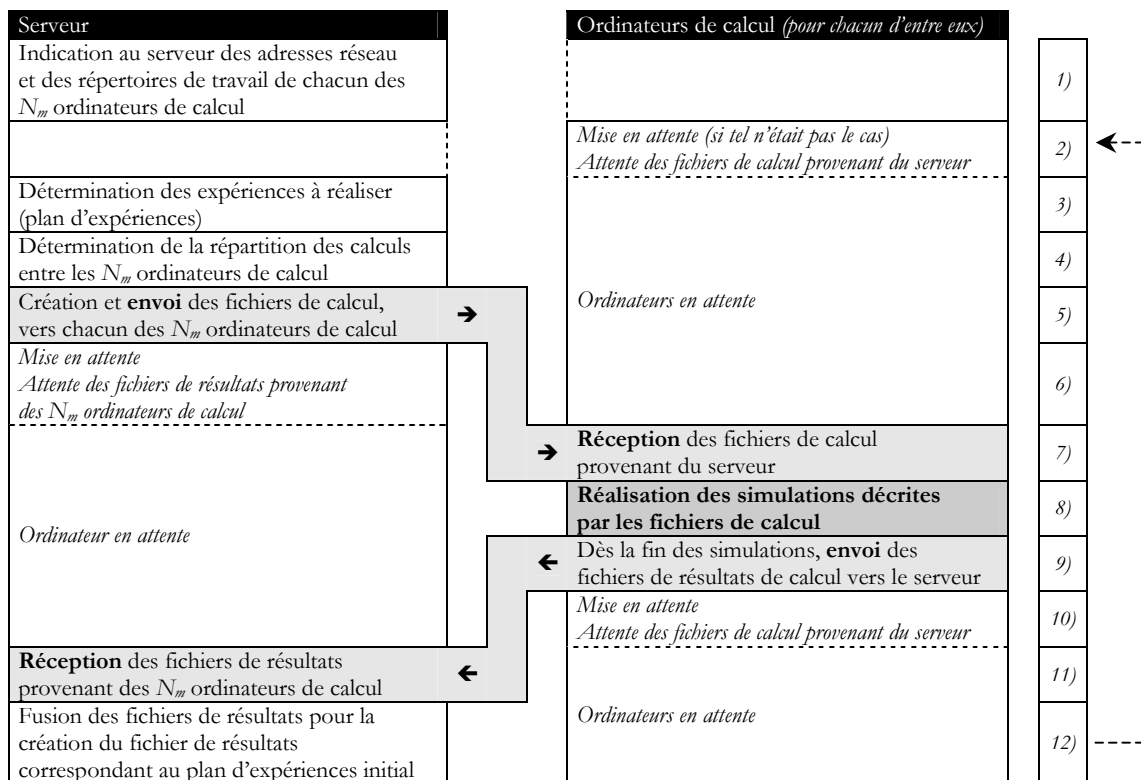
On parlera alors de *calculs multi-ordinateurs*, ou de *calculs distribués*.

La direction des opérations de simulation est assurée par un *ordinateur serveur*, ou plus simplement *serveur*. Celui-ci envoie les informations relatives aux simulations à réaliser à des *ordinateurs de calcul*, qui effectuent les simulations à la demande. On notera par la suite N_m le nombre d'ordinateurs de calcul disponibles.

L'architecture utilisée est donc du type maître - esclaves. Elle demande donc l'emploi de $N_m + 1$ machines.

Pour la réalisation de tout plan d'expériences, on peut distinguer les étapes suivantes (processus simplifié) :

Tableau 6 - Principales étapes pour le calcul d'expériences virtuelles sur ordinateurs distants



- 1) La première étape consiste en l'identification des ordinateurs entre eux. Ainsi, le serveur doit connaître les adresses réseau des machines de calcul vers lesquels il va envoyer les ordres de

Autres caractéristiques

simulation. En plus de cela, il lui faut savoir vers quels répertoires (des machines distantes) il devra envoyer les fichiers de calcul.

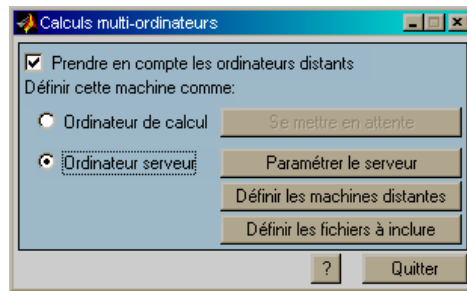


Figure 20 - Boîte de dialogue principale pour le paramétrage des calculs distribués

- On retrouve dans la boîte de dialogue représentée ci-dessus les principaux points cités par le Tableau 6.
 - Dans un premier temps, l'ordinateur courant doit être configuré soit en ordinateur de calcul (auquel cas il doit simplement se mettre en attente d'informations envoyées par le serveur) soit en ordinateur serveur. Dans ce dernier cas, des informations supplémentaires doivent être données.
 - Le serveur doit être identifié (nom de l'ordinateur dans le réseau) et le répertoire de travail précisé. C'est ce que permet la boîte de dialogue ouverte par l'appui sur le bouton « Paramétrer le serveur ».
 - Le bouton « Définir les machines distantes » permet de rentrer, un à un, les répertoires de travail (accessibles par le serveur) de chacun des ordinateurs distants.
 - Enfin, il peut arriver que la réalisation des simulations par les machines de calcul nécessite des fichiers spécifiques. Une autre boîte de dialogue (ouverte par le bouton « Définir les fichiers à inclure ») permet de prendre en compte ces fichiers qui seront envoyés aux machines distantes.
- 2) Afin de pouvoir utiliser des machines distantes comme ordinateurs de calcul, il faut mettre chacune d'elles en position d'attente. Sur chacun des ordinateurs de travail est alors créé un sous-répertoire de nom `\JobNet` dans lequel les fichiers de calcul envoyés par le serveur seront copiés (si ce répertoire existait déjà, les fichiers qu'il peut contenir sont supprimés).
 - 3) Cette étape correspond au lancement des calculs par l'utilisateur. Il peut s'agir de la réalisation d'un simple plan d'expériences, ou également d'un algorithme d'optimisation par PE. Dans tous les cas, l'élément de base à calculer est le plan d'expériences (i.e. une série de N simulations indépendantes entre elles) défini par les paramètres de la session en cours, et/ou de l'itération actuelle de l'algorithme.
 - 4) Suivant la valeur N du nombre d'expériences à réaliser et du nombre N_m d'ordinateurs de calcul disponibles, il y a déduction de la meilleure répartition possible des simulations afin de minimiser le temps global de calcul. Ce point est abordé au paragraphe 5.1.2. .
 - 5) L'ordinateur serveur crée les fichiers dits de « calculs ». Ceux-ci contiennent toutes les informations nécessaires aux ordinateurs (de calcul) pour réaliser les simulations demandées. Il s'agit principalement :
 - du fichier de session contenant tous les paramètres relatifs aux calculs courants ;
 - d'un fichier donnant les coordonnées des points d'expériences à réaliser ;
 - d'un fichier d'informations diverses incluant entre autres :
 - l'adresse réseau du serveur ;
 - le répertoire du serveur dans lequel envoyer les fichiers de résultats obtenus après simulations ;
 - le numéro de machine attribué par le serveur ; cette information est utilisée pour nommer les fichiers de résultats renvoyés, et après coup pour faciliter la fusion des résultats renvoyés par tous les ordinateurs de calcul.
 - des fichiers requis pour la réalisation des simulations électromagnétiques (fichiers de commande OPERA, fichiers caractérisant les matériaux modélisés, ...).

Ces différents fichiers sont envoyés aux N_m ordinateurs distants, chacun recevant des tâches différentes et adaptées.

Autres caractéristiques

- 6) Les ordres de simulation étant donnés, le serveur se met en attente des résultats des ordinateurs de calcul.

Les opérations 7, 8, 9 et 10 concernent chacun des N_m ordinateurs de calcul.

- 7) Dès la constatation de la présence de fichiers dans le sous-répertoire \JobNet, l'ordinateur distant :
 - attend de recevoir un témoin de fin de transmission, permettant ainsi de s'assurer d'avoir reçu l'ensemble des fichiers nécessaires à la réalisation des simulations ;
 - vérifie la validité de chacun des fichiers reçus.
- 8) Les fichiers de calcul reçus sont utilisés pour la réalisation des simulations. Ainsi, le fichier de session sert à initialiser tous les paramètres de Sophemis relatifs aux calculs à venir. Les coordonnées des points d'expériences désignent les simulations à effectuer. Les résultats de chacune d'elles sont mémorisés dans un fichier de résultats dont le nom contient le numéro d'identification de l'ordinateur de calcul courant.
- 9) Les simulations terminées, le fichier de résultats est envoyé dans un répertoire précis de l'ordinateur serveur (son nom figure dans le fichier d'informations envoyé parmi les fichiers de calculs). Cet envoi indique donc au serveur que la machine de calcul courante a terminé les travaux demandés.
- 10) L'ordinateur de calcul retourne à une position d'attente (cf. étape n°2) afin de pouvoir traiter de nouvelles demandes de simulations envoyées par le serveur.
- 11) En configuration d'attente, le serveur constate alors la réception dans son répertoire de réception des fichiers de résultats envoyés par les ordinateurs de calcul. Pour chaque fichier reçu, il en vérifie la validité. La fin du processus d'attente est déterminée par la réception des N_m fichiers de résultats (autant que de machines distantes utilisées et pour lesquelles des ordres de simulation ont été donnés).
- 12) En utilisant les numéros des machines distantes figurant dans les noms des fichiers de résultats, le serveur met les données numériques (valeurs des réponses) de ces fichiers dans l'ordre correct, et les fait correspondre avec les coordonnées des points d'expériences.

L'ordinateur serveur est alors à même de traiter les valeurs reçues comme celles obtenues de façon plus usuelle par une machine unique.

Dans le cadre d'une optimisation par PE, le traitement de ces données peut impliquer la réalisation d'un autre plan d'expériences ; le même processus se répète alors : il y a rebouclage vers l'étape n°2.

5.1.2. Considérations sur les temps de calcul

L'utilisation de N_m ordinateurs de calcul permet, bien entendu, de réduire considérablement de temps global nécessaire (que l'on notera T) pour la réalisation de N simulations.

Si les calculs étaient réalisés sur un seul ordinateur, on peut supposer (on le fera à l'avenir, même s'il s'agit d'une simplification) que toutes les simulations ont été effectuées en des temps constants ; soit t cette durée. Avec ces hypothèses, le temps global de calcul est donc :

$$T=N.t$$

Lors de l'utilisation de N_m ordinateurs de calcul ($N_m > 1$), il est inexact de penser que T s'écrit alors $N.t/N_m$. A cela, plusieurs raisons.

Autres caractéristiques

➤ Tous les ordinateurs de calcul ne sont pas obligatoirement identiques

Les machines de calcul ont de grandes chances d'être différentes entre elles, et par conséquent les temps de simulation sur chacune d'elles également.

La grandeur t n'est donc plus exploitable et il est alors nécessaire de définir une durée t_i de simulation pour chaque ordinateur i .

Par conséquent, en raison de ces différences, afin de minimiser le temps global T , il se peut que chaque machine n'ait pas le même nombre d'expériences à calculer.

➤ N n'est pas obligatoirement un multiple de N_m

Prenons un exemple pour lequel toutes les machines sont identiques (avec t le temps de réalisation d'une seule simulation).

Si l'on dispose de $N_m=4$ ordinateurs de calcul et que $N=9$ expériences doivent être faites, il faudra dans tous les cas attendre $T=3.t$ avant d'obtenir les résultats.

En effet, dans un premier temps, les 4 machines sont utilisées pour calculer les 4 premières expériences ; les résultats sont alors obtenus en même temps, au bout d'une durée t . Le même processus se répète pour la réalisation des 4 expériences suivantes. Une période de $2.t$ s'est alors écoulée. Reste alors une simulation à réaliser. Pour cela, un seul ordinateur est nécessaire. Au bout du compte, il faut donc bien $3.t$ (ou 3 cycles de durée t) pour réaliser 9 expériences sur 4 machines identiques.

Pour le calcul de ces 9 simulations, il y a eu implicitement la décomposition $4+4+1$. Il est alors intéressant de noter que la décomposition $3+3+3$ pouvait être réalisée avec la même durée de calcul ($3.t$) ; de plus, elle a l'avantage de ne nécessiter que 3 ordinateurs au lieu de 4. Cette simple considération permet de s'interroger sur le nombre minimal de machines de calcul à utiliser (N_{mi}).

➤ Il faut prendre en compte les temps de transfert des fichiers de calcul et de résultats

La communication entre le serveur et les ordinateurs de calcul ne sont pas de « durée nulle ».

Ainsi, lors de l'envoi des fichiers de calcul, le serveur procède machine par machine. Les calculs commençant dès la réception de ces fichiers, il y a donc des décalages entre les instants où débutent les simulations. Ce phénomène est d'autant plus visible que le nombre d'ordinateurs distants est important.

Néanmoins, cela est marginal vis-à-vis des deux points précédents. Il ne sera donc plus considéré par la suite.

La connaissance de la valeur exacte du temps nécessaire T au calcul des N expériences n'est pas un but en soi. Cependant, on cherchera systématiquement à minimiser T , quelques soient les situations rencontrées : les ordinateurs de calcul ont des performances similaires ou sont de puissances différentes.

5.1.2.1 Ordinateurs de calcul identiques

On considère N_m machines identiques pour la réalisation de N simulations ; chacune d'elles nécessite un temps t pour être menée à bien.

Le nombre minimum de cycles à effectuer est :

$$n = E\left(\frac{N}{N_m}\right) + 1$$

Le temps global de simulation est toujours de $T = n.t$.

Il s'agit donc du temps minimum d'attente avant l'obtention des résultats des N expériences.

On peut alors déduire le nombre minimum de machines nécessaires pour la réalisation de N simulations en un temps T :

$$N_{mi} = \frac{N}{n} \quad \text{si } (N/n) \in \mathbf{N}$$
$$N_{mi} = E\left(\frac{N}{n}\right) + 1 \quad \text{si } (N/n) \notin \mathbf{N}$$

Autres caractéristiques

N_{mu} désigne le nombre d'ordinateurs de calculs *utilisés*. On a toujours $N_{mu} \leq N_m$.

5.1.2.2 Ordinateurs de calcul différents

On considère ici N_m machines différentes entre elles destinées au calcul de N simulations. La machine n° i permet de réaliser une expérience en un temps t_i .

Comme il a été vu plus haut, la recherche de la solution optimale à ce problème revient à attribuer un nombre différent de simulations à chaque machine. Soit n_i le nombre d'expériences attribuées à la machine n° i .

On peut écrire :

$$\sum_{i=1}^{N_m} n_i = N$$

L'objectif théorique serait de trouver les valeurs des n_i donnant un temps global de calcul T minimal.

Dans un second temps, la recherche de la configuration utilisant le moins de machines possible peut éventuellement être menée (valeur de N_{mu}).

Le temps global de calcul pour l'ordinateur n° i ($i=1, 2, \dots, N_m$) est donné par :

$$T_i = n_i \cdot t_i$$

Par conséquent, le temps de calcul T sera donné par la plus grande valeur des T_i (les calculs sur chaque machine étant simultanés).

On recherche donc les valeurs des n_i afin de minimiser T :

$$\left\{ \begin{array}{l} T = \max(n_1 \cdot t_1, n_2 \cdot t_2, \dots, n_{N_m} \cdot t_{N_m}) \\ \sum_{i=1}^{N_m} n_i = N \\ t_i \text{ connus} \end{array} \right.$$

Il s'agit donc d'un problème « classique » de minimax. Cependant, la difficulté tient au fait que l'optimisation est discrète (les n_i sont des entiers).

La résolution dans \mathbf{R} de ce problème d'optimisation est simple. Elle donne malgré tout une solution non entière, qui n'est donc pas directement applicable.



Dans la pratique, il n'y a pas d'estimation théorique des n_i ou des t_i avant la réalisation des expériences. On va plutôt utiliser les valeurs t_i constatées, c'est-à-dire déduites de la réalisation d'au minimum une expérience sur chacune des N_m machines de calcul. La répartition des calculs est alors déduite de ces temps moyens (t_i).

La solution retenue procède comme suit. Parmi les N expériences à réaliser, chacun des N_m ordinateurs de calcul en réalise une. Dès qu'une machine distante n° i termine ses calculs et envoie ses résultats au serveur, celui-ci mémorise le temps passé par cette i^{me} machine : une première estimation de la valeur t_i est donc réalisée. Si le serveur demande à ce que cette même machine réalise une deuxième expérience, le temps t_i pourra alors être affiné grâce à la mesure du temps pris pour le calcul de cette nouvelle expérience. Ainsi, le serveur mesure et corrige les estimations des temps moyens t_i pour chaque ordinateur distant, au fur et à mesure de la réalisation des N expériences.

Dès que tous les t_i sont connus, ils servent à choisir les machines distantes pour que les N simulations soient calculées le plus rapidement possible. Les valeurs des n_i ne sont donc jamais calculées.

Cette méthode est intéressante dans la mesure où elle constitue une solution simple au problème ; elle permet indirectement de prendre en compte les différences de vitesse de transfert des données via le réseau reliant les machines de calcul avec le serveur. Le principal défaut est certainement l'accroissement du nombre d'informations échangées sur ce réseau, augmentant ainsi les temps de transfert.

Autres caractéristiques

5.1.3. Remarques sur les calculs distribués

La répartition et la réalisation simultanée des expériences sur plusieurs ordinateurs permet, tout naturellement, de diminuer efficacement le temps nécessaire au calcul de chaque plan.

Le calcul multi-ordinateurs présente cependant le principal inconvénient de nécessiter une quantité de « matériels » conséquente.

En plus de l'ordinateur serveur - qui ne nécessite pas d'avoir des capacités de calcul importantes -, il faut disposer d'ordinateurs capables de réaliser au mieux les simulations électromagnétiques demandées. Il s'agit donc de machines dotées de grandes quantités de mémoire et de processeurs rapides ; leurs disques durs doivent être de grande capacité pour le swap éventuellement requis, et surtout, la quantité de mémoire vive doit être conséquente pour garantir la rapidité des échanges de données et donc des calculs.

Toutes ces machines doivent être connectées en réseau, afin de pouvoir communiquer entre elles (échanges serveur ↔ ordinateurs de calcul).

Il faut ajouter également que les simulations sont réalisées par des applications qui doivent être installées sur chacune des machines de calcul (logiciels OPERA de Vector Fields dans notre cas). Étant donné le système de licence adopté par de nombreux éditeurs de logiciels, l'aspect économique vient alors imposer des limites quant au nombre N_m de machines réellement utilisables. Cette limitation logicielle est souvent plus contraignante que celle représentée par le matériel informatique lui-même.

A noter enfin, que les N_m+1 machines demandées lors de séances multi-ordinateurs doivent en principe être réservées à la réalisation de ces calculs.

Réaliser des opérations supplémentaires en parallèle ne peut que conduire à la diminution des performances de chacune des applications lancées, et en particulier de celles du programme de simulation.

Par conséquent, l'utilisation de sessions multi-utilisateurs implique la mobilisation exclusive des ordinateurs utilisés, pendant une durée en général indéterminée (relativement aux algorithmes d'optimisation par exemple)...

5.2. Modularité

Sophemis a été construit sur le principe de la modularité de ses fonctionnalités.

La structure adoptée est celle d'un noyau (*kernel*) auquel il est possible d'ajouter (et après coup d'enlever) des modules.

Un module est un ensemble de fichiers permettant d'augmenter des fonctionnalités de la composante souche (noyau) de l'application Sophemis.

La gestion des modules (création et suppression, inclusion et exclusion de Sophemis, etc.) est assurée par une application satellite appelée MatOptTools.

5.2.1. Rôle du noyau

Le noyau assume les tâches minimales suivantes.

➤ Gestion des fichiers

La création, l'ouverture, la lecture, la modification, la fermeture et la suppression des différents types de fichiers utilisés sont assurées par le noyau. Il s'agit ainsi :

- des fichiers de session (mémorisation de tous les paramètres choisis par l'utilisateur pour la réalisation du type de calculs choisis) ;
- des fichiers de résultats (enregistrement des résultats de simulation, ainsi que des paramètres de session ayant été appliqués pour les obtenir) ;
- les fichiers de fonction analytique (définition de fonctions mathématiques d'expression analytique connue, pouvant être utilisées en lieu et place de simulations électromagnétiques) ;
- les fichiers de plans d'expériences (définition des coordonnées de séries de points d'expériences définis par l'utilisateur) ;
- les fichiers temporaires (il s'agit dans la plupart des cas de fichiers donnant des informations sur les simulations réalisées ou en cours ; ainsi, par exemple, le fichier Temps.tmp enregistre les dates et horaires de début et de fin de chaque simulation).

➤ Gestion des variables de fonctionnement

Autres caractéristiques

Toutes les valeurs des champs de boîtes de dialogue que l'utilisateur remplit, sont enregistrées dans des variables. Il existe de plus des variables particulières, propres au fonctionnement interne de l'application. Le noyau se charge ainsi :

- de récupérer la valeur de chacune d'entre elles, à la demande ;
- d'attribuer de nouvelles valeurs pour certaines d'entre elles, à la demande, en respectant leurs caractéristiques propres (type, forme, condition, ...) et en vérifiant leur validité.

➤ Lancement des calculs

Tous les calculs (qu'ils soient issus ou non de modules externes) sont lancés à partir d'une procédure du *kernel*. Avant le lancement effectif de ces méthodes de calcul, il y a une vérification préalable de la validité des données qui seront utilisées par la suite. Toute anomalie est reportée, et l'utilisateur est invité à la corriger afin de poursuivre.

➤ Gestion des données expérimentales

Les expériences à réaliser ainsi que les résultats correspondants sont enregistrés dans des variables gérées par le noyau. Ces données sont extrêmement importantes, puisqu'elles sont principalement utilisées lors du *post-processing*, c'est-à-dire lors de l'analyse des résultats. Elles servent alors de données d'entrée pour les principaux modules de calcul.

➤ Communication avec les modules

L'utilisation des modules au sein de Sophemis résulte de modifications partielles du code du noyau, lors de l'inclusion de chacun d'entre eux. Le *kernel* a été conçu afin de prendre en compte ce genre d'opérations.

5.2.2. Modules

Jusqu'à présent, 63 modules ont été développés.

On peut les ranger en 6 grandes classes, selon leurs fonctionnalités.

5.2.2.1 Modules de calcul

Ils permettent d'analyser les données mémorisées lors de la réalisation de plans d'expériences ou de processus d'optimisation.

Dans l'exemple d'un plan d'expériences de screening, l'outil permettant de calculer les effets des facteurs principaux et des interactions est un module. Il est en de même pour ce qui est de la réalisation de l'analyse de la variance, du calcul des résidus, etc.

Concernant le screening, les modules de calculs pris en charge par Sophemis sont liés aux fonctions listées ci-dessous :

- analyse de la variance ;
- déduction des contrastes des plans fractionnaires ;
- calcul des effets ;
- calcul des résidus.

Les modules de calcul relatifs à la RSM ont les fonctions suivantes :

- analyse canonique ;
- points minimum et maximum des modèles dans le DEP ;
- analyse statistique (critères d'ajustement, coefficients de modèles polynomiaux, etc.) ;
- analyse de la variance ;
- calcul des valeurs de la fonction d'erreur de prédiction ;
- calcul des valeurs de la réponse donnée par les modèles polynomiaux ;
- calcul des valeurs de la réponse donnée par des méthodes d'interpolation ;
- analyse des observations (calcul des résidus, distances de Cook, etc.) ;
- création de plans D-optimaux approximatifs ;
- création de plans D-optimaux exacts ;
- calcul des principaux critères d'optimalité (critères A, D, E, etc.) ;
- écriture et classement des meilleurs sous-modèles.

5.2.2.2 Modules de représentation graphique

Autres caractéristiques

Les résultats numériques donnés par les modules de calculs peuvent être efficacement complétés par les modules de représentation graphique. Ceux-ci donnent la possibilité d'afficher graphiquement les données expérimentales, et peuvent faciliter leur compréhension.

On peut faire encore une fois la distinction entre l'analyse de screening et celle de RSM.

Les modules de représentation graphique servant à l'analyse de screening permettent d'assurer les fonctions suivantes :

- affichage des effets par segments de droites (effets principaux et interactions d'ordre 2) ;
- représentation des effets par histogrammes (effets principaux et interactions) ;
- tracé du graphe de Daniel.

Pour ce qui est des outils graphiques utilisés dans la méthodologie des surfaces de réponse, il est possible de bénéficier des modules permettant de représenter :

- les valeurs calculées de la réponse en fonction des valeurs mesurées correspondantes ;
- la fonction d'erreur de prédiction (pour $k=2$ uniquement) ;
- la modélisation polynomiale de la réponse, en fonction de 1 facteur (les autres étant fixés) ;
- la modélisation polynomiale de la réponse, en fonction de 2 facteurs (les autres étant fixés) ;
- la modélisation polynomiale d'ordre 2 de la réponse, en fonction de 3 facteurs (les autres étant fixés ; la réponse doit ici être représentée par des surfaces d'iso-valeurs) ;
- la fonction d'interpolation de la réponse en fonction de 2 facteurs (les autres étant fixés) ;
- la fonction d'interpolation de la réponse en fonction de 3 facteurs (les autres étant fixés ; la réponse doit ici être représentée par des surfaces d'iso-valeurs).

On citera de plus, 3 autres modules, dont les tâches sont de représenter respectivement :

- les valeurs de la réponse suivant leur ordre de calcul ;
- la fonction analytique servant de source aux valeurs des réponses ;
- la fonction d'optimisation utilisée par les optimisations indirectes.

Enfin, 2 autres modules ont été créés pour changer simplement les paramètres purement graphiques de chaque figure (palettes de couleurs, titre et étiquettes des axes, orientation, etc.).

5.2.2.3 Modules de plans d'expériences

Chaque type de plan d'expériences est défini par un module, qui doit être en mesure de :

- gérer les paramètres particuliers à ce PE (par exemple, pour définir un plan de Doehlert, il faut connaître son type parmi D-1, D-2, etc.) ;
- donner la liste des coordonnées des expériences, en fonction des valeurs des paramètres vus ci-dessus ;
- donner les définitions des monômes composant le modèle pouvant être calculé grâce à ce PE (pour les plans de screening uniquement).

13 modules ont ainsi été définis (les plans factoriels complets et fractionnaires à 2 niveaux par facteur sont gérés par le même module).

5.2.2.4 Modules d'optimisation

Ce sont les modules les plus intéressants au niveau de leurs possibilités, puisqu'ils gèrent le déroulement entier des processus d'optimisation : mémorisation des paramètres de l'algorithme et automatisation des tâches itératives (ou récursives) demandées par la méthode.

Ces modules utilisent les données de la session en cours et renvoient la liste de tous les points d'expériences utilisés, accompagnés des valeurs respectives des réponses. Ces données permettent d'utiliser dans un second temps les modules de calcul et/ou de représentation graphique éventuellement installés.

Chaque méthode d'optimisation est gérée par un module ; il y en a actuellement 14 (certaines méthodes sont encore au stade expérimental et n'ont pas été reportées dans ce rapport).

5.2.2.5 Modules de communication avec les programmes externes

Autres caractéristiques

Les simulations électromagnétiques étant réalisées par des programmes externes (OPERA®), il est nécessaire de faire usage d'une interface entre ces applications et Sophemis.

Ainsi, le module jouant ce rôle avec OPERA, donne la possibilité de :

- définir les fichiers de construction de modèle et d'extraction de données ;
- définir le type de solveur utilisé, ainsi que ses paramètres associés ;
- définir les fichiers à supprimer après chaque simulation.

Sont listées ci-après les caractéristiques des modules disponibles dans la version courante de Sophemis :

- gestion des paramètres utilisés pour la réalisation d'expériences par simulations éléments finis ;
- configuration du solveur statique (2D) d'OPERA ;
- configuration du solveur vitesse (2D) d'OPERA (prise en compte des déplacements relatifs linéaires) ;
- configuration du solveur statique (3D) d'OPERA.

5.2.2.6 Modules de gestion des calculs distribués

Un seul module assure toutes les tâches demandées pour la gestion des calculs distribués. Il réalise toutes les opérations résumées dans le Tableau 6.



La figure suivante donne une représentation de principe de l'architecture fonctionnelle de Sophemis.

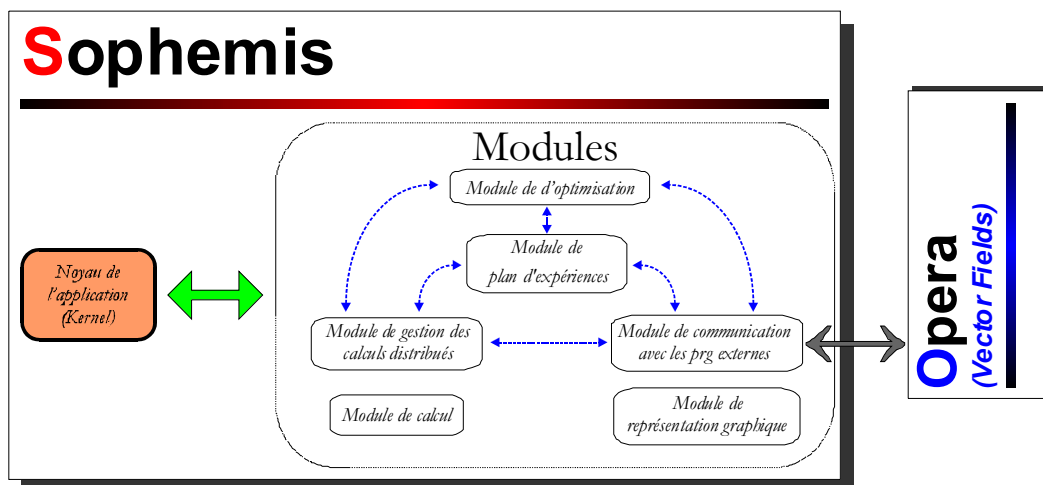


Figure 21 - Architecture fonctionnelle de Sophemis

On notera brièvement que certains modules nécessitent les fonctionnalités offertes par d'autres modules : les ordres respectifs d'installation sont donc importants.

Autres caractéristiques

5.3. Aide HTML

Tableau 7 - Structure hiérarchique de l'aide HTML de Sophemis

Utilisation du logiciel	Calculs réalisables	Screening
		Surfaces de réponse (RSM)
		Optimisation directe
		Optimisation indirecte
	Exemples commentés	Calculs multi-ordinateurs
		Fonction test analytique
		Définition de la session
		Etude de screening
		Etude de RSM
	Menu général	Optimisations directes
		Optimisations indirectes
	Boîtes de dialogue	Valeurs
		Configuration
		Screening
Surfaces de réponse		
Optimisation directe		
Optimisation indirecte		
Méthodes d'optimisation		
Représentations graphiques		
Calculs multi-ordinateurs		
Annexes		
Dysfonctionnements	Anomalies résolues	
	Anomalies non résolues	
Améliorations possibles		
Fonctions de base	Liste alphabétique	A-F
		G-L
		M-R
		S-Z
	Liste thématique	Fonctions classiques d'optimisation unidimensionnelles sans contrainte
		Fonctions classiques d'optimisation multidimensionnelles sans contrainte
		Fonctions classiques d'optimisation multidimensionnelles avec contraintes
		Fonctions d'optimisation automatiques par plans d'expériences
		Autres fonctions d'optimisation
		Fonctions de création de matrices
		Fonctions de création de plans d'expériences
		Fonctions de calcul des coefficients des modèles polynomiaux
		Fonctions de RSM
		Fonctions de screening
		Fonctions de représentation graphique
		Fonctions de calcul sur matrices
		Fonctions de calcul sur vecteurs
		Fonctions de calcul sur scalaires
		Fonctions de dénombrement
		Fonctions statistiques
Fonctions de gestion de polynômes		
Autres fonctions		
Notes sur le fonctionnement interne de l'application	Principes généraux	Modules – Registres – Variables globales
		Calculs
		Fichiers
		Compilation éventuelle MATLAB
	Base de modules	
Base de registres		
Variables globales - Macros		
Plan de l'aide		

Sophemis bénéficie d'une aide écrite en langage HTML.
Cet outil peut donc être consulté sur tout type de machine.

Le tableau précédent donne la structure et le type des données pouvant y être consultées.
L'aide s'articule ainsi selon 3 axes majeurs.

➤ **Utilisation du logiciel**

Cette partie se concentre sur l'utilisation pratique de Sophemis.

Dans un premier temps, les possibilités du logiciel sont présentées de manière générale. Ensuite, ces propos sont illustrés par la description détaillée d'une analyse complète (screening, RSM et optimisations)

Autres caractéristiques

portant sur une fonction analytique donnée. L'utilisateur est donc à même de se familiariser avec l'application Sophemis et d'apprécier les possibilités qu'elle offre, en réalisant une à une les étapes listées et détaillées dans le cadre de cette application particulière.

Cette partie donne également une description de toutes les boîtes de dialogue de Sophemis. Celles-ci sont regroupées suivant le type de fonction qu'elles assurent.

Il est possible de retrouver ces boîtes de dialogue en utilisant les liens hyper-textes disposés dans la page affichant le contenu complet du menu général de Sophemis.

L'utilisation courante du logiciel a fait apparaître quelques dysfonctionnements suivant le type d'installation. 2 pages de l'aide y sont spécialement dédiées ; elles mentionnent dans le même temps les solutions à y apporter.

Enfin, afin de favoriser le développement et l'amélioration des fonctionnalités offertes par Sophemis, une page a été créée pour garder les propositions éventuelles de ses utilisateurs.

➤ Fonctions de base

On ne considère ici que les fonctions dont l'emploi n'est pas restreint à l'usage exclusif de Sophemis. On exclut donc en particulier toutes les procédures gérant l'interface graphique (menus, boîtes de dialogue) du logiciel.

Ces fonctions de base peuvent ainsi être exécutées sans l'aide de Sophemis. Elles ont été classées de 2 manières différentes : alphabétiquement et thématiquement.

➤ Notes sur le fonctionnement interne de l'application

L'aide de Sophemis a bien entendu été écrite afin de fournir une assistance aux utilisateurs du logiciel, tant dans ces fonctionnalités que dans ses procédures de calcul.

Elle a également été conçue pour favoriser la pérennité de ce projet. Pour faire évoluer Sophemis, il est en effet nécessaire de comprendre son mode de fonctionnement : c'est le but recherché dans cette dernière partie.

Ainsi, dans un premier temps, les explications portent sur les données manipulées : leur structure, leur type et leur stockage en mémoire. On retrouve alors la notion de *modules* ; celle-ci est complétée par les concepts de *registres*, de *variables globales* et de *macros*. On trouvera par exemple les instructions pour la création et l'inclusion de modules dans Sophemis.

Les solutions adoptées pour la réalisation des calculs (en calcul ou sur machines distantes) sont ensuite détaillées.

Tous les modules, registres, variables globales et macros sont listés, décrits et mutuellement référencés.

Etant donné qu'elle en décrit les caractéristiques, l'aide doit être mise à jour dès que les fonctionnalités de Sophemis sont modifiées.

L'aide de Sophemis compte environ 270 pages. Elle est accessible de l'application elle-même.

Autres caractéristiques