**Hackathon n° 3**

**Questionnaire sur un cas d’école parmi les applications de votre entité**

**BASILE et JEAN-PHILIPPE**

**Partie II - Spécifications techniques (après-midi)**

Pour les notions suivantes :

1. Etes-vous d’accord avec leurs définitions ci-dessous ?
2. Sinon, comment les redéfiniriez-vous ?
3. Sont-elles utilisées dans votre cas d’usage ?
4. Si oui, si cette notion devait être représentée par une classe, quels seraient ces attributs dans votre implémentation ? En d’autres termes, quelle serait la structure interne de l’objet correspondant ?

***Problem to solve*** (*cas à résoudre*) : Question d’intérêt à résoudre dans un domaine quelconque, par un raisonnement scientifique traduisible en équations. On parle aussi de problème.

* **Plutôt d’accord**
* ~~Plutôt pas d’accord~~

Compléments ou définition alternative : ……………………………………………………………………………………………………… ……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

Attributs :

* <nom> : <type> → <description>
* Typage mathématique des équations à résoudre, C.L., C.I.

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Case data****(données du problème) :* Informations « de haut niveau » définissant un *cas à résoudre*, utilisables par le solveur HF pour traduire le problème en équations. C’est la description métier du problème à résoudre, le plus souvent, le « métier » consiste en l’étude d’un phénomène physique. On peut donc dire que la notion de *données du problème* représente l’ensemble de la description *physique* du cas à résoudre.

* **Plutôt d’accord**
* ~~Plutôt pas d’accord~~

Compléments ou définition alternative :

… de *données du problème* représente l’ensemble de la description **numérique de la** *physique* du cas à résoudre.

**Les données sont la particularisation numérique des équations et de leurs paramètres. Par exemple Loi de Comportement = pb à résoudre ; paramètre de la LdC = données du problème**

Attributs :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Raw data*** (*données brutes*) : désigne la description mathématique des équations du problème à résoudre. On peut donc dire que la notion de *données brutes* recouvre tout ce qui est utile à un utilisateur « réducteur de modèles » (qui ne connaît pas nécessairement la physique) pour appliquer une méthode de réduction non intrusive.

* ~~Plutôt d’accord~~
* **Plutôt pas d’accord**

Compléments ou définition alternative :

**Très difficile d’imaginer cette notion pour un cas industriel.**

Attributs :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Solver*** */ resolution procedure :* Ensemble d’opérations permettant la traduction en équations algébriques du problème à partir d’informations « de plus haut niveau » (*case data* et *solver* *options*), puis sa résolution. Il prend la forme d’un logiciel.

En résumé :

PDE

Case data

Solver options

Raw data

Case solution

Solver

Solver

Traduction

Résolution

* **Plutôt d’accord avec la définition**
* **Plutôt pas d’accord avec le schéma**

Compléments ou définition alternative :

Pas d’accord avec le schéma car la première flèche n’est réaliste que pour des cas simples ( pas non-linéaire, géométrie simple)

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***High-fidelity solver*** (*solveur HF*): Solveur développé et utilisé par les spécialistes du domaine considéré, préexistant à la tâche de réduction de modèles.

* **Plutôt d’accord**
* ~~Plutôt pas d’accord~~

Compléments ou définition alternative : ……………………………………………………………………………………………………… ……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

Propriétés:

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Solver options****(options du solveur)* : pour un problème métier donné, représente les variantes disponibles dans le solveur pour le résoudre. On peut distinguer : (i) options de discrétisation en espace ou en temps, (ii) options de résolution algébriques (traitement des conditions de Dirichlet, solveur linéaire...), (iii) options d’exécution (version du code, serveur etc)

* **Plutôt d’accord avec le terme**
* **Plutôt pas d’accord avec la définition**

Compléments ou définition alternative :

**Autre définition : toutes les données nécessaires au solveur qui ne sont pas les paramètres par rapport auquels on veut ou on pourrait vouloir réduire**

Attributs :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Fixed data*** (*données fixes*) : données considérées pour la question d’intérêt comme invariantes, ~~soient parce que suffisamment bien connues soit parce que peu influentes sur les résultats d’intérêt~~.

Compléments ou définition alternative :

**L’important est que les données soient invariantes, pas la raison pour laquelle elles le sont.**

Attributs / propriétés:

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Variable data*** or ***parameters*** (*données variables*): Données considérées pour la question d’intérêt comme variables. On parle aussi de paramètres. **Une première information permet de savoir où elles s’insèrent dans la « mise en donnée » du problème continu (*données du problème*) ou algébrique (*données brutes*). D’autres informations peuvent décrire la nature physique de ce paramètre, en vue de les afficher dans le modèle réduit.**

Compléments ou définition alternative :

**La partie en gras souligné ne doit pas faire partie de la définition, mais c’est des propriétés de la quantité définie.**

Attributs / propriétés:

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***PDE***(*EDP*): indique la nature des équations continues correspondant au cas à résoudre. Exemple : Navier-Stokes…

Compléments ou définition alternative :

C’est une définition qui devrait venir du dictionnaire et pas de nous. Et il manque les EDO.

Attributs / propriétés:

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Discrete support*** (*support discret*): Hiérarchie d’entités de (points, arêtes, faces, volumes...) sur lesquelles s’appuient à la fois la définition du problème et sa traduction en équations. Il doit prévoir la possibilité de tagguer des groupes d’entités. Des *domaines de définition* (espace, temps, paramètres ou une combinaison des précédents) peuvent être générés par produit cartésien de supports discrets.

* **Plutôt d’accord**
* ~~Plutôt pas d’accord~~

Compléments ou définition alternative :

**Attention : rajouter ou pas le lien avec l’espace d’approximation dont on a besoin en state variable.**

Attributs / propriétés :

* dimension : int → dimension N of the underlying space
* type : string → into “space”, “time” or “parameter”
* **Nature géométrique (et pas maillage) des éléments supports**

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***DOF support****(support de DDL)* : association entre une entité géométrique ou une combinaison d’entités géométriques et une inconnue (1 ddl) du problème. Dans la plupart des cas, ce sont des nœuds (éléments finis) ou des points (volumes finis), mais pas nécessairement.

* **Plutôt d’accord**
* ~~Plutôt pas d’accord~~

Compléments ou définition alternative : ……………………………………………………………………………………………………… ……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Numbering system*** (numérotation) : agrégation de *supports de DDL* de toutes les inconnues du problème

* **Plutôt d’accord**
* ~~Plutôt pas d’accord~~

Compléments ou définition alternative : **On ne sait pas trop a quoi cela peut servir ici.**

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Physical quantity*** (quantité physique ou grandeur) : nature d’un résultat, elle est associée à une seule unité au sens SI.

* **Plutôt d’accord**
* ~~Plutôt pas d’accord~~

Compléments ou définition alternative :

**Unités = SI + sans unité (par exemple déformation)**

**Attention : tensor-order n’est pas adapté à une quantité qui n’est pas un champ : énergie, débit, ...**

Attributs / propriétés :

* label : string → describes the quantity
* unit : string → unit in which resultas are expressed
* tensor-order : int → 0 for scalars fields, 1 for vector, 2 for matrices

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Field****(champ)*: valeurs portées par un *support discret*, en général sur les *nœuds* ou les *points*. Correspond à une seule *quantité physique* (éventuellement tensorielle). Peut être d’origine expérimentale ou un post-traitement, ~~rarement~~ **ou** la sortie brute de la simulation. Tous les *points* sur lesquels le champ porte une valeur (attention au cas données manquantes) ont le même nombre de composantes.

Définition alternative plus générale : quelque chose qu’on peut évaluer à n’importe quelle position à l’intérieur du domaine .

* **Plutôt d’accord**
* ~~Plutôt pas d’accord~~

Compléments ou définition alternative :

**Attention au temporel comme support discret, en particulier dans le cas où l’on change le maillage en temps.**

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Vector of unknowns*** *(vecteur d’inconnues)* : C’est le vecteur d’état (vecteur des variables d’état discrètes) que le problème doit déterminer. C’est la sortie primale de la modélisation. Il peut mélanger des inconnues de différentes unités, autrement dit mélanger différentes *quantités physiques* : déplacement, pression, multiplicateur de Lagrange… Une *inconnue* peut être associée à un *support de ddl*. Ce n’est pas systématique (cf certains multiplicateur de Lagrange introduits de façon purement algébrique). Un *vecteur d’inconnues* est donc également associé à un *support discret*, mais de façon plus indirecte.

Un *vecteur d’inconnues* agrège plusieurs *variables d’état* (déplacement, ~~multiplicateurs~~...), dont certaines peuvent ne pas avoir de représentation spatiale.

* **Plutôt d’accord**
* ~~Plutôt pas d’accord~~

Compléments ou définition alternative :

**Dire plutot que le vecteur d’inconnues = {Vecteur des variables d’état } + {Autres : Lagrange, … }**

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***State variable*** *(variable d’état)* : vecteur correspondant à une seule quantité physique et pouvant se représenter comme un champ via un *espace d’approximation*, lequel s’appuie sur le *support discret*. Exemples d’espaces d’approximation : espace polynomiaux par éléments associées aux éléments finis de Lagrange, de Hermite… A l’inverse, un champ se représente comme variable ~~d’état~~ par une méthode d’*interpolation* visant à définir un vecteur à partir d’opérations (éventuellement intégrales) sur les valeurs du champ.

* **Plutôt d’accord**
* ~~Plutôt pas d’accord~~

Compléments ou définition alternative : ……………………………………………………………………………………………………… ……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

Attributs / propriétés :

* <nom> : <type> → <description>
* **lien vers espace d’approximation (et pas vers le support discret directement)**

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Model*** *(modèle)* : fonction du domaine (paramètres x temps), qui donne en retour ~~un champ. Tous~~ les champs produits au final doivent se rapporter à un unique maillage « de référence » . La transformation avec d’éventuels maillages intermédiaires est masquée à l’intérieur de la fonction en quelque sorte

* ~~Plutôt d’accord~~
* **Plutôt pas d’accord**

Compléments ou définition alternative :

**Non car un champ a été défini comme portant une seule quantité physique => la sortie devrait être un snapshot.**

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Case solution*** or *snapshot (solution cas)*: Valeur d’une quantité mathématique pour laquelle le système d’équations discret est satisfait, pour une valeur de paramètre donnée, temps inclus. Une solution se rapporte, via son cas et éventuellement sa valeur de paramètre, à un support discret. Une solution contient une clé (*indexation*) permettant de repérer la valeur de paramètre à laquelle elle se rapporte, et une valeur, qui peut être un *champ* ou un *vecteur d’inconnues*.

* ~~Plutôt d’accord~~
* **Plutôt pas d’accord**

Compléments ou définition alternative :

Valeur **d’un ensemble de quantités solution du** système d’équations discret est satisfait, pour une valeur de paramètre donnée, temps inclus. Une solution se rapporte, via son cas et éventuellement sa valeur de paramètre, à un **ou des** support discret. Une solution contient une clé (*indexation*) permettant de repérer la valeur de paramètre à laquelle elle se rapporte, et une valeur, qui **est un vecteur solution**

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Collection of solutions*** (*collection de solutions*): ensemble de solutions, chacune étant qualifiée par sa valeur des données variables (temps inclus), parmi les valeurs effectivement balayées. Souvent utilisée par les algorithmes sous forme d’une matrice Q, dite matrice des snapshots.

* **Plutôt d’accord**
* ~~Plutôt pas d’accord~~

Compléments ou définition alternative : ……………………………………………………………………………………………………… ……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Indexing support*** (*support d’indexation*) : domaine de définition dans lequel les paramètres sont autorisés à prendre leurs valeurs ou ensemble mathématique discret permettant d’indexer de façon univoque des champs. Ce *domaine de définition* est défini par un produit cartésien de supports discrets.

* **Plutôt d’accord**
* ~~Plutôt pas d’accord~~

Compléments ou définition alternative :

**Attention : pas forcément un produit cartésien. Cas des contraintes entre paramètres, ou s’il y a des trous.**

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Indexing*** *(indexation)* : Valeur de paramètres pour laquelle une solution est effectivement disponible.

* **Plutôt d’accord**
* ~~Plutôt pas d’accord~~

Compléments ou définition alternative :

Nom proposé : **Indexing Value**

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Reduced domain*** *(domaine réduit)* : filtre sur un *support discret*, permettant de sélectionner des entités géométriques d’intérêt pour la construction d’opérateurs compressés. Exemple: points d’intégrations empiriques (*nuage de points*), éléments finis d’une hyper-réduction (*sous-domaine de calcul réduit*)

* **Plutôt d’accord**
* ~~Plutôt pas d’accord~~

Compléments ou définition alternative :

filtre : slection, ui entraine que le reduced domain est un nouveau support discret

discret : doit pouvoir être indépendant des support Haute Fid.

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Building operator*** */ resolution data (opérateur de construction)* : toute donnée de nature non géométrique ou toute procédure utile à la traduction en équations discrètes du problème. Ce sont des “prises” ajustables complétant (le plus souvent) ou modifiant (plus rarement) l’ensemble d’opérations du solveur.

Exemples : poids du schéma de quadrature, matrices et vecteurs précalculés, routine de comportement, fonctions de calcul des quantités duales, procédures de reconstruction du gradient, procédures d’assemblage particulières etc.

* Plutôt d’accord
* Plutôt pas d’accord

Compléments ou définition alternative :

**Définition compliquée ou notion trop compliquée !?**

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Compression of data*** (*compression des données*) : opérations de la phase offline permettant la définition d’un petit nombre de fonctions de l’espace, dites « fonctions de bases », servant à la définition d’un espace réduit de recherche de la solution. Elle produit une collection de solutions, laquelle est tagguée comme « base réduite ».

* Plutôt d’accord
* Plutôt pas d’accord

Compléments ou définition alternative :

**collection de solution : attention ce n’est pas une matrice de snapshots**

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Compression of operators*** (*compression des opérateurs*) : opérations de la phase ~~offline~~ ayant pour but la production des opérateurs de construction d’un modèle réduit ou la diminution de leur complexité algorithmique (dans le but d’accélérer la phase online). Il produit des opérateurs de construction réduits.

Exemples: il peut s’agir d’une méthode de projection d’opérateurs de construction précalculés sur une base réduite, ou de méthodes faisant appel à un domaine réduit. Les premiers prennent en entrée des opérateurs complets, les seconds des opérateurs réduits dans le but d’effectuer une deuxième opération de réduction.

* **Plutôt d’accord**
* ~~Plutôt pas d’accord~~

Compléments ou définition alternative :

**offline et/ou online**

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Reduced case to solve*** (*cas réduit à résoudre*): ensemble des informations nécessaires à la réalisation de la phase online. On trouve donc des données du problème, les mêmes que celles qui ont servies à la définition du problème complet associé, mis à part que les données fixes ne sont que consultables, seules les données variables sont modifiables dans la plage spécifiée. On trouve également un solveur réduit assorti des opérateurs de constructions réduits qui le complètent.

* **Plutôt d’accord**
* ~~Plutôt pas d’accord~~

Compléments ou définition alternative : ……………………………………………………………………………………………………… ……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Hyper-reduction*** : méthode de compression définie par Ryckelynck et al [Ryckelyck09], consistant à résoudre un problème aux éléments finis sur un sous-maillage du maillage existant.

* **Plutôt d’accord**
* ~~Plutôt pas d’accord~~

Compléments ou définition alternative : ……………………………………………………………………………………………………… ……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Weights****(poids)*: Poids d’un schéma de quadrature utilisé pour approximer une intégrale.

* **Plutôt d’accord**
* ~~Plutôt pas d’accord~~

Compléments ou définition alternative :

**Changer le nom, pour Quadrature Weights. Car la masse existe aussi.**

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Reduced-order basis vector*** *(Vecteur de base d’ordre réduit)* : Champ ou vecteur d’inconnue issu d’un algorithme de sélection (méthode base réduite) ou de compression (SVD) appliqué à une *collection de solutions*.

* Plutôt d’accord
* Plutôt pas d’accord

Compléments ou définition alternative :

**Manque de précision (champ ou vecteur d’inconnue)**

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Reduced-order basis*** *(base d’ordre réduit)* : Base de l’espace réduit sur lequel on projette les équations du problème. Collection de vecteur de base d’ordre réduit produite par une compression des données et utilisée par une compression des opérateurs.

* Plutôt d’accord
* Plutôt pas d’accord

Compléments ou définition alternative : ……………………………………………………………………………………………………… ……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Reduced basis*** *(base réduite)* : Désigne les méthodes de production d’une base d’ordre réduit reposant sur de sélections gloutonnes de vecteurs pour construire la base d’ordre réduit. Exemples: celles proposées par Maday, Prudhomme, Patera

* **Plutôt d’accord**
* ~~Plutôt pas d’accord~~

Compléments ou définition alternative :

**Ajouter les RB Method**

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

***Autre notion***:

Définition : ……………………………………………………………………………………………………… ……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

…………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

***Autre notion***:

Définition : ……………………………………………………………………………………………………… ……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

…………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

***Autre notion***:

Définition : ……………………………………………………………………………………………………… ……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

…………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

***Autre notion***:

Définition : ……………………………………………………………………………………………………… ……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

Attributs / propriétés :

* <nom> : <type> → <description>

Lien avec les autres objets : ……………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

1. Environnement logiciel / partie exécutable

Quel logiciel avez-vous utilisé dans la mise en œuvre du cas ?

**Z-set/Zébulon**

Licence : Commercial (gratuite pour académique)

Type d’installation :

* **Exécutable pré-compilé**
* ~~Installeur~~
* **Construction depuis les sources**
* ~~Paquet Debian~~
* ~~Module Linux~~
* ~~Autre~~

Lien vers une procédure d’installation : ………………………………………………………………………………..

Dépendances logicielles (« pré-requis ») les plus notables : **MUMPS, Lapack**

Quel est le système d’exploitation utilisé (ou cible) pour la réalisation des simulations complètes ?

* **Linux.** Distribution : Centos ou Debian . Version : …………………………………
* ~~MAC OS / OS X. Version : …………………………………~~
* ~~Windows. Version : ……………………..~~
* ~~Linux embedded. Version : ……………………..~~
* ~~Autre. Précisez : ………………………….~~

Quel est le système d’exploitation utilisé (ou cible) pour la production du modèle réduit ?

* **Linux**. Distribution : ……………………………. Version : …………………………………
* ~~MAC OS / OS X. Version : …………………………………~~
* ~~Windows. Version : ……………………..~~
* ~~Linux embedded. Version : ……………………..~~
* ~~Autre. Précisez : ………………………….~~

Quel est le système d’exploitation utilisé (ou cible) pour l’exploitation du modèle réduit ?

* **Linux. Distribution : ……………………………. Version : …………………………………**
* ~~MAC OS / OS X. Version : …………………………………~~
* ~~Windows. Version : ……………………..~~
* ~~Linux embedded. Version : ……………………..~~
* ~~Autre. Précisez : ………………………….~~

Quel est la machine utilisée (ou cible) pour la réalisation des simulations complètes ?

* **Cluster. Mode d’accès**
  + **SSH**
  + ~~Cloud, lancement d’une instance de VM sur le cluster~~
  + ~~Cloud, service spécialisé~~
* **PC**
* ~~Support léger. Précisez : ……………………..~~
* ~~Autre. Précisez : ……………………..~~

Quel est la machine utilisée (ou cible) pour la production du modèle réduit ?

* **Cluster. Mode d’accès**
  + **SSH**
  + ~~Cloud, lancement d’une instance de VM sur le cluster~~
  + ~~Cloud, service spécialisé~~
* **PC**
* ~~Support léger. Précisez : ……………………..~~
* ~~Autre. Précisez : ……………………..~~

Quel est la machine utilisée (ou cible) pour l’exploitation du modèle réduit ?

* ~~Cluster. Mode d’accès~~
  + ~~SSH~~
  + ~~Cloud, lancement d’une instance de VM sur le cluster~~
  + ~~Cloud, service spécialisé~~
* **PC**
* ~~Support léger. Précisez : ……………………..~~
* ~~Autre. Précisez : ……………………..~~

1. Parmi le code source écrit dans le cadre de ce cas par votre entité, et candidat pour être un futur module adapté dans Mordicus :

Quel est le langage de programmation ?

* **C++**
* ~~Python~~
* ~~Matlab~~
* ~~Scilab~~
* ~~Fortran~~
* Autre. Précisez : …………………………….

Se construit-il comme une bibliothèque ou un exécutable ?

* ~~exécutable~~
* **bibliothèque**

Quel est le moteur de construction ? **SCONS**

Lister les dépendances les plus notables pour la construction :

* Lapack

Décrire brièvement l’environnement de développement :………………………………………………………………………..

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

……………………………………………………………………………………………………………………………………………….

1. Inventorier les opérations génériques utilisées dans votre cas d’usage :

* SVD
* Produit Matrice, vecteurs
* Produit Matrice Matrice
* Transposition de matrice
* Résolution système linéaire sparse

1. Voici une proposition d’architecture d’ensemble de la future bibliothèque Mordicus. Modifiez-la pour y faire apparaître les éléments manquants qui vous semblent essentiels. Dans quel composant classeriez-vous les opérations précédemment listées ?

