De la modélisation littérale à la simulation numérique certifiée

Habilitation à Diriger des Recherches
de
Yves Papegay

Equipe Projet Inria COPRIN

Ecole Doctorale
Sciences et Technologie de l'Information et de la Communication
Université de Nice Sophia Antipolis

22 Juin 2012

Dynamique dans la chronologie

- De . . . littéral[e] à . . . numérique . . .

1992 Thèse avec Jacques Morgenstern
"Outils formels pour la modélisation"

1993 Post-doc avec Hirohisa Hirukawa
ElectroTechnical Lab. (Tsukuba)

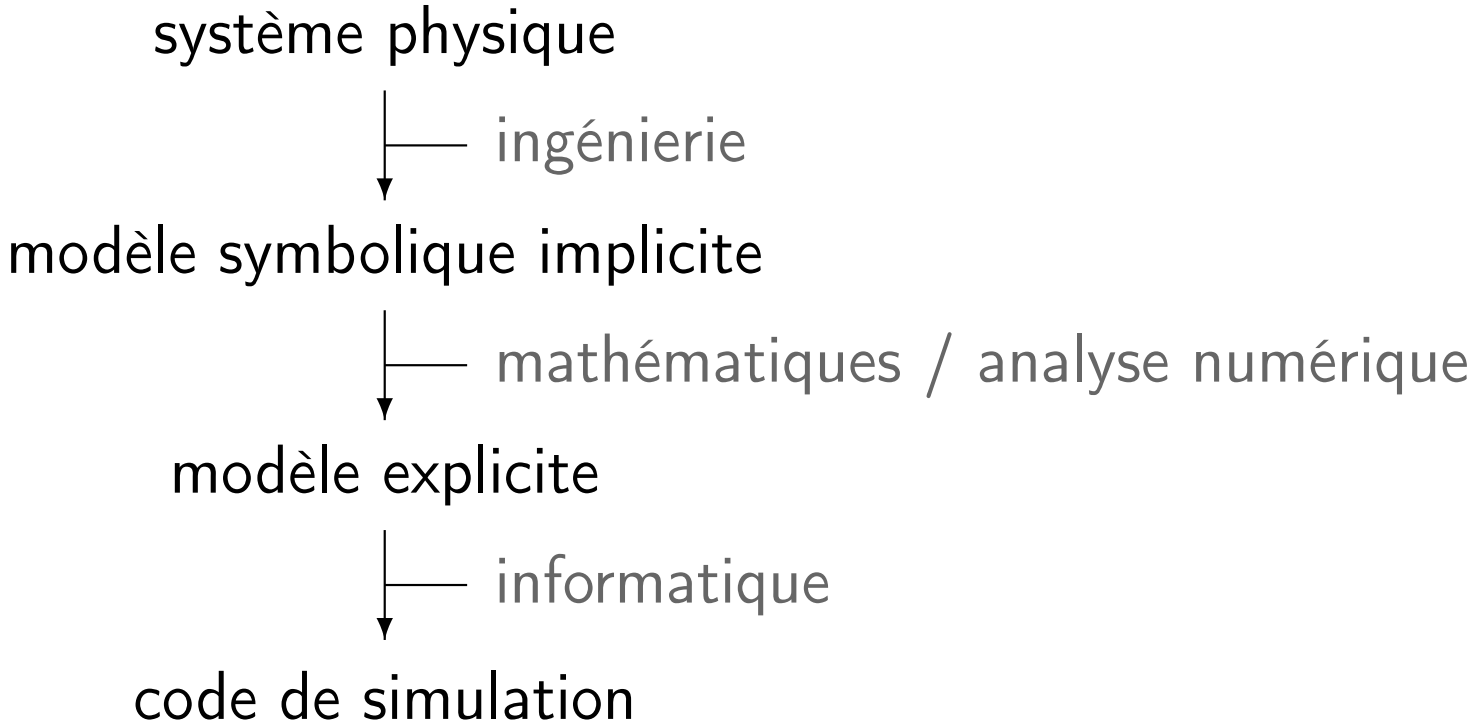
1994 Chargé de Recherche Inria - projet SAFIR
Systèmes Algébriques et Formels pour l'Industrie et la Recherche

1999 projet SAGA
Systèmes Algébriques, Géométrie et Applications

2002 E.P.I. COPRIN
Contraintes, optimisation et résolution par intervalles

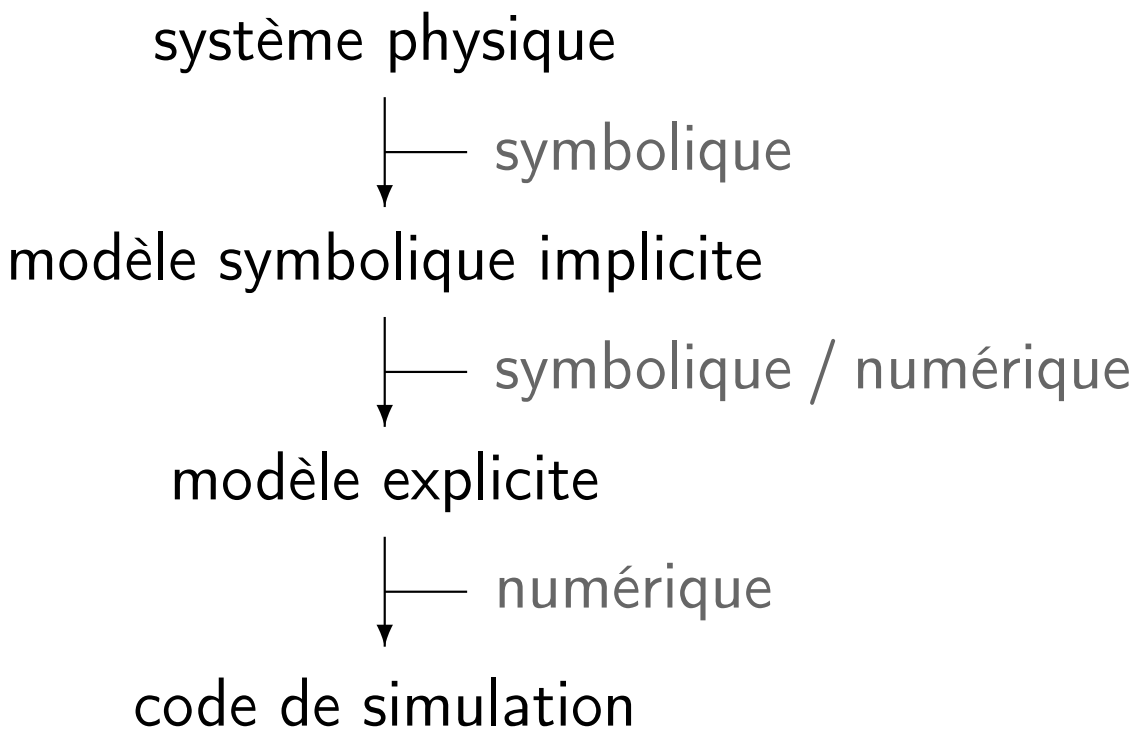
Dynamique dans les méthodes

- De la modélisation . . . à la simulation . . .



Dynamique dans les méthodes

- De . . . littéral[e] à . . . numérique . . .



Plan

Outline

Perspective thématique

- processus de modélisation/simulation

système physique

↓

modèle symbolique implicite

↓

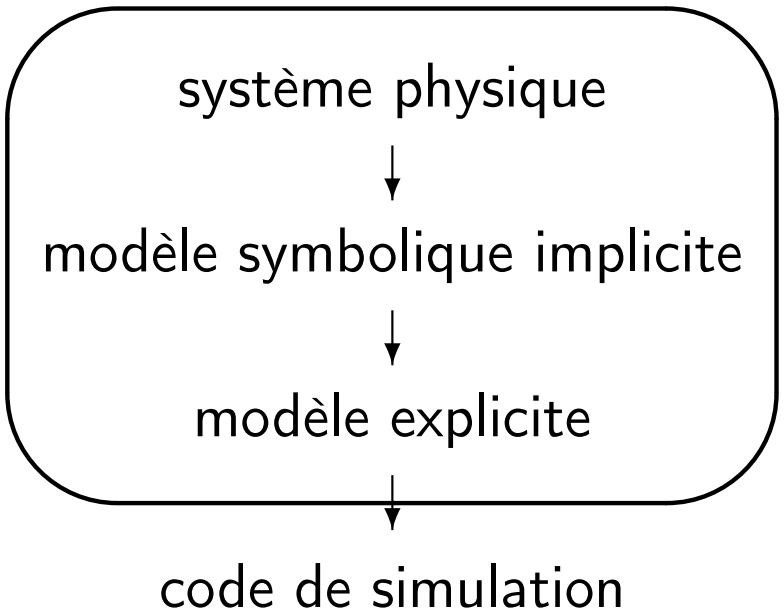
modèle explicite

↓

code de simulation

Perspective thématique

- processus de modélisation/simulation



- formalisation du processus
- langages de description de systèmes
- gestion de l'orientation des modèles
- modèle calculatoire

Formalisation du processus

Concepts

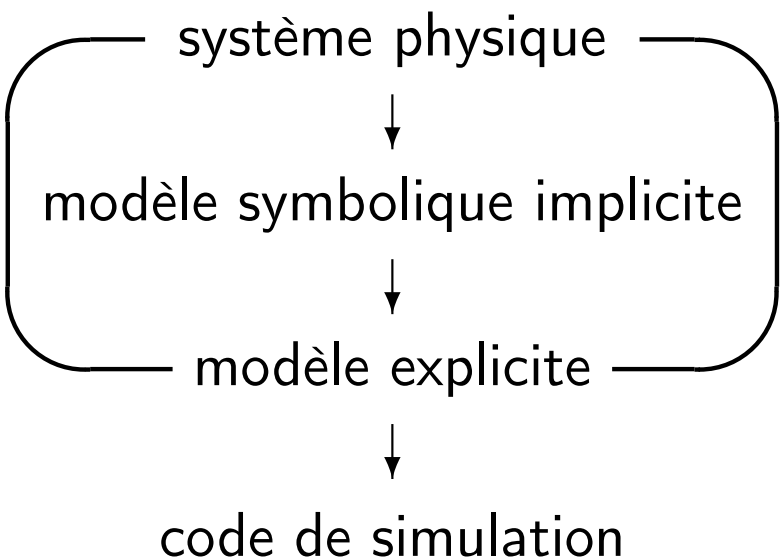
- variable
- modèle formel, modèle orienté, modèle calculatoire

Projets

- dynamique des systèmes polyarticulés (1992)
- LACAPIO – conception et évaluation des performances d'instruments optiques (1995)
- YPAMA – un assistant à la modélisation aerodynamique (2002)

Perspective thématique

- processus de modélisation/simulation



- formalisation du processus
- langages de description de systèmes
- gestion de l'orientation des modèles
- modèle calculatoire

Langages de description des systèmes

LSD - Langage Symbolique de Description

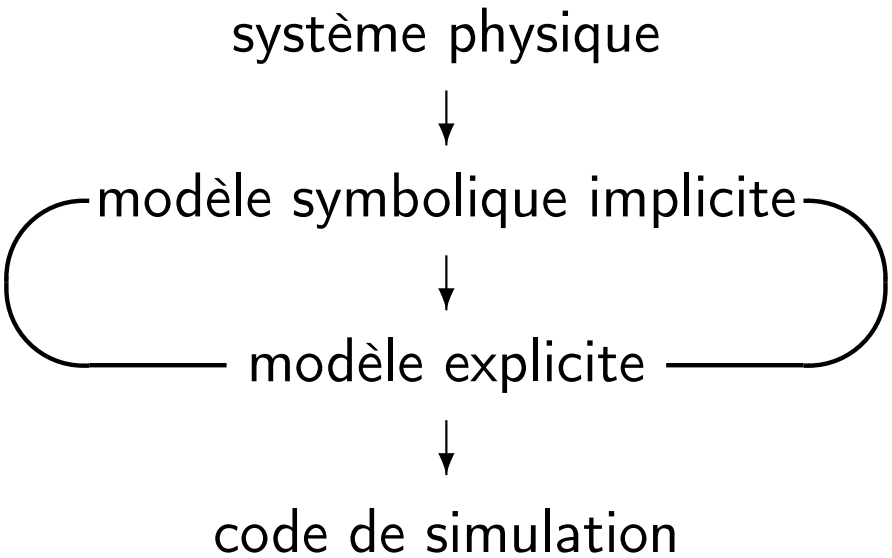
- dynamique des systèmes polyarticulés (1992)
- description des corps et des liaisons
- catalogue de liaisons et liaison générique
- description de la structure (graphe)

LM2 - Langage de Modélisation en *Mathematica*

- MOSELA (2007)
- déclarations des variables
- déclarations des relations entre variables
- appels d'instances de sous-modèles

Perspective thématique

- processus de modélisation/simulation



- formalisation du processus
- langages de description de systèmes
- gestion de l'orientation des modèles
- modèle calculatoire

Orientation des modèles

Concepts

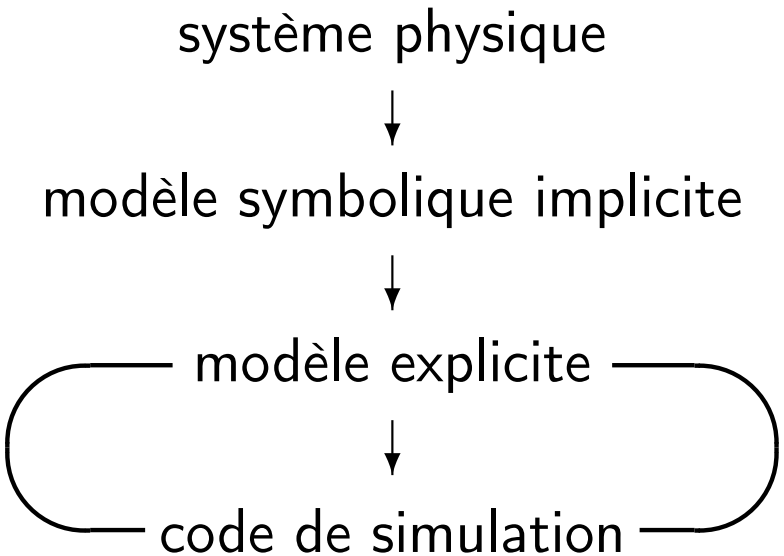
- utiliser le même modèle pour la conception et la simulation
- $F - m.\gamma = 0$
- $F = m.\gamma$, $\gamma = \frac{F}{m}$ ou $m = \frac{F}{\gamma}$
- explicitation formelle, numérique, impossible, ...

Projets

- LACAPIO – conception et évaluation des performances d'instruments optiques (1995)
- YPAMA – un assistant à la modélisation aerodynamique (2003)

Perspective thématique

- processus de modélisation/simulation



- formalisation du processus
- langages de description de systèmes
- gestion de l'orientation des modèles
- modèle calculatoire

Modèle Calculatoire

Concepts

- programme de calcul
- ordonnancement

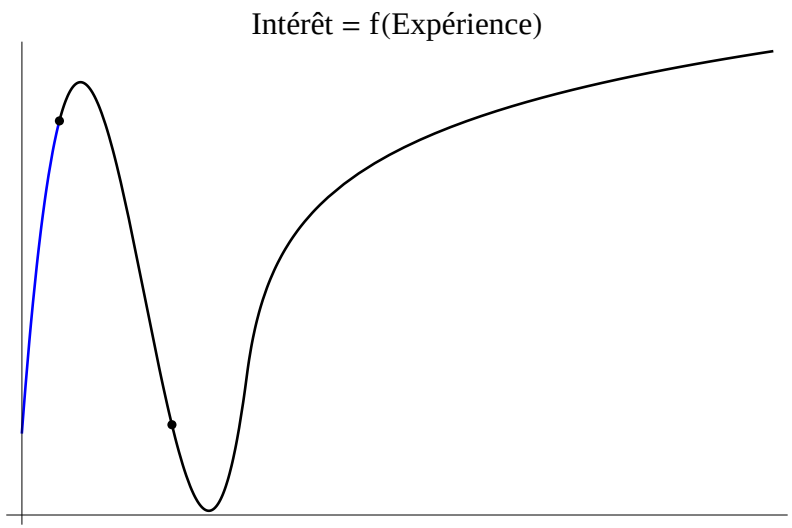
Projets

- MOSELA – environnement de modélisation et de simulation (2003)

Perspective méthodologique

- méthodes et outils de calcul formel

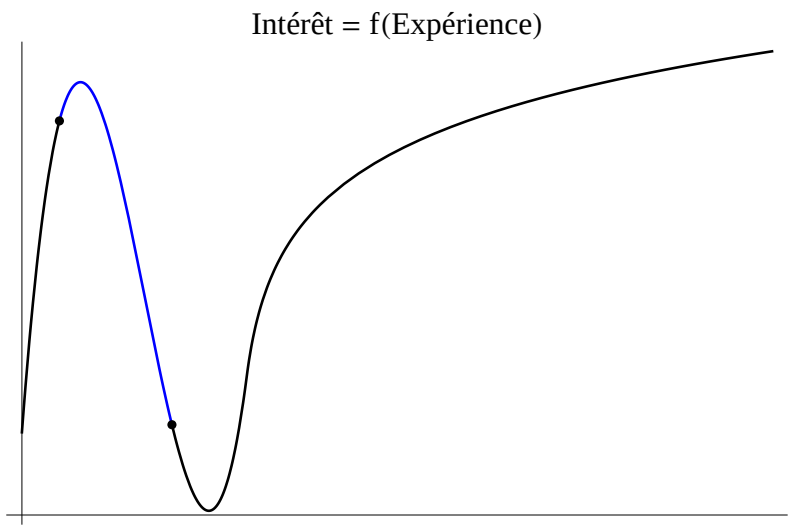
- calculette symbolique
- puissance de résolution
  vs. taille des expressions
- extension
  - objets symboliques
  - nouveaux algorithmes
  - interfaces
- algorithmes
  symboliques/numériques



Perspective méthodologique

- méthodes et outils de calcul formel

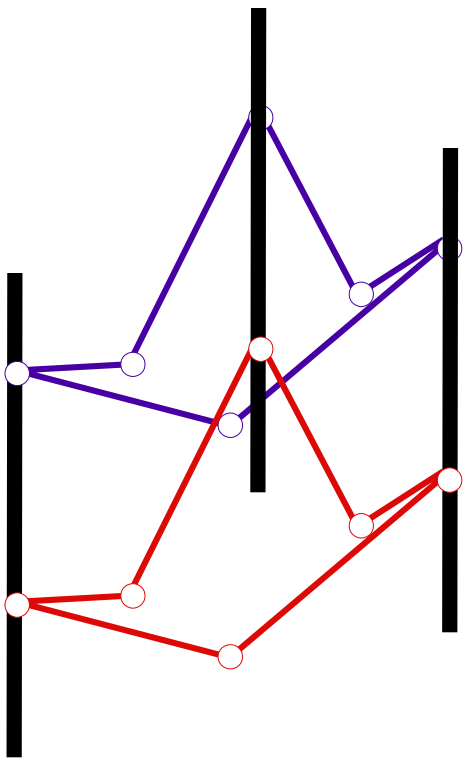
- calculette symbolique
- puissance de résolution
  vs. taille des expressions
- extension
  - objets symboliques
  - nouveaux algorithmes
  - interfaces
- algorithmes
  symboliques/numériques

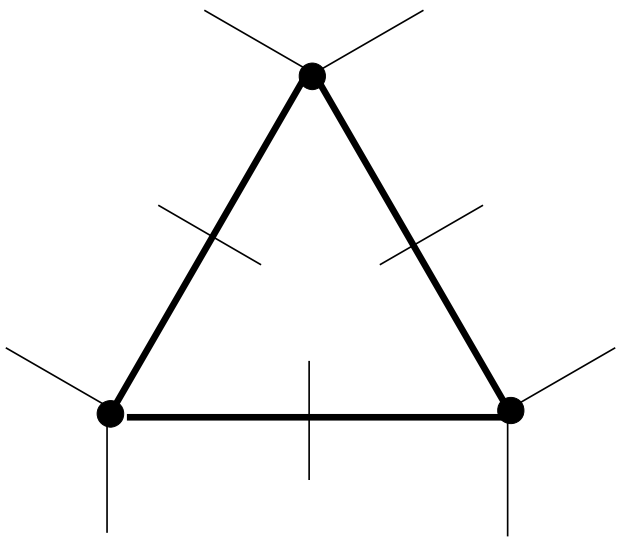


Puissance de résolution

Calcul du nombre de degrés de liberté

- résolution du module de syzygies

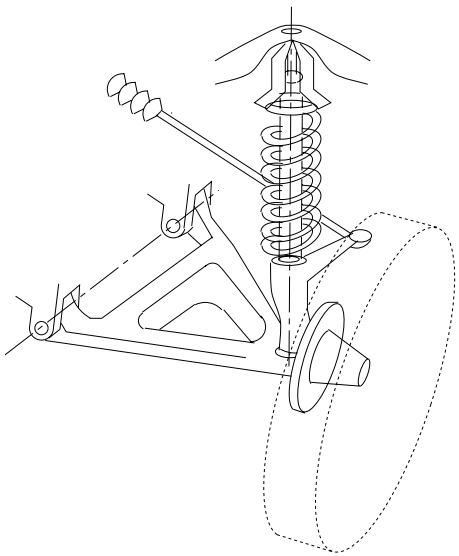


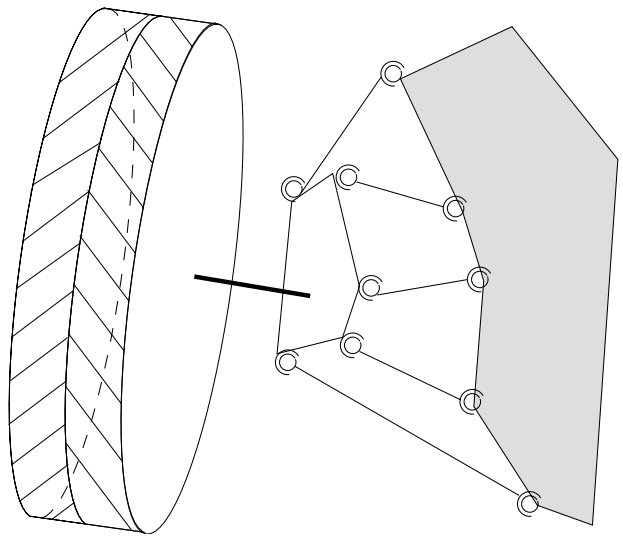


Taille des expressions

Etude de suspensions automobiles (2002)

- Equations de distances

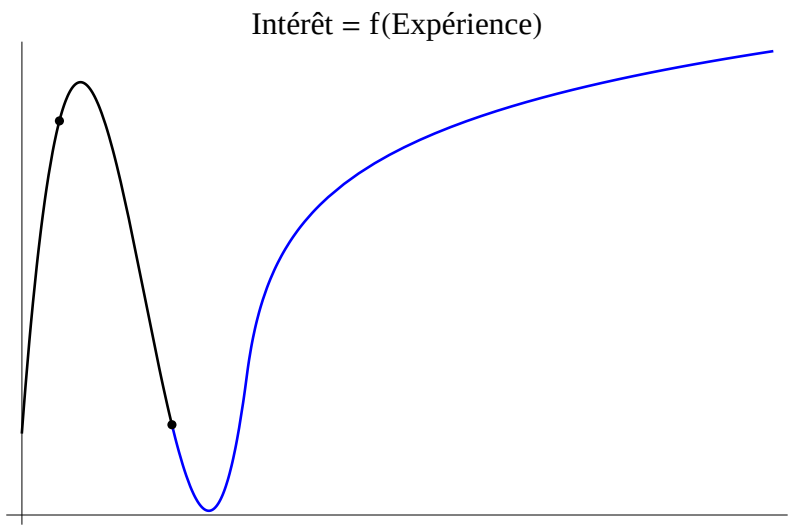




Perspective méthodologique

- méthodes et outils de calcul formel

- calculette symbolique
- puissance de résolution
  vs. taille des expressions
- extension
  - objets symboliques
  - nouveaux algorithmes
  - interfaces
- algorithmes
  symboliques/numériques



Objets symboliques

Quaternions (1993-1995)

- représentations des quaternions
  - symbole
  - scalaire et vecteur
  - 4 coordonnées
- règles de simplification
- manipulation symboliques

SYMBOLICC (2006)

- Sémantique du langage C

Nouveaux algorithmes

UNCERTAINTIES (2008)

- analyse par intervalles
- évaluation, bisection, résolution, optimisation

Calibration de robots parallèles (2004)

- résolution de systèmes surcontraints
- choix de positions de mesure

Algorithmes symboliques/numériques

- silhouette d'un ensemble semi-algébrique (2000)
- positionnement de robots en présence d'incertitudes (2007)

Interface

OPENMATH (2000)

- implémentation du protocole
- content dictionnary

ALIAS (2005)

- évaluateurs, solveurs

Axe théorique

- effectivité et efficacité
- applications industrielles - transfert technologique

- réalisation des calculs
- exploitation des résultats
  - lisibilité des résultats
  - évaluation numérique
- interface utilisateur
- qualité du code

- simulation d'un transistor MOS (1997)
- étude de suspensions automobiles (2002)
- édition de modèles aérodynamiques (2003)
- implémentation de jeux de plateaux (2006)

Outline

Génération de C

Motivations

- efficacité et performances
- intégration avec du code existant

Concepts

- traduction d'expression
- traduction de procédures
  - calculs
  - algorithmes
- instanciation de *templates*
- génération de code C

Traduction d'expression

In[8]:=
```
Integrate[g = Sin[t]^6, {t, 0, x}]
% // CForm
```

Out[8]=
$$\frac{1}{192}\,(60\,x - 45\,\mathrm{Sin}[2\,x] + 9\,\mathrm{Sin}[4\,x] - \mathrm{Sin}[6\,x])$$

Out[9]//CForm=
```
(60*x - 45*Sin(2*x) + 9*Sin(4*x)
- Sin(6*x))/192.
```

```
#include "mdefs.h"
double f(double x)
{
double y;
y = ;
return(y) ;
}
```

Instanciation de *template*

In[13]:=
```
Import["templ.mc", "Text"]
```

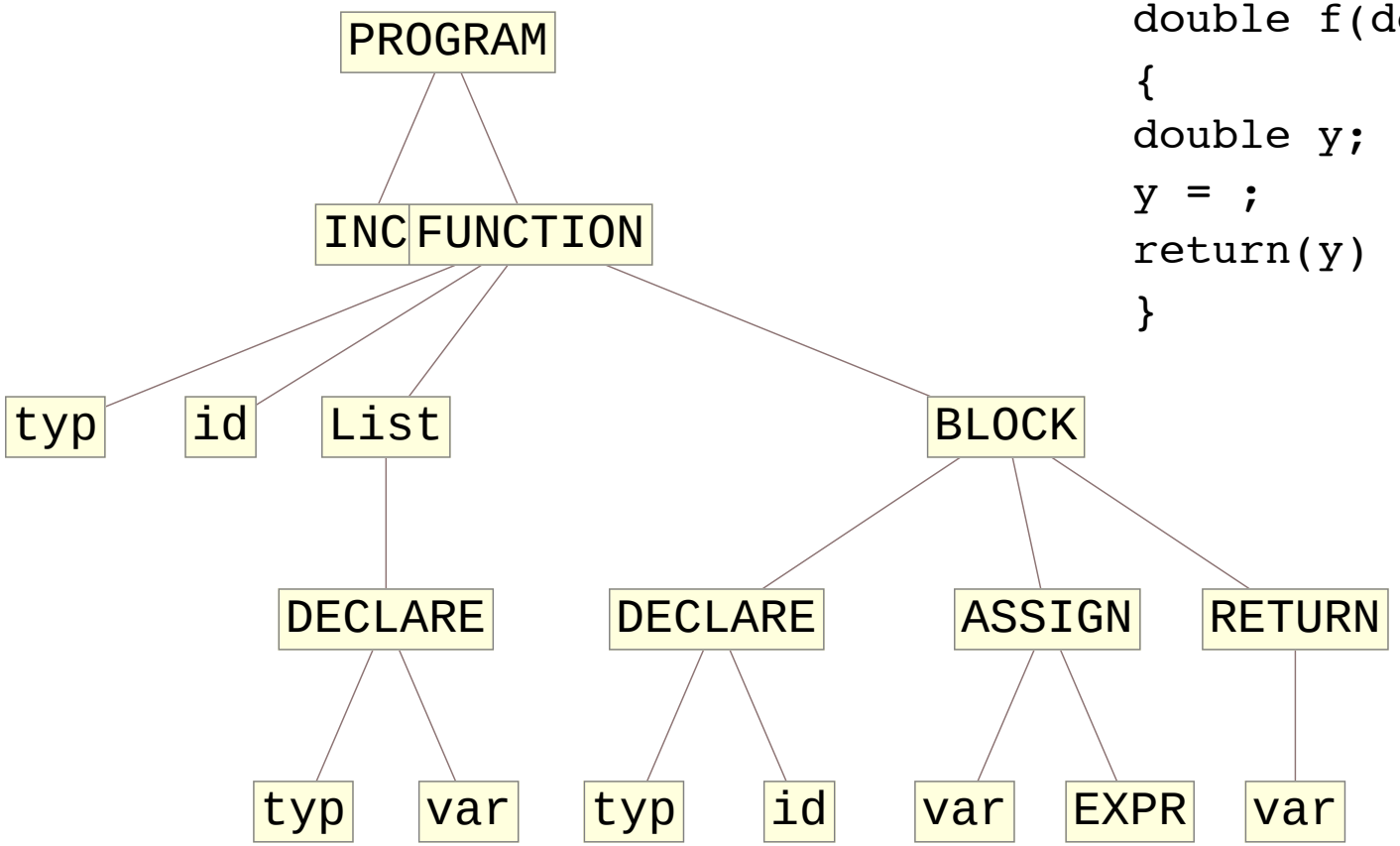
Out[13]=
```
#include "mdefs.h"
double f<*i*>(double x)
{
double y;
 y = <*
  Integrate[g,{t,0,x}] *>;
return(y) ;
}
```

In[16]:=
```
Splice["templ.mc",
 "integsplice.c"]
Import["integsplice.c", "Text"]
```

Out[16]=
```
templ.mc
```

Out[17]=
```
#include "mdefs.h"
double fi(double x)
{
double y;
 y = (60*x - 45*Sin(2*x) +
  9*Sin(4*x) - Sin(6*x))/192.;
return(y) ;
}
```

Représentation du langage




```
#include "mdefs.h"
double f(double x)
{
double y;
y = ;
return(y) ;
}
```

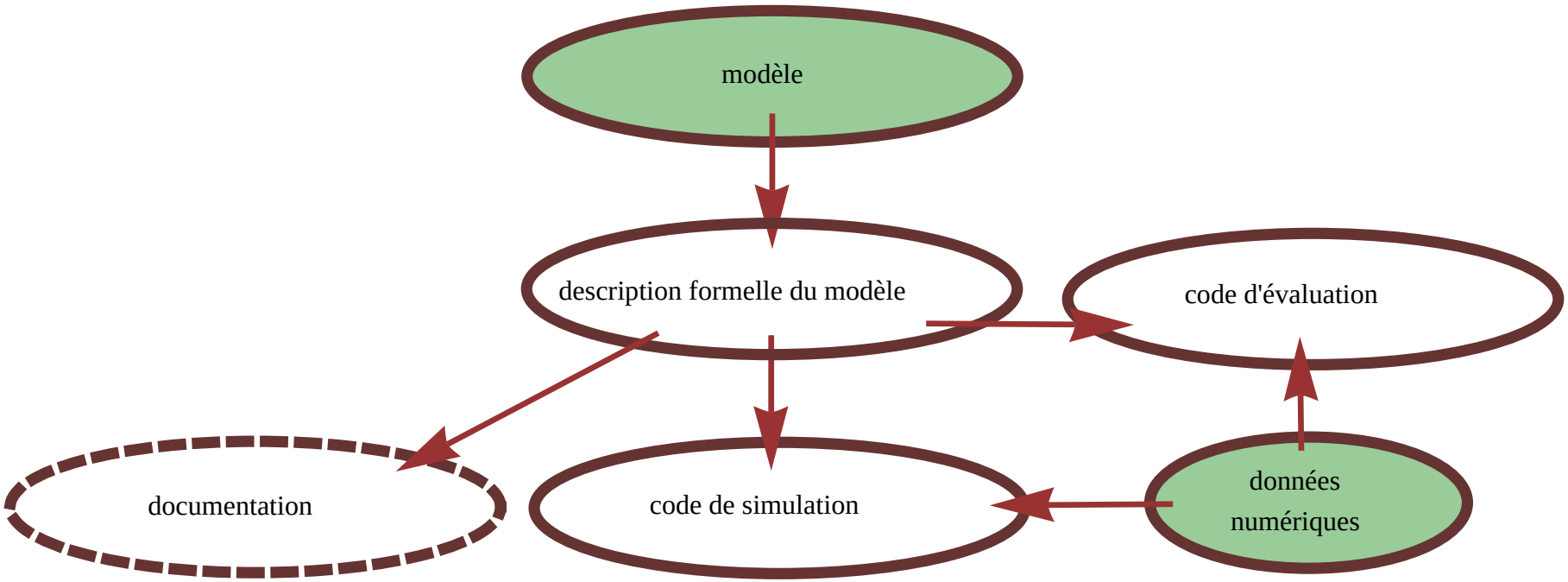
SymbolicC

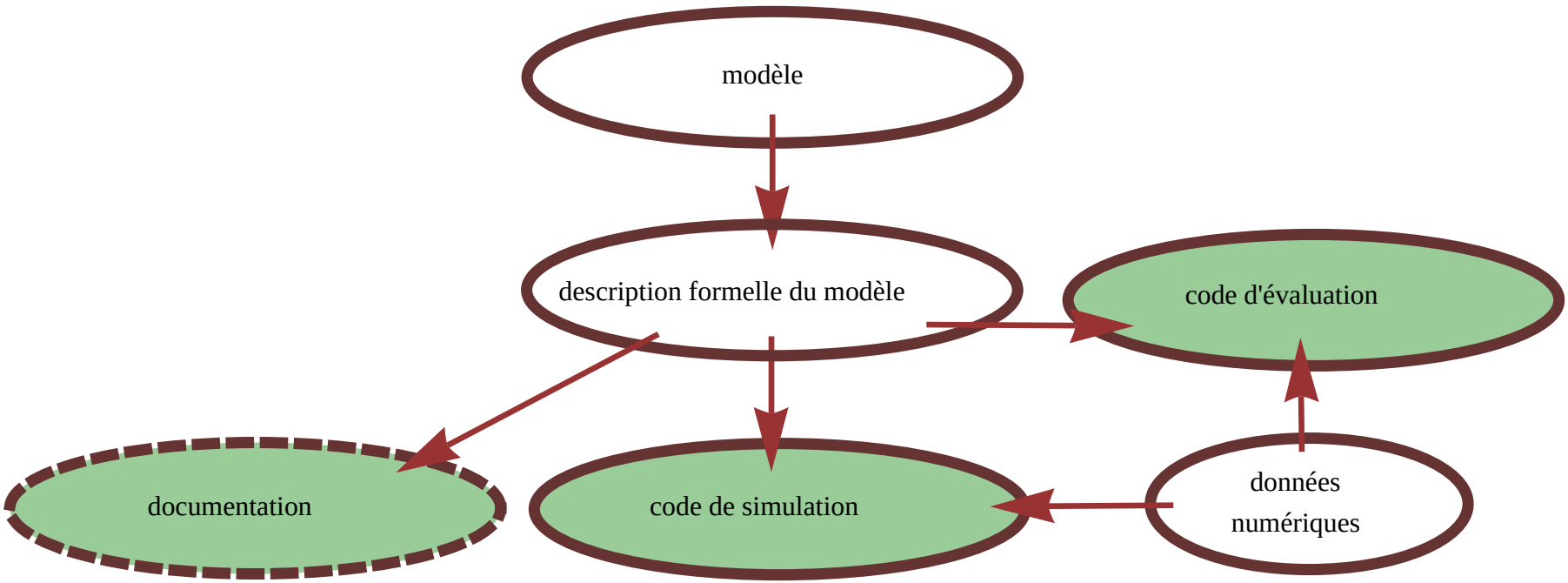
In[41]:=
```
p = ProgramC[
    IncludeC["mdefs.h"],
    FunctionC[double, "f",
     {DeclareC[double, x]},
     BlockC[{
       DeclareC[double, y],
       AssignC[y,
        Integrate[Sin[t]^6,
         {t, 0, x}]],
       ReturnC[y]}]]];
CCode[p]
```

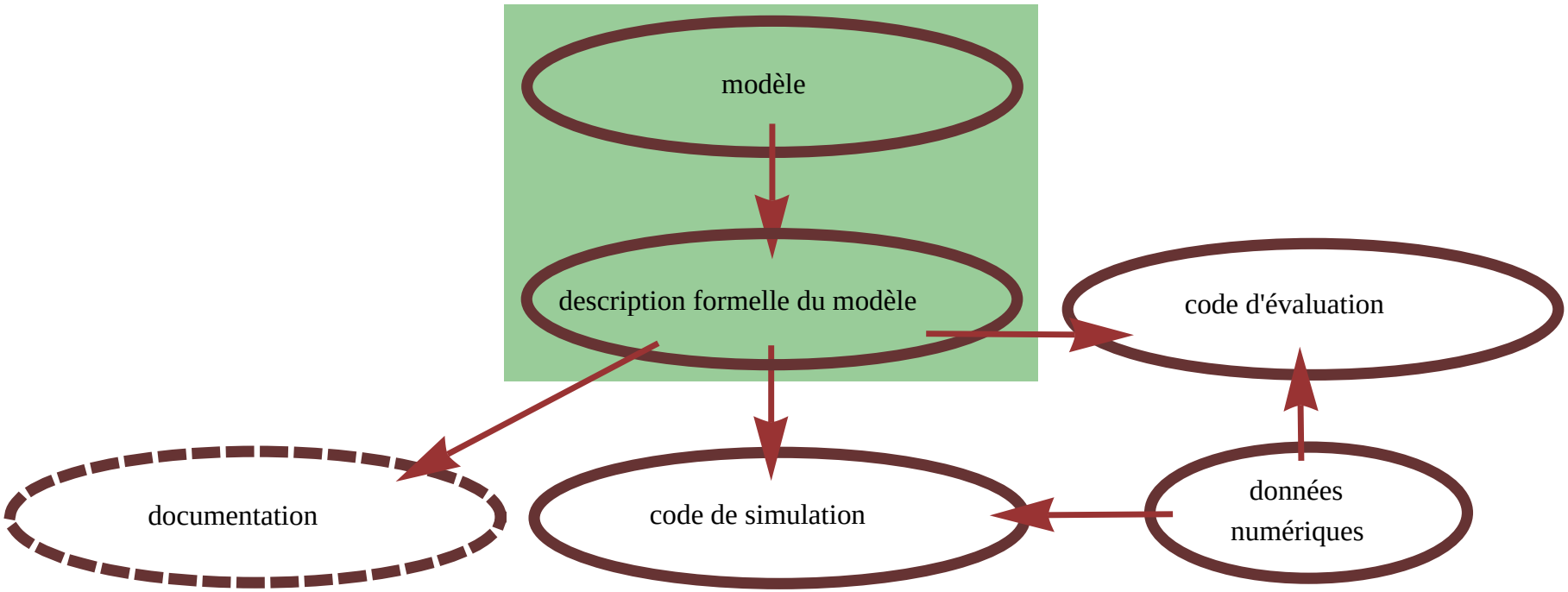
Out[42]=
```
#include "mdefs.h"
double f(double x)
{
double y;
y = 0.005208333333333333 * (60
  * x + -45 * sin(2 * x) + 9
  * sin(4 * x) - sin(6 * x));
return y;
}
```

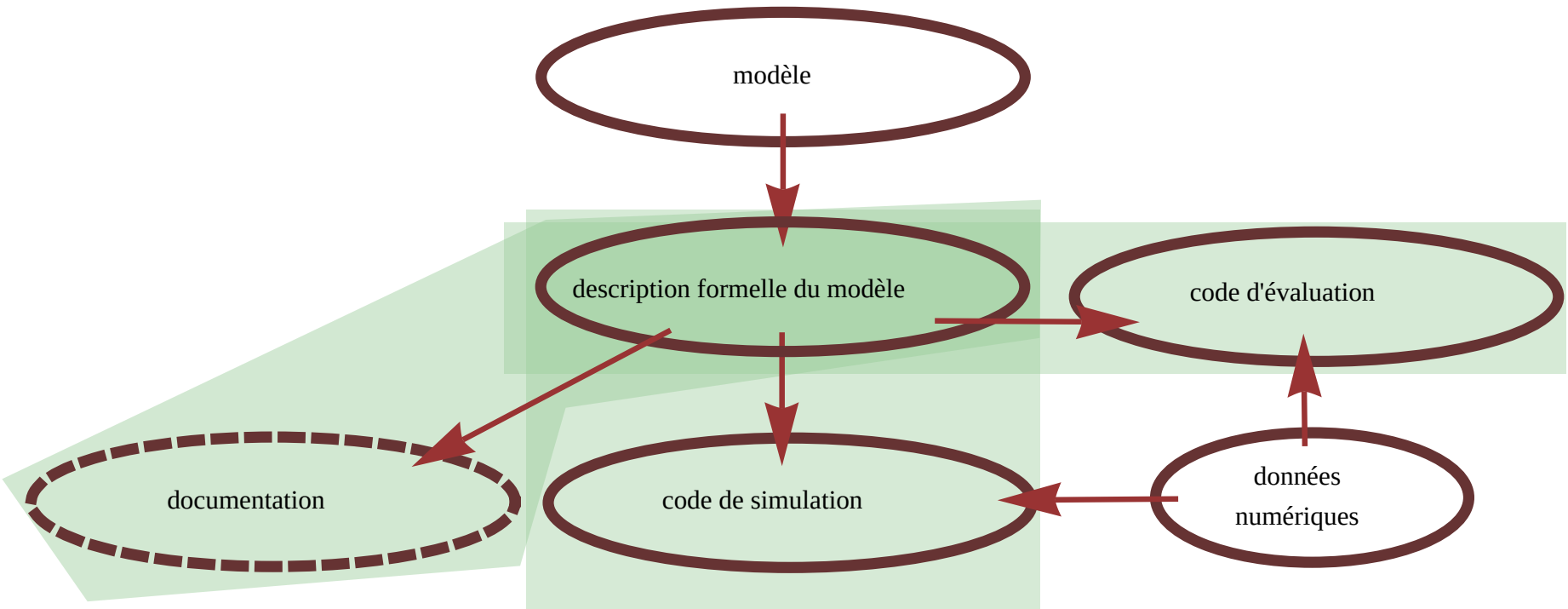
```
In[36]:= GenCode[expr_, N_Integer] :=
 CCode[With[{inputs = Union[Cases[expr, _Symbol, Infinity]]},
   ProgramC[IncludeC["math.h"], FunctionMC[double, PtrC["gen_diff" <> ToString[N]], Map[{#, double} &, inputs],
     Flatten[{DeclareC[Nest[ArrayC[#, Length[inputs]] &, double, N], m],
       MapIndexed[Function[{s, t}, AssignC[Fold[ArrayC[#1, #2 - 1] &, m, t], s]], D[expr, {inputs, N}], {N}],
       ReturnC[PtrC[m]]}]]]]

In[40]:= GenCode[expr, 2]

Out[40]= #include "math.h"
double *gen_diff2(x double, y double, z double)
{
double[3][3] m;
m[0][0] = pow(y, 3) * pow(z, 2) * sin(x * z);
m[0][1] = -3 * pow(y, 2) * z * cos(x * z) + -5 * sin((y + z));
m[0][2] = x * pow(y, 3) * z * sin(x * z) + -5 * sin((y + z)) - pow(y, 3) * cos(x * z);
m[1][0] = -3 * pow(y, 2) * z * cos(x * z) + -5 * sin((y + z));
m[1][1] = -5 * x * cos((y + z)) + -6 * y * sin(x * z);
m[1][2] = -3 * x * pow(y, 2) * cos(x * z) + -5 * x * cos((y + z));
m[2][0] = x * pow(y, 3) * z * sin(x * z) + -5 * sin((y + z)) - pow(y, 3) * cos(x * z);
m[2][1] = -3 * x * pow(y, 2) * cos(x * z) + -5 * x * cos((y + z));
m[2][2] = -5 * x * cos((y + z)) + pow(x, 2) * pow(y, 3) * sin(x * z);
return *m;
}
```

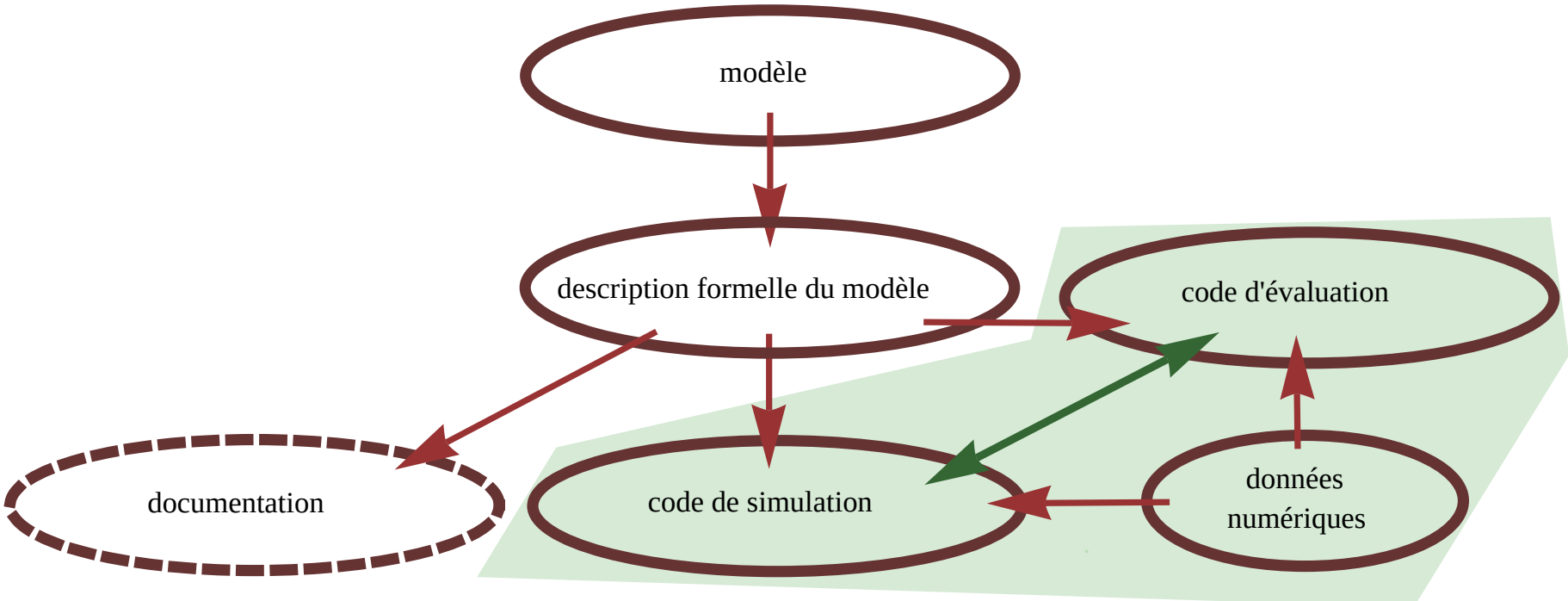
Outline


modèle
description formelle du modèle
code d'évaluation
documentation
code de simulation
données
numériques


modèle
description formelle du modèle
code d'évaluation
documentation
code de simulation
données
numériques


modèle
description formelle du modèle
code d'évaluation
documentation
code de simulation
données
numériques


modèle
description formelle du modèle
code d'évaluation
documentation
code de simulation
données
numériques


modèle
description formelle du modèle
code d'évaluation
documentation
code de simulation
données
numériques

Langage de modélisation

Caractéristiques

- simple et complet
- déclaratif

Modes du simulateur

- chargement
- initialisation
- exécution

Valeurs

- expression
- données
- instance de modèle
- programme de calcul

Composants

- éditeur de modèles
- analyseur de modèles
  - vérifications sémantique et syntaxique,
  - complétude des modèles
  - cohérence des types, des dimensions et des unités
  - détection de cycles dans le graphe de dépendance
- générateur de code d'évaluation numérique des modèles
- générateur de code C de simulation numérique respectant la norme Airbus dédiée
- validateur numérique des modèles
  - code d'évaluation
  - code C généré
- générateur de la documentation « métier » associée aux modèles

Prototype réalisé

Caractéristiques

- prototype pré-industriel opérationnel
- 34 bibliothèques de code source (14.200 lignes)
- complètement intégré
- testés sur des dizaines de modèles ($>$ 100 variables)

Produit en quelques (fractions de) secondes

- un code C compatible avec les normes de codage Airbus intégrable aux outils internes de simulation de vol,
- la documentation métier des modèles,
- un code d'évaluation numérique du modèle avec une arithmétique par intervalles.

Outline

Environnement pour la modélisation et la simulation

Spécifications

- gérer un contexte
- attacher des propriétés aux symboles
- manipuler des programmes
- évaluer de façon réversible

Qualité numérique

- calcul de sensibilité
- propagation d'incertitudes
- arithmétiques exotiques