Contraintes : équations ou procédures ?

D. Michelucci, J-P. Pernot, M. Daniel, S. Foufou

GTMG 2018, Aix-en-Provence

20-21 mars 2018

Procédures et équations

Le modeleur appelle le solveur.

Pour évaluer la partie gauche des équations, le solveur appelle des procédures (de génération, d'interrogation) du modeleur.

Les procédures appellent le solveur pour des sous-problèmes.

Arguments pour préférer les procédures

- C'est possible

- Les procédures sont plus commodes que les équations

- Parfois les équations ne sont pas disponibles

- Beaucoup d'avantages et peu d'inconvénients

Plan

Rappels sur la **modélisation géométrique par contraintes**

Avantages de remplacer les équations par des procédures

Inconvénients. Quelles méthodes deviennent inutilisables ?

Questions: peut-on utiliser les méthodes existantes pour dériver, pour décomposer, pour résoudre, pour la parcimonie ?

Réponses, Exemples, Conclusion

LA MODELISATION GEOMETRIQUE PAR CONTRAINTES

Modélisation géométrique par contraintes

- L'utilisateur fournit une esquisse, indispensable.

- Il spécifie des contraintes de distance, angle, incidence, tangence

- entre des objets Euclidiens purs: points, droites, cercles, coniques en 2D, et plans, sphères, quadriques en 3D. $\Rightarrow F(X) = 0$, avec F polynomial. Ex: problème d'Appolonius.

- ...

Modélisation géométrique par contraintes

- ou entre objets composites : segment, triangle, maillage, fractales, surface de subdivision, BRep, CSG...

Mais : ceci pose un problème combinatoire.

$\Rightarrow$ premier argument en faveur des procédures

Comment résoudre les équations ?

Décomposer et assembler les solutions des problèmes élémentaires résolus par :

- des formules (souvent suffisantes en 2D)

- la méthode des lieux : oublier une contrainte et suivre la courbe

- les homotopies trouvant toutes les racines des systèmes polynomiaux (robotique)

- ...

Résoudre avec des équations...

- Newton (amorti, généralisé), homotopie, Levenberg-Marquardt (1 seule solution)

- optimisation non contrainte : $\min ||G(X)||$, par BFGS, lm-BFGS, Hooke-Jeeves, simplexe de Nelder-Mead ou de Torczon (1 seule solution)

- optimisation contrainte : $\min ||G(X)||$ avec $F(X) = 0$ (1 seule solution)

- ...

Résoudre avec des équations...

- calcul formel (mécanisme: P Serré, F Rameau)

- analyse par intervalles (en robotique, L Jaulin, JP Merlet)

- solveurs par subdivision (Gershon Elbert et al, C Fuenfzig + DM + S Foufou)

Ces 3 méthodes deviennent inutilisables quand les équations sont remplacées par des procédures.

Décomposer, détecter les conflits

Aider les utilisateurs à détecter les conflits et à corriger est essentiel, de même que tirer parti de la parcimonie.

3 types de méthodes pour décomposer, détecter les conflits :

- structurelles, ou combinatoires

- méthodes du témoin : un exemple de solution est étudié

- protocoles appelant le solveur

Décomposer, détecter les conflits, méthodes structurelles

Les méthodes structurelles = combinatoires = graphes ou matroïdes, décomposent et détectent les conflits structurels (uniq^t).

- graphe biparti équations-inconnues, couplages, Dulmage-Mendelsohn

- graphe non biparti (en 2D, John Owen, décomposition en bi-connexes), ou hypergraphe

Utilisable avec les procédures s.c. interface modeleur-solveur.

Décomposer, détecter les conflits, avec témoin

La méthode du témoin détecte toutes les dépendances. Principe :

- Il faut résoudre $F(U^1, X) = 0 \Leftrightarrow$ trouver X^1 tq $F(U^1, X^1) = 0$.

- Un témoin U^0, X^0 tq $F(U^0, X^0) = 0$, est connu.

- Le (Jacobien du) témoin est connu, étudié, décomposé, et ses propriétés (rang de ses mineurs) sont transférées sur U^1, X^1.

- Hypothèse : U^0, X^0 est typique (ou même proche) de U^1, X^1.

- Utilisable avec les procédures s.c. interface, témoin.

Décomposer, détecter les conflits, par protocole

Autre possibilité, résoudre incrémentalement.

Dès que le solveur échoue, on se dit que la dernière contrainte introduit un conflit (... ou que le solveur est incomplet).

On peut trouver un ensemble min de contraintes conflictuelles.

Appelle le solveur, contrairement aux autres méthodes.

Toujours utilisable avec des procédures.

LES AVANTAGES DES PROCEDURES SUR LES EQUATIONS

Les procédures sont plus commodes

Les procédures sont plus commodes et plus puissantes que les equations. Exemple :

$|M(X)| = 0$. Si M_{lc} est un polynome en X, ou un Bézier ou une NURBS, développer l'équation est en temps exponentiel.

Les bibliothèques d'algèbre linéaire creuse donnent des procédures efficaces pour calculer $|M(V)|$, avec V un vecteur numérique. **Or un solveur type Newton donne toujours des valeurs V aux inconnues X.**

Les procédures sont plus commodes

Remplacer $|M(X)|$ par une procédure géométrique. Ex:

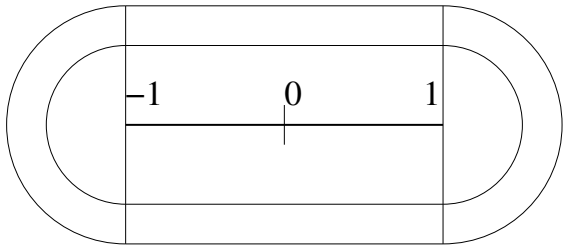
Tout modeleur fournit une procédure efficace $q = f(S, p)$, qui calcule le point q de S le plus proche du point p. S et p sont connus.

Newton donne des valeurs aux inconnues.

Donc, même si q, S, p sont des inconnues, on peut utiliser ces valeurs pour calculer la partie gauche de l'équation :
$q - f(S, p) = 0$.

Les procédures sont plus commodes

Equations de la distance de (x, y) à $[(-1, 0), (1, 0)]$?



$$
\begin{array}{lcllll}
d(x,y)^2 & = & \min(x+1,0)^2 & +y^2 & +\max(0,x-1)^2 \\
x \leq -1 & \Rightarrow & (x+1)^2 & +y^2 & +0^2 \\
-1 \leq x \leq 1 & \Rightarrow & 0^2 & +y^2 & +0^2 \\
1 \leq x & \Rightarrow & 0^2 & +y^2 & +(x-1)^2
\end{array}
$$

Les procédures sont plus commodes

En 1D, distance signée de $x \in \mathbb{R}$ à $[-1, 1]$? C'est $|x| - 1$. Non analytique $\Rightarrow$ non polynomial. Remplaçable par min, max, sgn.

Les systèmes polynomiaux ne suffisent pas :(

min $G(X)$ avec $F(X) = 0$, et F, G polynomial ? Mais :

Son Lagrangien a des racines parasites $\Rightarrow$ min, max, sgn, $|.|$ sont inévitables.

Parfois les équations ne sont pas disponibles

- surface de subdivision

- fractales

mais les procédures générant ces objets sans équations sont disponibles.

INCONVENIENTS DES PROCEDURES

Quelles méthodes ne sont plus utilisables ?

- Le calcul formel.

- L'analyse par intervalles, car il faut les équations pour limiter l'inflation des intervalles : $f(x) = x(1-x)$, $f([0,1]) \subset [0,1]$ mais $f([0,1]) = [0,1/4]$. L'arithmétique des intervalles reste utilisable pour le tolérancement.

- Les solveurs par subdivision (Elbert et Kim).

Mais : les procédures appelées par le solveur peuvent continuer à utiliser ces méthodes pour résoudre des sous-problèmes.

en partie gauche des équations procédurales :

1. Les homotopies trouvant toutes les racines ne marchent que pour les systèmes polynomiaux.

Or les polynomes par morceaux ne sont pas polynomiaux.

Ces homotopies ne peuvent plus être utilisées par le solveur. Mais les procédures du modeleur peuvent continuer à les utiliser.

2ème conséquence des fonctions non analytiques f

La dépendance de $f(X)$ à X_k est indécidable.

$\Rightarrow$ **le modeleur doit transmettre les dépendances au solveur.**

Indispensable pour décomposer, détecter les conflits, la parcimonie.

QUESTIONS POSEES PAR LES PROCEDURES

Questions : Avec les procédures, peut-on :

- calculer les dérivées ? Oui, avec les nombres duaux

- résoudre ? Oui, avec des solveurs numériques itératifs

- être parcimonieux ? Oui s.c. interface modeleur-solveur

- utiliser les méthodes structurelles ? Oui s.c. ...

- utiliser le témoin ? Oui s.c. ...

REPONSES

Dérivation automatique, nombres duaux

Les nombres duaux sont implicitement utilisés

- la méthode du témoin pour le calcul d'une base des déplacements infinitésimaux

- les quaternions duaux (ou bi-quaternions).

$\psi : \mathbb{R} \rightarrow$ quaternions (pour la représentation des rotations en 3D)

$\psi : \mathbb{R} + \epsilon\mathbb{R} = \mathbb{R}[\epsilon]/(\epsilon^2 = 0) \rightarrow$ quaternions duaux (pour la représentation des rotations et translations en 3D)

Dérivation automatique, nombres duaux

La dérivation automatique a lieu à l'exécution (et la dérivation symbolique à la compilation).

Dans l'arithmétique des nombres complexes $z = x + iy$, $x \in \mathbb{R}$, $y \in \mathbb{R}$, le terme i^2 est réduit à -1.

Dans l'arithmétique des nombres duaux : $z = x + \epsilon y$, $x \in \mathbb{R}$, $y \in \mathbb{R}$, le terme ϵ^2 est réduit à 0.

On en déduit $+, -, \times, \div$ sur les nombres duaux.

Nombres duaux, fonctions transcendantes

$$f(a+b\epsilon) = f(a) + b\epsilon\, f'(a) + 1/2(b\epsilon)^2 f''(a) + \ldots \Rightarrow$$

$$\begin{aligned}
\exp(a+b\,\epsilon) &= e^a + be^a\,\epsilon \\
\cos(a+b\epsilon) &= \cos(a) - b\sin(a)\,\epsilon \\
\sin(a+b\epsilon) &= \sin(a) + b\cos(a)\,\epsilon \\
\tan(a+b\epsilon) &= \tan(a) + b(1+\tan^2(a))\epsilon \\
|a+b\epsilon| &= |a| + (\mathrm{sgn}(a)b + (1-\mathrm{sgn}(a)^2)|b|)\epsilon
\end{aligned}$$

Nombres duaux, plusieurs ϵ

$$\exp(a + b_1\epsilon_1 + b_2\epsilon_2) = e^a + b_1 e^a \epsilon_1 + b_2 e^a \epsilon_2$$

$$F(X_v + \sum_k \epsilon_k) = F(X_v) + \sum_k \partial F/\partial x_k(X_v)\epsilon_k$$

Calcul de déterminants : $|A + \epsilon B|$. Il y a aussi des formules.

Nombres duaux, autre généralisation

$\epsilon^2 \neq 0$, mais $\epsilon^3 = 0$

Si plusieurs ϵ_i, prévoir des $\epsilon_{ij} = \epsilon_i \epsilon_j$ et $\epsilon_i \epsilon_j \epsilon_k = 0$

Dérivation automatique, nombres duaux

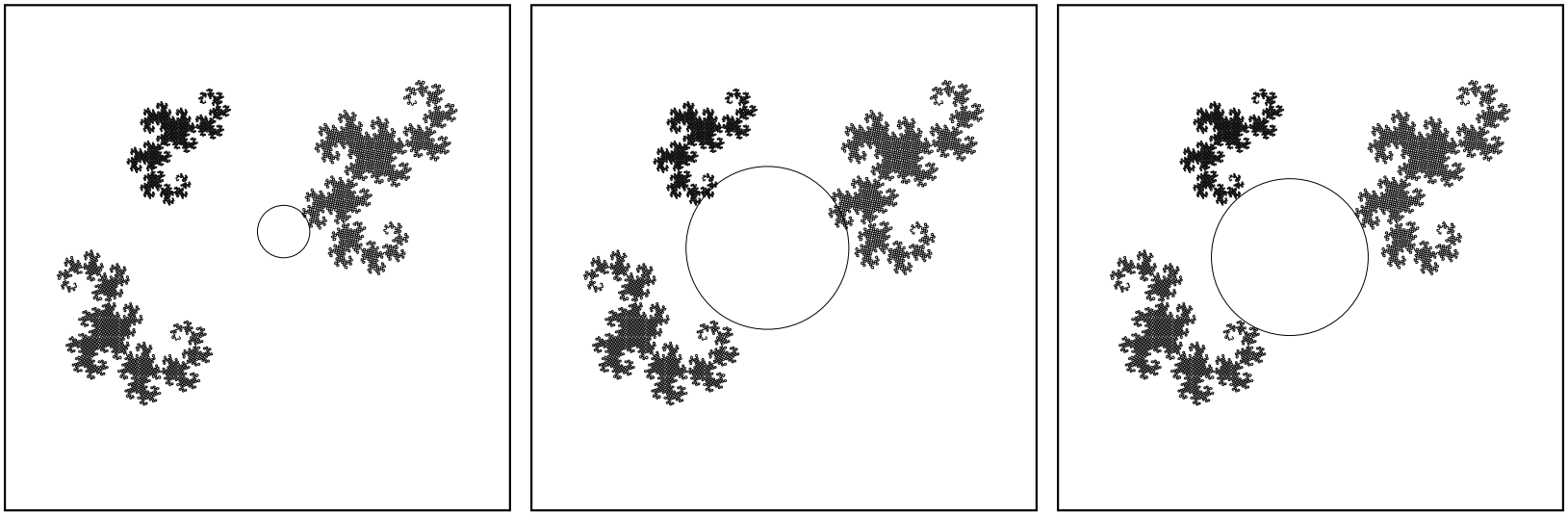
Une forme algorithmique (mécanisme, engrenages) dépend d'un petit nombre $n = 10$ de paramètres $U = (u_1, ...u_n)$.

Une simulation FEM utilise les nombres duaux pour calculer une performance $p(U) \Rightarrow$ elle calcule $\nabla p = (\partial p / \partial U(U))$.

$\Rightarrow$ on peut optimiser par L-BFGS $\Rightarrow U^* = \text{argmax } p(U)$.

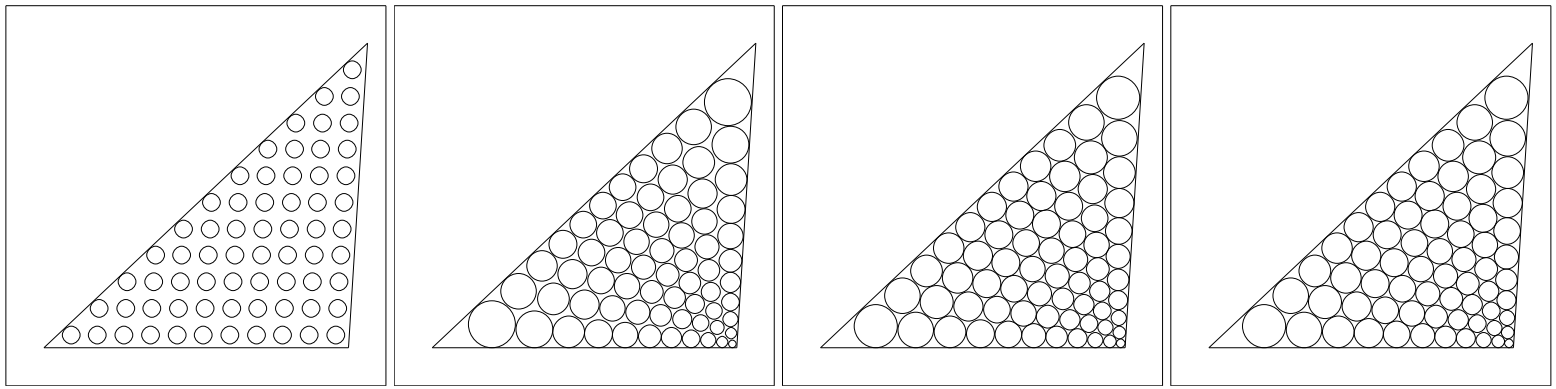
Il suffit de n infinitésimaux (nombres duaux), même si p génère un maillage avec $N = 10^6$ sommets, $N \gg n$.

Résoudre sans les équations



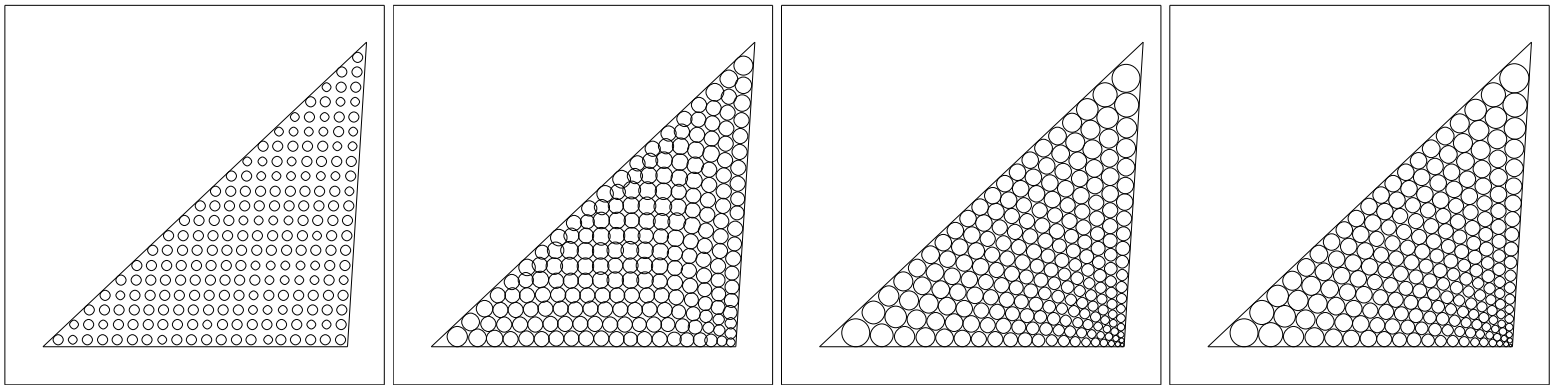
Généralisation d'Appolonius à 3 objets quelconques A, B, C.
$(d_A(x,y) - R)(D_A(x,y) - R) = (d_B(x,y) - R)(D_B(x,y) - R) = (d_C(x,y) - R)(D_C(x,y) - R) = 0$ avec $d_A(x,y)$ la plus courte distance à A de (x,y), et $D_A(x,y)$ la plus grande.

Résoudre sans les équations



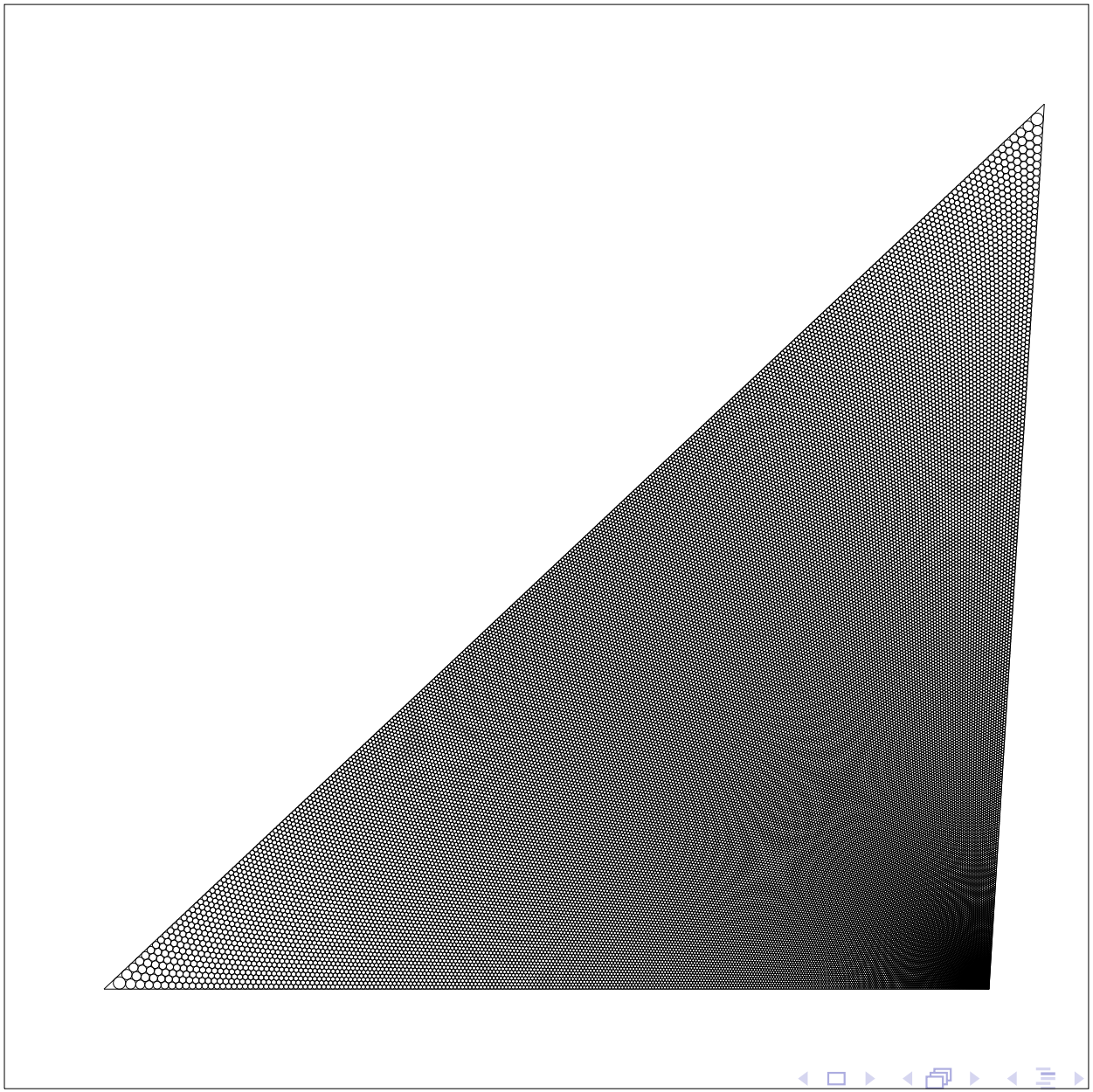
Résolution avec Newton (11 rangées). Trois itérations suffisent

Résoudre sans les équations



Quelques étapes de résolution avec Hooke-Jeeves (20 rangées)

Résoudre sans les équations



Résoudre sans les équations

r	50	100	150	200	300	400
c	1275	5050	11325	20100	45150	80200
u	3825	15150	33975	60300	135450	240600

r, c, u sont les nombres de rangées, de cercles, et d'inconnues

Résoudre sans les équations

Algorithme	Complexité
lm-BFGS ($m = 10$)	$O(n^{1.33})$
Newton	$O(n^{1.4})$
Levenberg-Marquardt	$O(n^{1.4})$
Hooke-Jeeves	$O(n^{1.9})$

n est le nombre de cercles ou d'inconnues et de contraintes (et non le nombre de rangées). Cette table donne la complexité empirique (la pente du diagramme log-log) des algorithmes pour le problème de test, en exploitant la faible densité des systèmes.

Interface pour la parcimonie

Soit les DAG

Soit les tableaux F, U, X^0 pour $F(U, X) = 0$, et un tableau $D[f]$: liste des variables X_k dont dépend $F[f]$.

Ceci donne le graphe biparti entre les F_f et les X_k.

Interface pour utiliser le témoin

Pour décomposer en sous-systèmes rigides, la méthode du témoin construit une base des mouvements rigides infinitésimaux (nombres duaux), indépendante des contraintes.

Pour les variables qui sont des coordonnées, il lui faut une étiquette :

- le x, le y ou le z d'un point. Pour un même point, il faut les lier : X_3 est le x du point (X_3, X_4, X_5).

- le x, le y ou le z d'un vecteur.

- le a, le b, le c d'un vecteur normal, ou le a, b, c, d d'un plan.

Problèmes techniques

Gérer les bornes: $u \in [0, 1]$

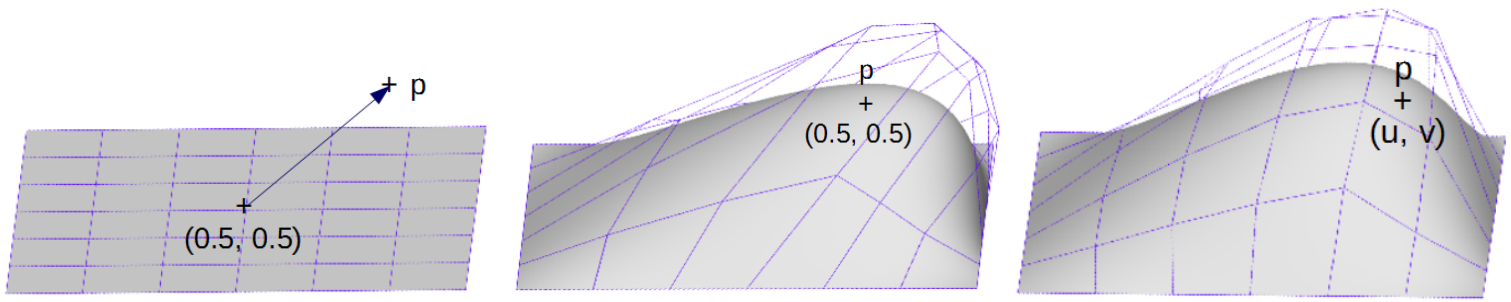
Si une procédure calcule $f(X) \in \mathbb{R}^3$, alors elle équivaut à 3 fonctions: f_x, f_y, f_z, par ex. pour le graphe biparti équations-inconnues. Ne pas évaluer f 3 fois : "mémoïsation".

Conditions mathématiques nécessaires : continuité et dérivabilité p.p. comme d'habitude.

EXEMPLES (Gilles Gouaty, JP Pernot)

GCS Exemple 1/3 en CFAO

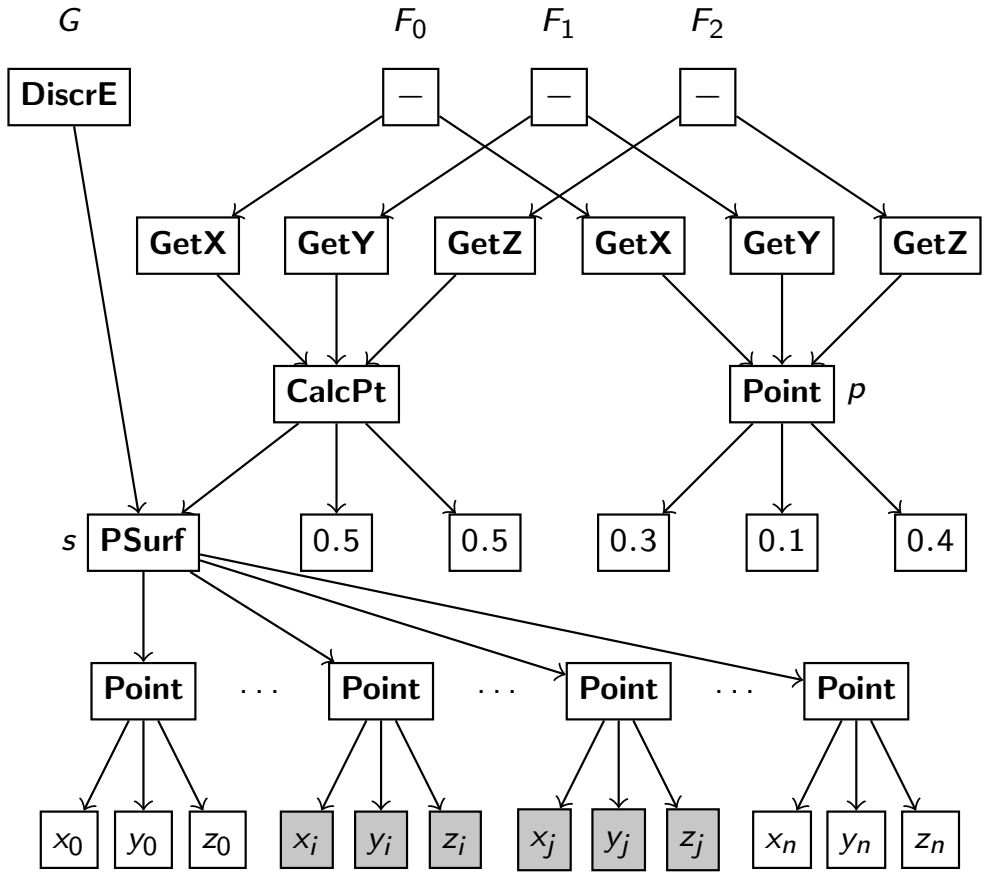
Gouaty et al: Variational geometric modeling with black box constraints and DAGs. CAD 75–76(2016)1–12.



Initial position, final positions: fixed parameters, non fixed parameters.

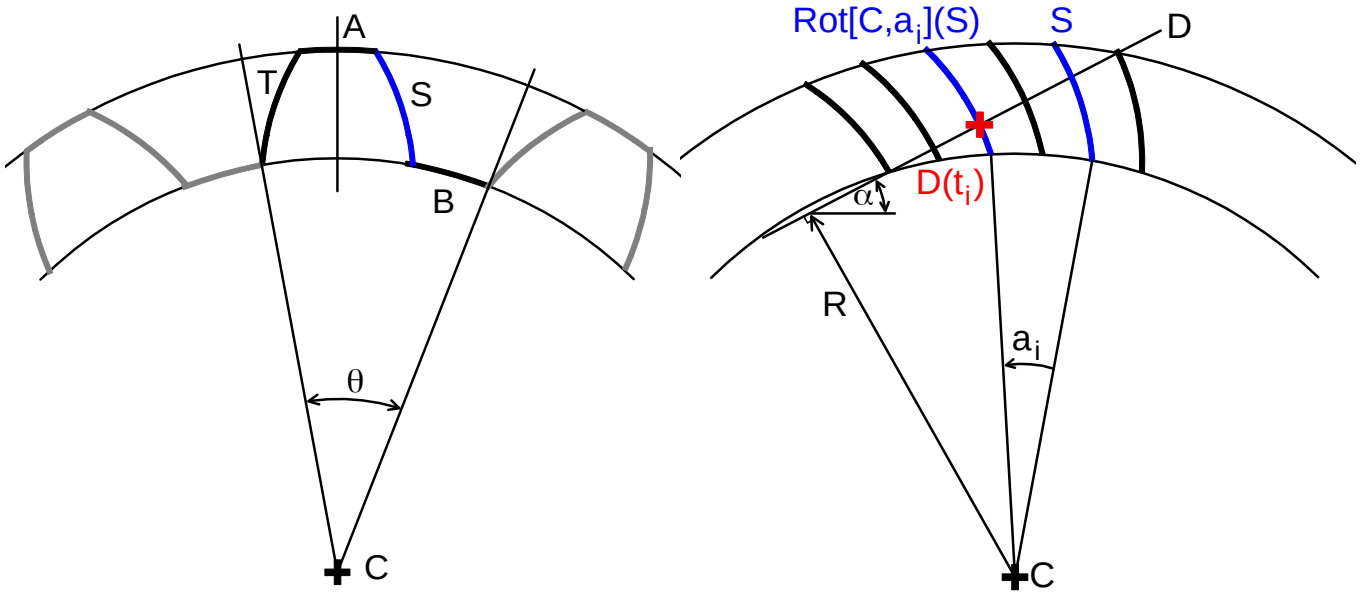
Python source. A given point must lie on a Bspline:

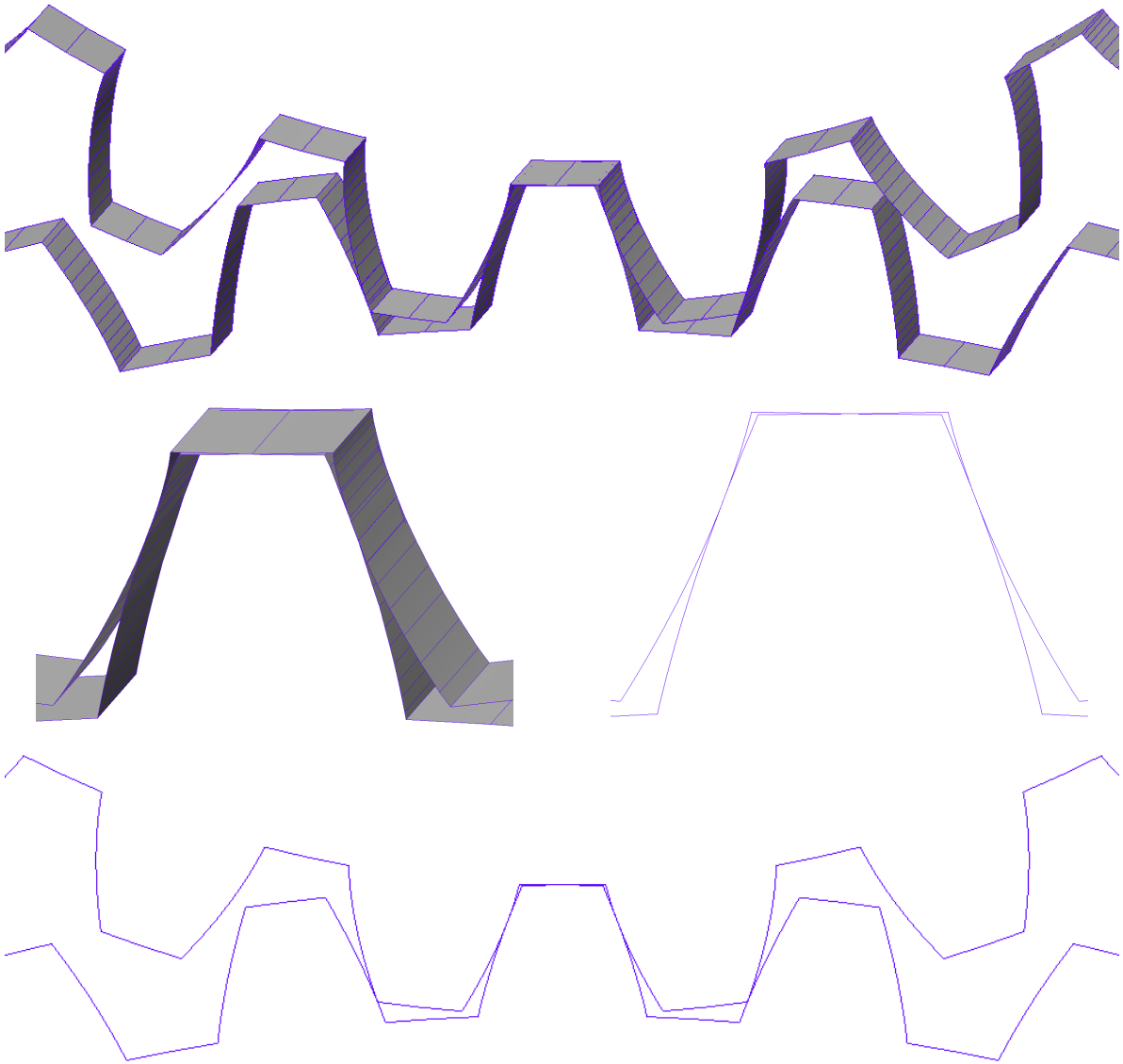
```
1  def point_de_passage() :
2      res = Desc()
3      s = Surf(7, 7)
4      s.bordsFixes()
5      p = Point(0.3, 0.1, 0.4)
6      res.addEqs(Egalite(CalcPt(s, 0.5, 0.5), p))
7      res.fMin = EnDiscr(s)
8      res.addObj(s)
9      return res
10
11 def Egalite(p1, p2) :
12     res = Desc()
13     res.addEq(getX(p1) - getX(p2))
14     res.addEq(getY(p1) - getY(p2))
15     res.addEq(getZ(p1) - getZ(p2))
16     return res
```



DAG du système. Inconnues en gris. Gilles Gouaty et JP Pernot.

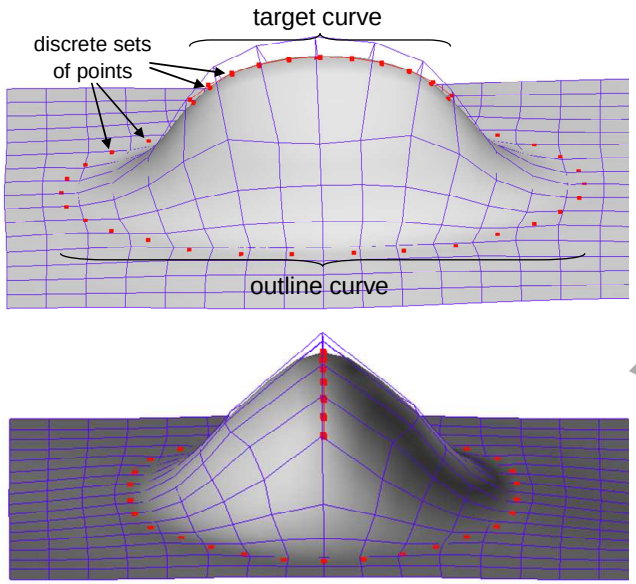
Exemple 2/3 en CFAO: 2 roues dentées (Gouaty, Pernot)

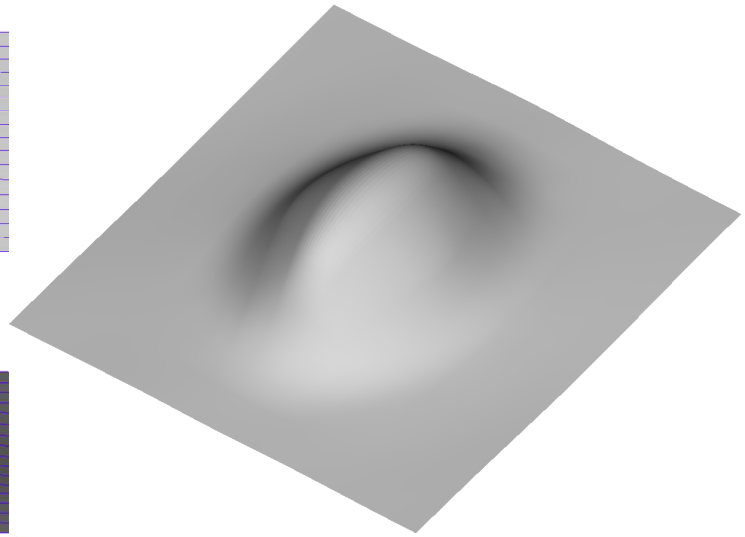


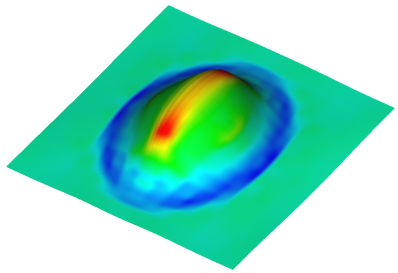


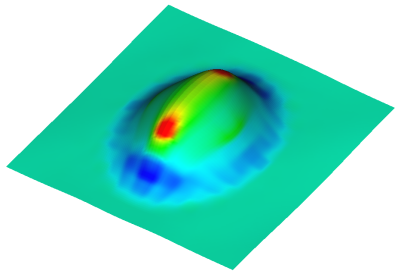
Les deux roues dentées (pas de jeu). 173 équations, 90 inconnues.

Example 3/3 in CADCAM: Creux dans une portière









908 inconnues, 213 équations.

Conclusion

Remplacer les équations par des procédures a peu d'inconvénients, et beaucoup d'avantages :

- efficacité des procédures (GPU), simplicité pour poser les contraintes

- des contraintes plus générales,

- des objets contraints plus généraux (objets composites, fractales, surfaces de subdivision) avec des formats hétérogènes.

- utilise les méthodes classiques pour décomposer, corriger, résoudre.

Travaux futurs

- un format STEP de contraintes ? faciliter l'inter-opérabilité ?

- Gérer un couple (G, C), avec G: géométrie, C: contraintes procédurales. Deux problèmes :

– modifier C et actualiser G = résoudre avec une esquisse

– modifier G et actualiser C = rétro-ingéniérie, reconnaissance de contraintes, analyse de formes.

Merci pour votre attention.