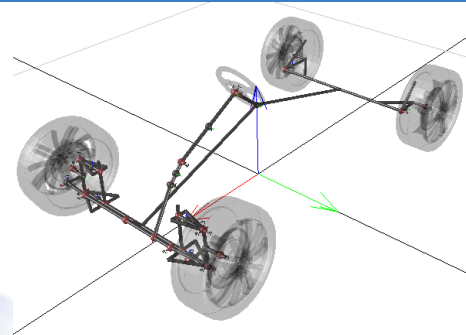
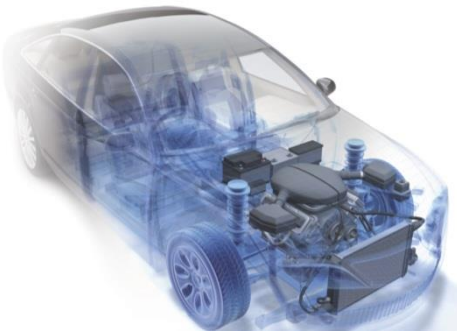


Calcul haute performance Validation statistique du contrôle d'une éolienne

Nicolas Gachadoit, Ingénieur d'Applications



Introduction

- Dans un contexte de « Smart Grids », nécessité de contrôler l'orientation et la vitesse de rotation des éoliennes pour que l'énergie électrique fournie soit la plus constante possible malgré des conditions de vent variable
- Modèles de vent statistiques → réalisation de simulations très longues ou d'un grand nombre de simulations pour balayer de nombreux cas de figure
- Objectif : utilisation de techniques de calcul haute performance pour réduire le temps de simulation global

Sommaire

Présentation du modèle d'éolienne

Simulation interactive, mesure et analyse des performances

Réduction du temps de simulation via génération de code sur la base d'équations optimisées

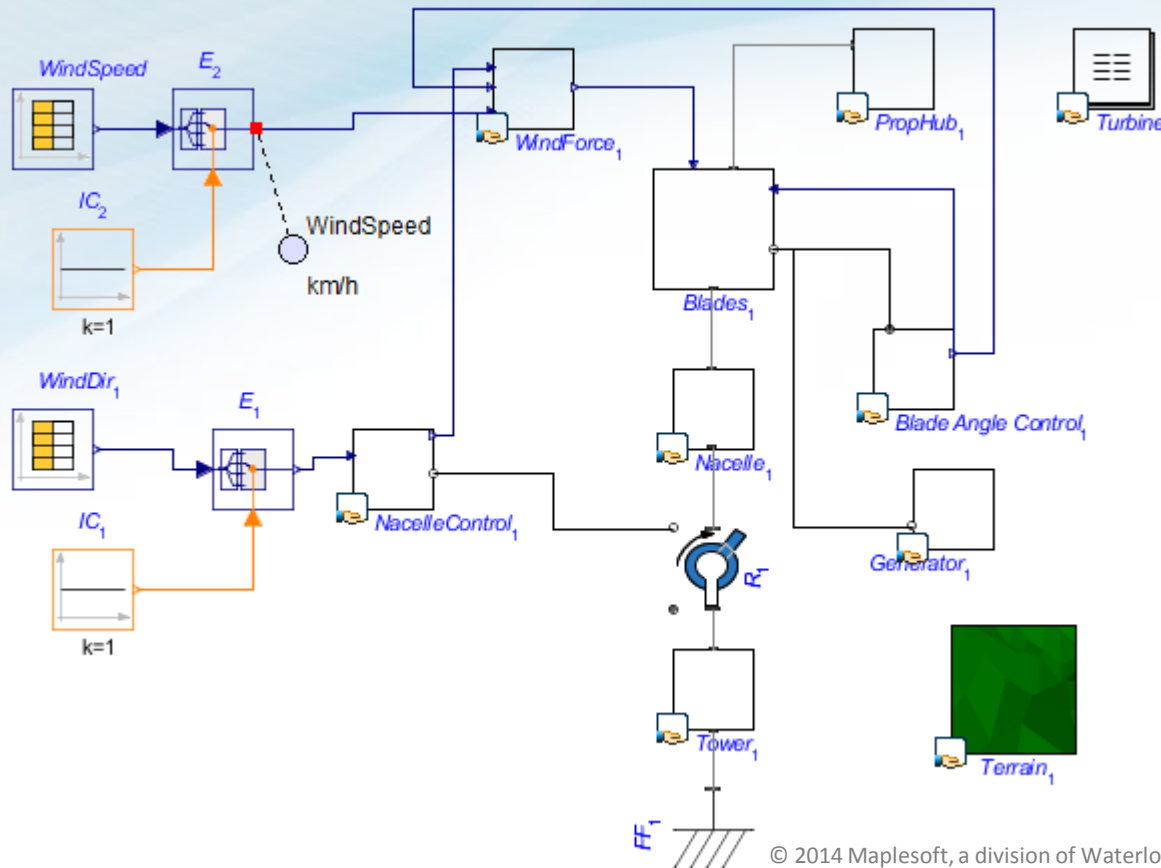
Distribution des simulations sur une grille de calculateurs

Conclusion

Modèle (physique multi-domaine) d'éolienne

Modèle (physique multi-domaine) d'éolienne

- Modèle réalisé avec MapleSim, logiciel de modélisation physique multi-domaine basé sur Modelica:



Modèle (physique multi-domaine) d'éolienne

- **Domaines couverts par le modèle:**
 - Mécanique poly-articulée
 - Electricité
 - Signal et contrôle
- **Sous-ensembles modélisés**
 - Mécanique du système
 - Motorisation et contrôle de l'orientation du pylône en fonction de la direction du vent et de l'orientation des pales en fonction de la vitesse du vent
 - Génération de la direction et de la vitesse du vent
 - Mesure de l'énergie électrique produite

Simulation interactive

Simulation interactive

- **Permet de mettre au point le modèle**
- **Temps de simulation : 40 s de temps machine pour 200 s de temps réel**
- **Inadapté pour lancer un grand nombre de simulations**
 - Chaque simulation prend trop de temps
 - Interactif...
Les simulations pourraient cependant être lancées par programme depuis Maple

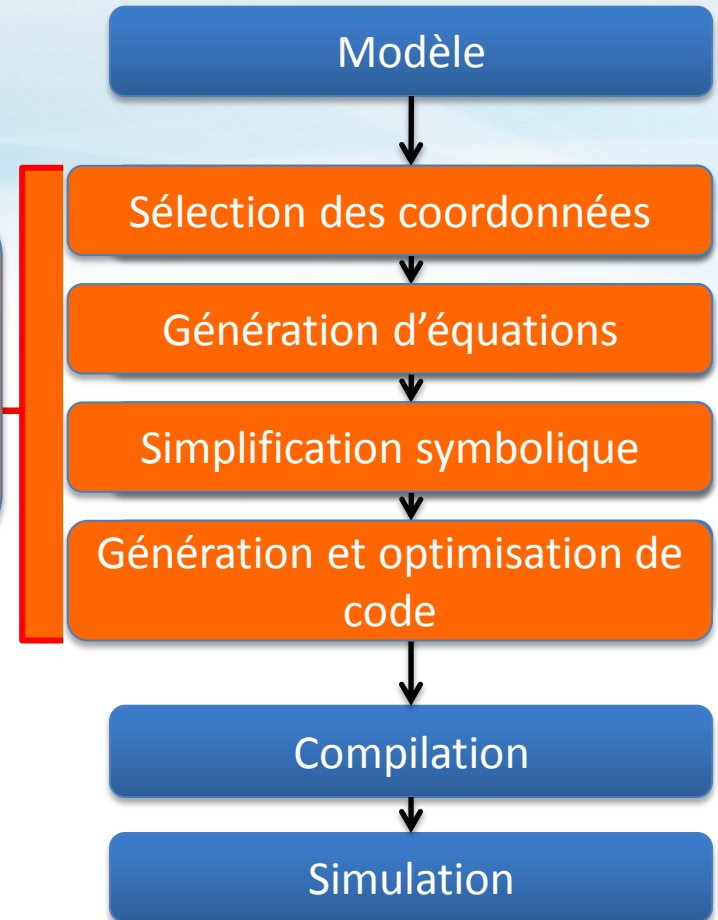
Accélération via optimisation et génération de code

Optimisation et génération de code

- Génération d'une procédure compilée à partir du modèle

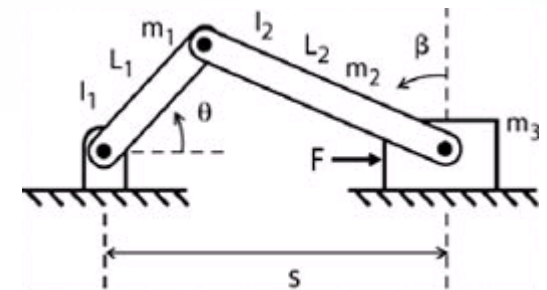
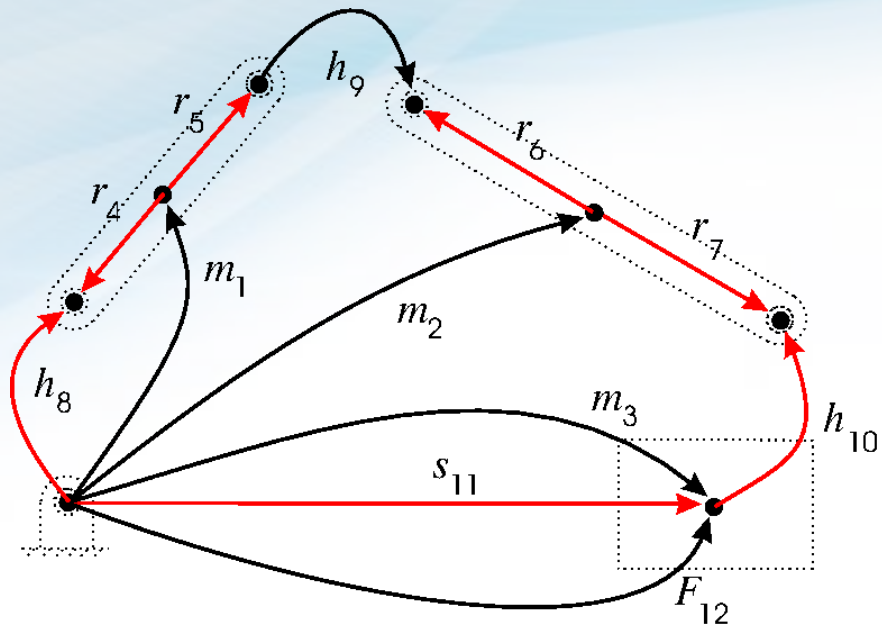
MapleSim applique 4 niveaux d'optimisation

- La formulation symbolique sous-jacente au modèle permet des optimisations très efficaces



Optimisation et génération de code

- Sélection du système de coordonnées



$$Tree = \{r_4 - r_7, h_8, h_{10}, s_{11}\} \Rightarrow \mathbf{q} = [\theta_8, \beta_{10}, s_{11}]^T$$

$$Tree = \{r_4 - r_7, m_1, m_2, m_3\} \Rightarrow \mathbf{q} = [x_1, y_1, \theta_1, x_2, y_2, \theta_2, x_3, y_3, \theta_3]^T$$

Optimisation et génération de code

- Génération d'équations et simplification symbolique

$$\begin{bmatrix} Y1_{t+1} \\ Y2_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & Y1_t & c \\ Y2_t & 0 & e \end{bmatrix} \cdot \begin{bmatrix} c \\ Y2_t \\ -1 \end{bmatrix} \quad \longrightarrow \quad \begin{bmatrix} Y1_{t+1} \\ Y2_{t+1} \end{bmatrix} = \begin{bmatrix} Y1_t Y2_t \\ c Y2_t - e \end{bmatrix}$$

6 mult, 4 add/sub 2 mult, 1 add/sub

$$\begin{aligned} & \frac{1}{2500} \cos(\beta_{UI}(t))^2 + \frac{1}{250} \sin(\beta_{UI}(t))^2 \\ & + \frac{9}{800} \cos(\beta_{UI}(t))^2 \cos(\alpha_{UI}(t))^2 \\ & + \frac{9}{800} \cos(\beta_{UI}(t))^2 \sin(\alpha_{UI}(t))^2 \end{aligned} \quad \longrightarrow \quad \frac{153}{20000} \cos(\beta_{UI}(t))^2 + \frac{1}{250}$$

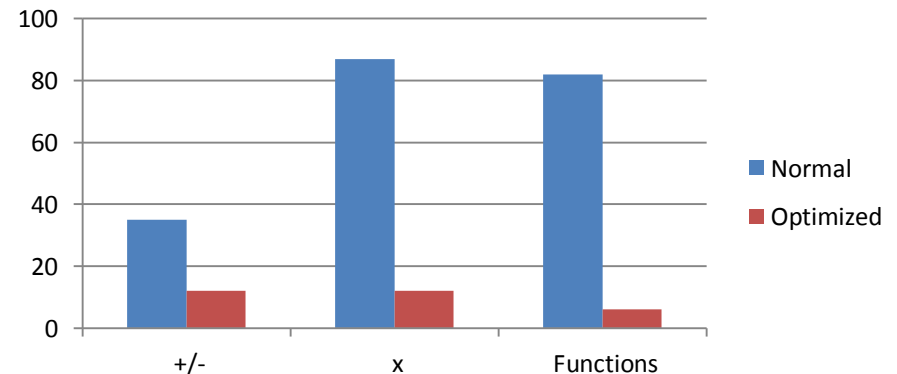
Optimisation et génération de code

- Génération et optimisation de code

$$\begin{aligned}
 & [[-1. - 1. s + X \cos(ang) \cos(\beta) + Y \sin(ang) \cos(\beta) \\
 & + 1. Xp \cos(ang) \cos(\beta) + 1. Yp \sin(ang) \cos(\beta) \\
 & - 1. Yg \sin(ang) \cos(\beta) + \cos(\alpha) \sin(\beta) Z \\
 & - 1. Xg \cos(ang) \cos(\beta) - 1. Xp \sin(ang) \sin(\alpha) \sin(\beta) \\
 & + 1. Yp \cos(ang) \sin(\alpha) \sin(\beta) + Xg \sin(ang) \sin(\alpha) \sin(\beta) \\
 & - 1. Yg \cos(ang) \sin(\alpha) \sin(\beta) + Y \cos(ang) \sin(\alpha) \sin(\beta) \\
 & - 0.4000000000 \cos(\alpha) \sin(\beta) - 1. X \sin(ang) \sin(\alpha) \sin(\beta)], \\
 & [-1. \sin(ang) \cos(\alpha) X + \sin(ang) \cos(\alpha) Xg \\
 & + \cos(ang) \cos(\alpha) Y - 1. \cos(ang) \cos(\alpha) Yg \\
 & + 0.4000000000 \sin(\alpha) - 1. \sin(ang) \cos(\alpha) Xp \\
 & + 1. \cos(ang) \cos(\alpha) Yp - 1. \sin(\alpha) Z], \\
 & [-1. X \sin(ang) \sin(\alpha) \cos(\beta) + Y \cos(ang) \sin(\alpha) \cos(\beta) \\
 & - 1. Xp \sin(ang) \sin(\alpha) \cos(\beta) + 1. Yp \cos(ang) \sin(\alpha) \cos(\beta) \\
 & + Xg \sin(ang) \sin(\alpha) \cos(\beta) - 1. Yg \cos(ang) \sin(\alpha) \cos(\beta) \\
 & - 1. Xp \cos(ang) \sin(\beta) + Yg \sin(ang) \sin(\beta) \\
 & - 0.4000000000 \cos(\alpha) \cos(\beta) + Xg \cos(ang) \sin(\beta) \\
 & - 1. Y \sin(ang) \sin(\beta) - 1. X \cos(ang) \sin(\beta) \\
 & + \cos(\alpha) \cos(\beta) Z - 1. Yp \sin(ang) \sin(\beta)]]
 \end{aligned}$$

$$\begin{aligned}
 & [t1 = \cos(ang), t2 = \cos(\beta), t3 = \sin(ang), t4 = \cos(\alpha), t5 = \sin(\beta), t6 \\
 & = \sin(\alpha), t7 = -X + Xg - Xp, t8 = Y - Yg + Yp, t9 = t8 t1 \\
 & + t7 t3, t10 = -\frac{2}{5} + Z, t11 = t6 t9 + t10 t4, t1 = t7 t1, t3 = t8 t3, \\
 & CON1 = (-t1 + t3) t2 + t11 t5 - 1 - s, CON2 = t4 t9 - t10 t6, \\
 & CON3 = t11 t2 + (t1 - t3) t5]
 \end{aligned}$$

Comparaison des coûts de calcul entre un code normal et un code optimisé



Optimisation et génération de code

- La simplification symbolique, la génération de code et la compilation sont réalisées par une seule fonction de l'API MapleSim, *GetCompiledProc()*, exécutée depuis Maple
- Le résultat est une procédure compilée :
 - qui s'exécute depuis Maple
 - qui peut être lancée par programme un très grand nombre de fois avec des données de vent différentes
- La compilation est exécutée une seule fois
- Chaque simulation est beaucoup plus rapide : 1.5 s de temps machine pour 200 s de temps réel

Calcul en grille

Calcul en grille

- L'étape précédente a permis d'améliorer le temps d'une simulation
- Pour améliorer le temps de simulation global d'un grand nombre de simulations, on peut les distribuer sur une « grille de calculateurs » qui peut être
 - Un ordinateur « classique » avec plusieurs cœurs
 - Un ordinateur avec plusieurs processeurs
 - Plusieurs ordinateurs, eux-mêmes dotés de plusieurs cœurs ou processeurs
- Nous appellerons « nœud de calcul » une entité capable de réaliser une simulation de façon indépendante des autres

Calcul en grille

- **2 possibilités (avec utilisation des mêmes codes)**
 - Utiliser Maple seul → nombre de nœuds limité à ceux de l'ordinateur
 - Utiliser la Grid Computing Toolbox → permet de déployer sur une « vraie » grille avec de très nombreux nœuds de calcul
- **Ceci permet encore d'accélérer le temps global :**
4 s pour 8 simulations réparties sur 8 nœuds
→ moyenne de 0.5 s de temps machine pour 200 s de temps réel

Conclusion

Conclusion

- **Les simulations interactives (ou leur automatisation) ne sont pas acceptables lorsqu'il s'agit de lancer un grand nombre de simulations**
- **L'accélération d'un ensemble de simulations peut se faire en combinant:**
 - Un modèle compilé à partir d'un code C optimisé, lui-même généré à partir du jeu d'équations (simplifié symboliquement) correspondant au modèle
 - La distribution des simulations sur une grille de calculateur
- **Possibilité de récupérer le code source C du modèle optimisé pour intégrer dans une plateforme quelconque**

Questions ?