Suivi de chemin par méthodes homotopiques différentielles : application au contrôle optimal.

Olivier Cots

Université de Toulouse
INP-ENSEEIHT-IRIT

8 septembre 2009

Objectif et contexte

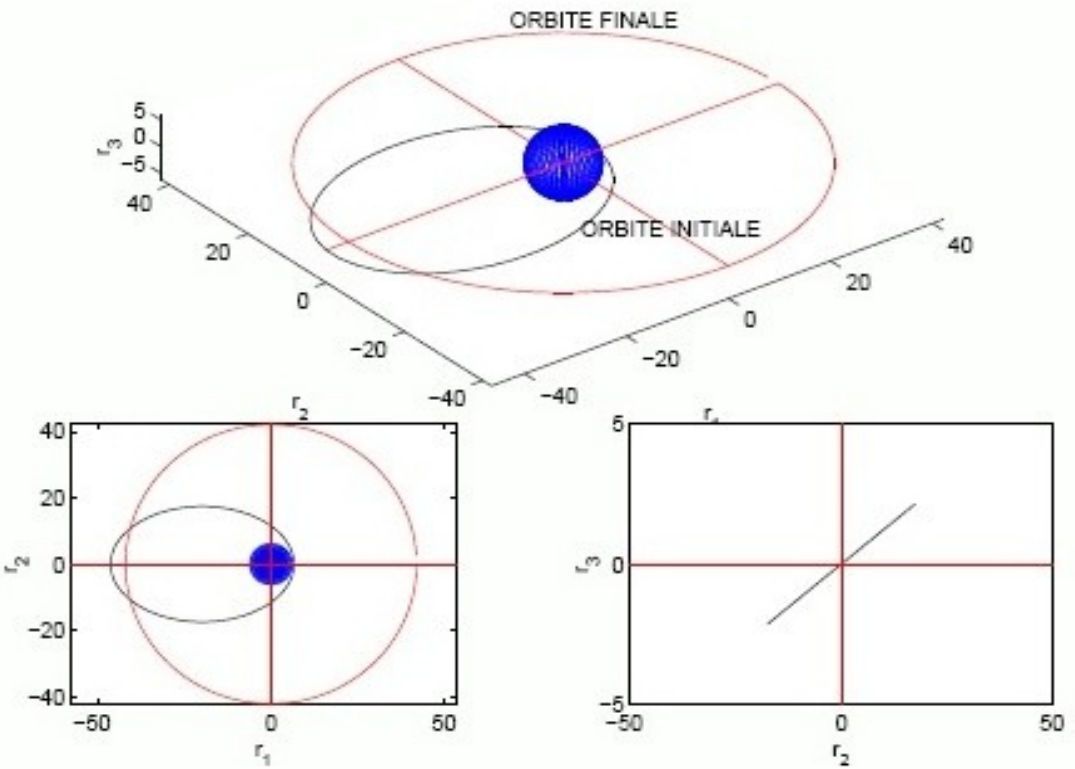
Cadre

- Contexte : avec le groupe Contrôle de l'équipe **APO** (Algorithmique Parallèle et Optimisation) de l'**IRIT** (Institut de recherche en Informatique de Toulouse), en collaboration avec l'**Université de Bourgogne**.
- Objectif : Réaliser un programme en fortran 90, permettant la résolution de deux problèmes de **contrôle optimal** particuliers.

Les problèmes

- Un problème de **transfert orbital** à moyenne ou faible poussée, dont le modèle vient de la collaboration avec le **CNES** (Toulouse, Evry).
- Et un autre en contrôle de la **dynamique quantique** par champs laser. Etudié à l'**Université de Bourgogne**, en partenariat avec l'Agence National pour la Recherche (**ANR-CNRS**).

Transert orbital



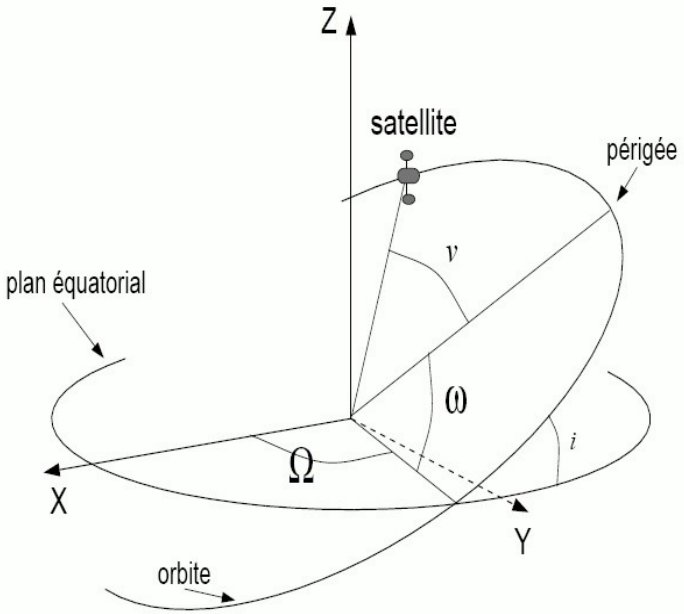
- Transfert LEO - GEO : $P_0 = 11.625$ Mm, $e_0 = 0.75$ et $i_0 = 7°$
- Masse initiale : $m_0 = 1500$ kg
- Poussées faibles : $T_{\max} = 0.1$ N
- Coûts
  - Minimisation de t_f
  - Maximisation de la masse finale
  - Minimisation de "l'énergie"

Coordonnées

- Coordonnées cartésiennes (on néglige le terme en J_2)

$$\ddot{r} = -\frac{\mu_0}{|r|^3} r + \frac{T}{m}$$

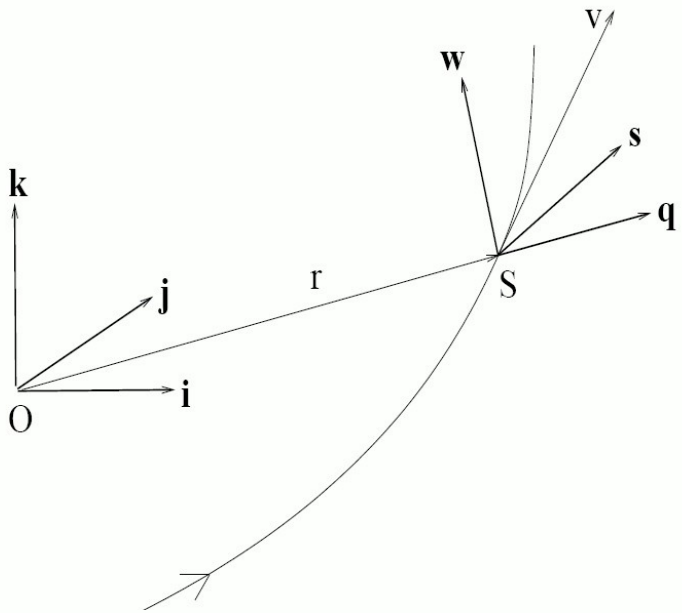
- Coordonnées de Gauss modifiées $x = (P, e_x, e_y, h_x, h_y, L)$:



$$\begin{cases} P &= a(1-e^2) \\ e_x &= e\cos(\Omega+\omega) \\ e_y &= e\sin(\Omega+\omega) \\ h_x &= \tan(i/2)\cos\Omega \\ h_y &= \tan(i/2)\sin\Omega \\ L &= \Omega+\omega+\nu+2\pi n \text{ (cumulée)} \end{cases}$$

$n =$ Nombre de révolutions

Contrôle



- Repère ortho-radial (q, s, w)
- On normalise le contrôle
  $T = T_{\max} u$

  La contrainte sur le contrôle devient
  $|u| \leq 1 \left(|u| = \sqrt{u_1^2 + u_2^2 + u_3^2} \right)$
- Équation d'état :
  $\dot{x} = f_0(x) + \frac{Tmax}{m} \sum_{i=1}^{3} f_i(x) u_i$
  $\dot{m} = -\frac{T_{\max}}{I_{\mathrm{sp}} g_0} |u|$

Problème de contrôle optimal

$$(P)\begin{cases} \min \int_0^{t_f} |u|dt \iff \max m(t_f) \\ \dot{x} = f_0(x) + \frac{T_{max}}{m}\sum_{i=1}^{3} u_i f_i(x) \\ \dot{m} = -\beta T_{max}|u| \\ |u| \leq 1 \\ \text{Conditions initiales et finales} \\ t_f \text{ ou } L_f \text{ fixé} \end{cases}$$

avec $x = (P, e_x, e_y, h_x, h_y, L)$, et comme conditions initiales et finales :

$P^0 = 11625km$	$P^f = 42165km$
$e_x^0 = 0.75$	$e_x^f = 0$
$e_y^0 = 0$	$e_y^f = 0$
$h_x^0 = 0.0612$	$h_x^f = 0$
$h_y^0 = 0$	$h_y^f = 0$
$L^0 = \pi$	$L^f =$ libre ou fixé
$m^0 = 1500kg$	$m^f =$ libre

Problème général

$$(P)\begin{cases} Min \;\; g(t_0, x(t_0), t_f, x(t_f)) + \int_{t_0}^{t_f} l(t, x(t), u(t))dt & \text{fonction objective} \\ \dot{x} = f(t, x, u) \in \mathbb{R}^n & \text{dynamique} \\ u \in U \subset \mathbb{R}^m & \text{contrôle admissible} \\ \psi(t_0, x(t_0)) = 0 & \text{conditions initiales} \\ \psi(t_f, x(t_f)) = 0 & \text{conditions finales} \end{cases}$$

Problème aux deux bouts

Résoudre (P) $\equiv$ **Résoudre (BVP)** (Principe du maximum de Pontriaguine)

$$(BVP) \begin{cases} \dot{y}(t) = \varphi(t, y(t)) & \text{dans } [t_0, t_f] \\ c_0(t_0, y(t_0)) = 0 \in \mathbb{R}^n & \text{conditions initiales} \\ c_f(t_f, y(t_f)) = 0 \in \mathbb{R}^n & \text{conditions finales} \end{cases}$$

où :

- $y = (x, p) \in \mathbb{R}^{2n}$ (état, état adjoint)
- $\varphi(t, y) = (\frac{\partial H}{\partial p}(t, y, u), -\frac{\partial H}{\partial x}(t, y, u))$ ($H(t, y, u)$ hamiltonien)
- $\bar{u}(t) = ArgMin_{w \in U} H(t, y, w)$
- c_0 et c_f sont obtenues à partir des conditions de transversalité.

IVP et fonction de tir

On introduit alors :

- $$(IVP)\begin{cases} \dot{y}(t) = \varphi(t, y(t)) & \text{dans } [t_0, t_f] \\ y(t_0) = z & \text{valeur initiale} \end{cases}$$

- $$\text{et } S : z \longmapsto \begin{pmatrix} c_0(t_0, z) \\ c_f(t_f, y(t_f, z)) \end{pmatrix} \quad \text{la fonction de tir,}$$

où $y(t_f, z)$ est la solution en t_f de l'IVP.

Alors **résoudre (BVP)** $\equiv$ **Trouver un zéro de S**

=> La méthode de tir consiste à résoudre l'équation **S(z)=0**.

Remarques

Le tir simple (méthode indirecte) :

- très précise comparée aux méthodes dites directes ;
- mais très **sensible au choix du point de départ** lors de la recherche de zéros (algorithme de type Newton).

=> **méthode de continuation** : permet de trouver un bon point de départ pour la résolution de problèmes difficiles.

Principe général

- Relier de manière continue une fonction G facile à résoudre à la fonction de tir S de notre problème.

- Par exemple :

$$h(z, \lambda) = \lambda S(z) + (1 - \lambda) G(z)$$

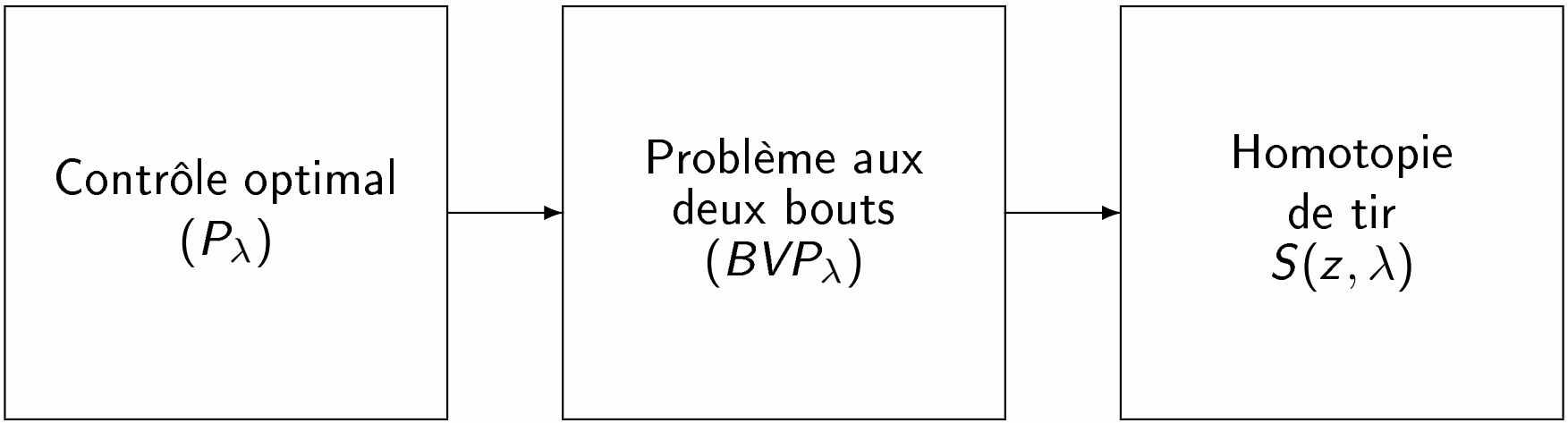
$$\begin{cases} h(z, 0) = G(z) \\ h(z, 1) = S(z) \end{cases}$$

- Le principe est alors de suivre un chemin menant d'un zéro de G à un zéro de S.

Cas du transfert orbital

Coût $J_\lambda(u) = \int_0^{t_f} (|u(t)| - (1-\lambda)(ln|u(t)| + ln(1-|u(t)|)))dt$

- Problème facile à résoudre pour $\lambda = 0$
- Pour $0 \leq \lambda < 1$, $u(x,p)$ est lisse.
- Problème de départ pour $\lambda = 1$



Existence du chemin

1. $\forall (z,\lambda) \in S^{-1}(0)$, $S'(z,\lambda)$ de rang plein ;
2. $\forall (z,0) \in S^{-1}(0)$, $\frac{\partial S}{\partial z}(z,0)$ de rang plein et $\forall (z,1) \in S^{-1}(0)$, $\frac{\partial S}{\partial z}(z,1)$ de rang plein.

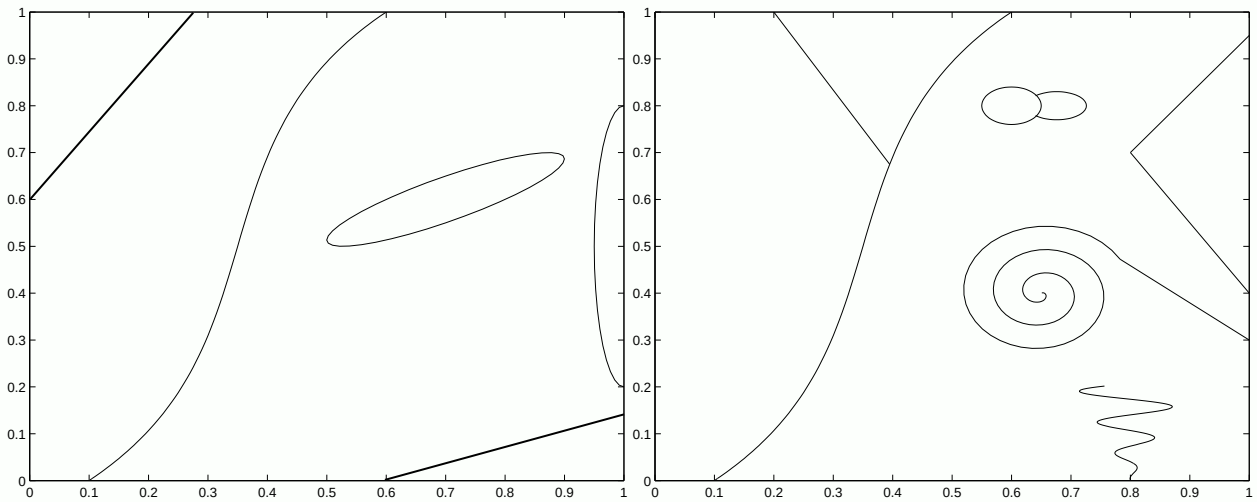


Fig.: *Chemins possibles (à gauche) et impossibles (à droite) (z est en abscisse et λ en ordonnée).*

Méthode homotopique simple

Suivi du chemin de zéros de l'homotopie $S(z, \lambda)$
Algorithme de "Newton global"

Initialisation z^0 solution de $S(z, 0) = 0$

$0 = \lambda_0 < \lambda_1 < \ldots < \lambda_n = 1$
Pour $i = 1, \ldots, n$ **faire**
Résoudre $S(z, \lambda_i) = 0$ par un algorithme de type Newton avec comme point de départ z^{i-1}

Comment choisir la suite $(\lambda_i)_i$?

Algorithmes d'intégration

Deux méthodes différentielles :

- methode PC, prediction (Euler) + Correction ;
- Suivi de chemin avec un intégrateur efficace (type Runge-Kutta).

Calcul du vecteur tangent

Si $c(s) = (z(s), \lambda(s))$ est une courbe lisse telle que

1. $c(0) = (z^0, 0)$
2. $S(c(s)) = 0$
3. $S'(c(s))$ de rang plein
4. $\dot{c}(s) \neq 0$

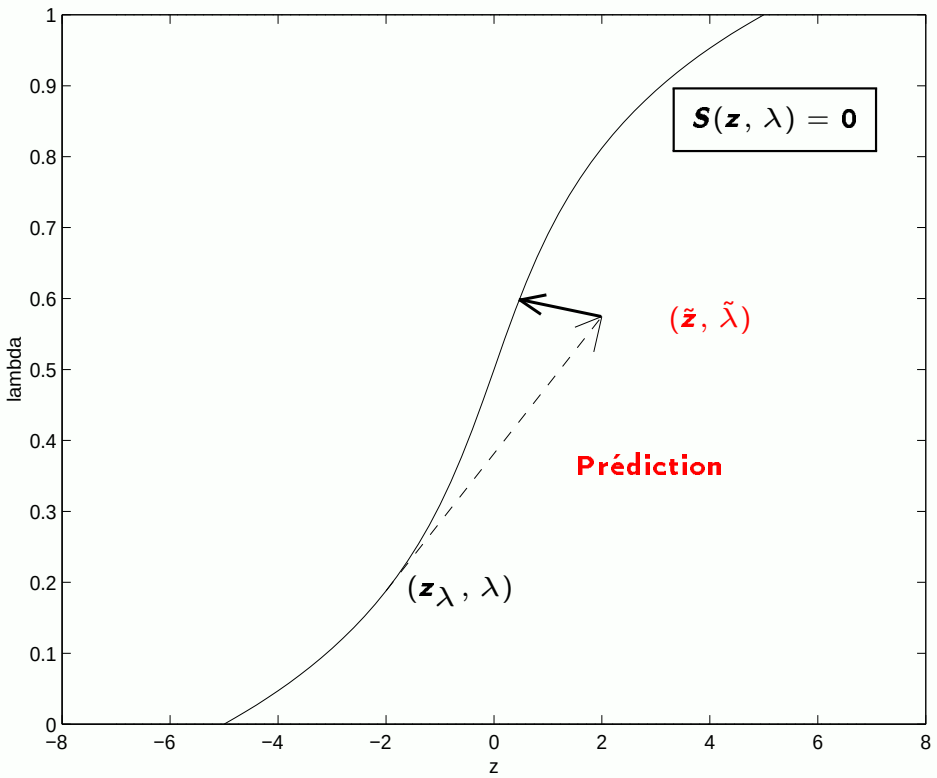
alors $\dot{c}(s) = k(S'(c(s)))$ est déterminé par

1. $S'(c(s))\dot{c}(s) = 0$
2. $|\dot{c}(s)|=1$
3. $\det \begin{pmatrix} S'(c(s)) \\ {}^t\dot{c}(s) \end{pmatrix}$ est de signe constant

Méthode Prédiction-correction

- Prédiction : schéma d'Euler

$$(\tilde{z}, \tilde{\lambda}) = (z_\lambda, \lambda) + k(S'(z_\lambda, \lambda)).\delta s$$



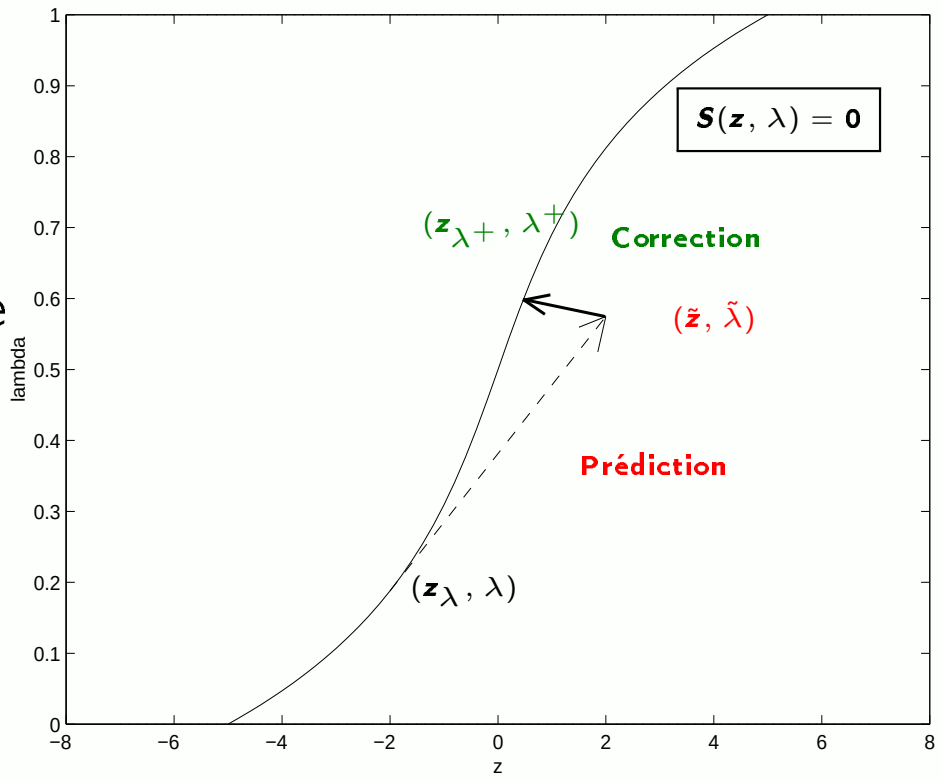
Méthode Prédiction-correction

- Prédiction : schéma d'Euler

$$(\tilde{z}, \tilde{\lambda}) = (z_\lambda, \lambda) + k(S'(z_\lambda, \lambda)).\delta s$$

- Correction : retour sur le chemin de zéros $(z_{\lambda^+}, \lambda^+)$

$$= \begin{cases} argmin\{|(z, \lambda) - (\tilde{z}, \tilde{\lambda})|^2\} \\ S(z, \lambda) = 0 \end{cases}$$



Méthode Prédiction-correction

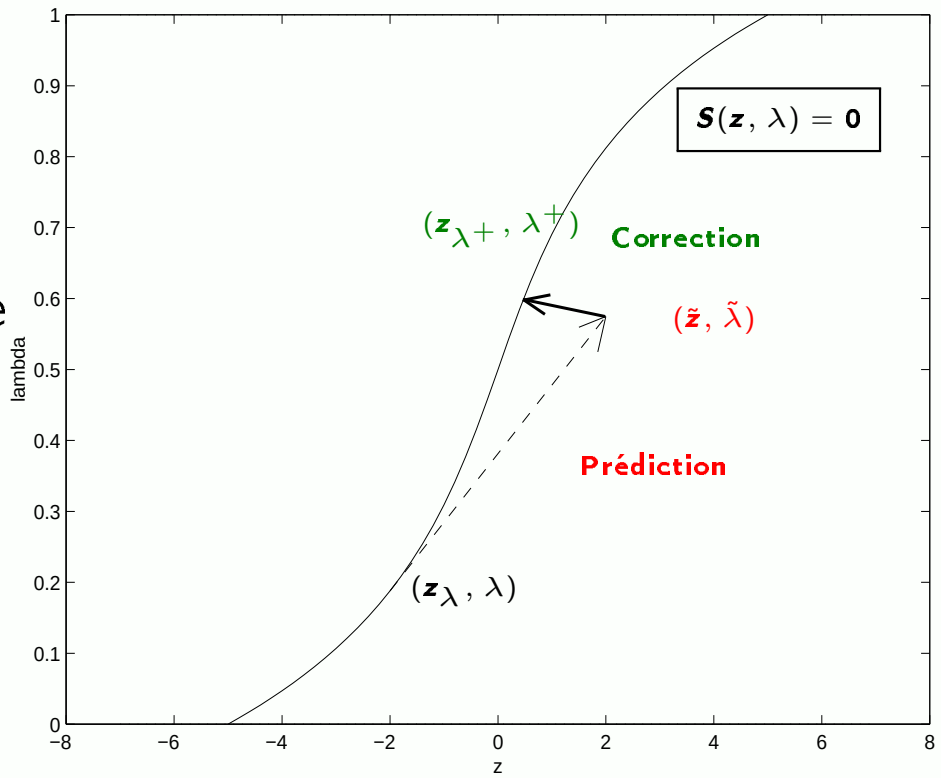
- Prédiction : schéma d'Euler

$$(\tilde{z}, \tilde{\lambda}) = (z_\lambda, \lambda) + k(S'(z_\lambda, \lambda)).\delta s$$

- Correction : retour sur le chemin de zéros $(z_{\lambda^+}, \lambda^+)$

$$= \begin{cases} argmin\{|(z, \lambda) - (\tilde{z}, \tilde{\lambda})|^2\} \\ S(z, \lambda) = 0 \end{cases}$$

- jusqu'à $\lambda = 1$.



Méthode d'intégration (Runge-Kutta)

Suivi du chemin de zéros $(c(s) = (z(s), \lambda(s)))$ de l'homotopie $S(z, \lambda)$

Initialisation z^0 solution de $S(z, 0) = 0$

Puis intégrer avec un intégrateur efficace :

$$(IVP) \begin{cases} \dot{c}(s) = k(S'(c(s))) \\ c(0) = (z^0, 0) \end{cases}$$

Jusqu'à s_f tel que $\lambda(s_f) = 1$. (besoin de le détecter).

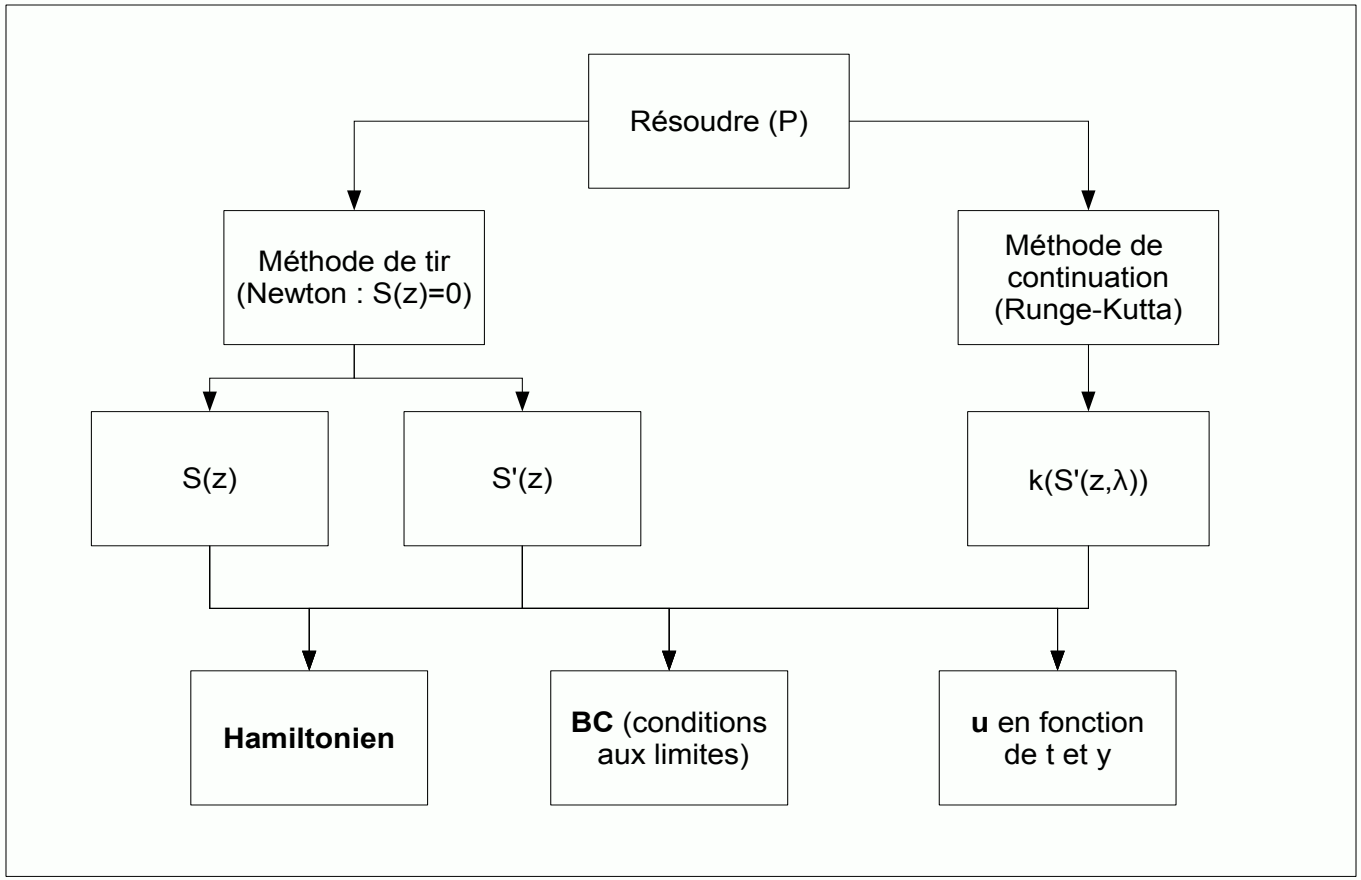
Démarche

- (Présent) Utilise une **méthode de tir** pour l'initialisation puis une **méthode homotopique** différentielle pour le suivi de chemin (en restant dans le cas lisse).

  $=>$ or dans le cas du transfert orbital, quand $\lambda = 1$ le contrôle est **discontinu**. On s'arrête donc à $\lambda_f < 1$ (approximation de la solution).

- (Futur) Interfacer avec un code spécifique (détection de commutations ...) pour obtenir la solution en $\lambda = 1$ (cas non lisse).

Organisation schématique



La méthode de tir

Trouver un zéro de $S_\lambda(z) = \begin{pmatrix} c_0(t_0, z) \\ c_f(t_f, y(t_f; z, \lambda)) \end{pmatrix}$ où $y(t_f; z, \lambda)$ est solution de :

$$(IVP_\lambda) \begin{cases} \dot{y}(t) = \varphi(t, y(t), \lambda) & \text{dans } [t_0, t_f] \\ y(t_0) = z & \text{valeur initiale} \end{cases}$$

- **dopri5** comme intégrateur (Runge-Kutta d'ordre (4)5).
- **hybrj** (librairie minpack) comme solveur. On lui fournit S_λ ainsi que S'_λ.
- S'_λ est calculé par **équations variationnelles** et différentiation automatique.
- On utilise **Tapenade** pour la différentiation automatique.

Calculer S'_λ

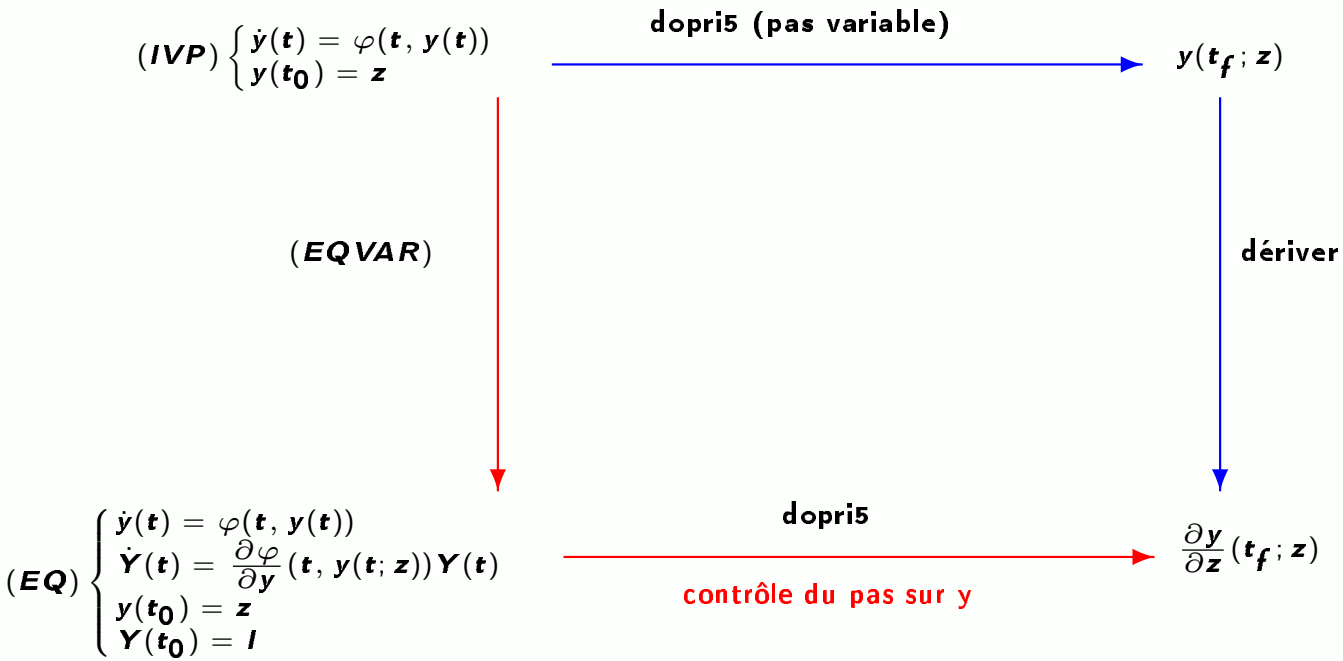
$$S_\lambda : z \longmapsto \begin{pmatrix} c_0(t_0, z) \\ c_f(t_f, y(t_f; z, \lambda)) \end{pmatrix} \qquad S'_\lambda(z(s)) = \begin{bmatrix} \frac{\partial c_0}{\partial z} \\ \frac{\partial c_f}{\partial y}\frac{\partial y}{\partial z} \end{bmatrix}$$

- $\frac{\partial c_f}{\partial y}$ et $\frac{\partial c_0}{\partial z}$ sont faciles à calculer.
- $\frac{\partial y}{\partial z}(t_f; z, \lambda)$ est solution du système de Cauchy (équations variationnelles) :

$$(EQ_1) \begin{cases} \dot{Y}(t) = \frac{\partial \varphi}{\partial y}(t, y(t; z, \lambda), \lambda) Y(t) \\ Y(t_0) = I_{2n} \end{cases}$$

- $\frac{\partial \varphi}{\partial y}$ et $\frac{\partial \varphi}{\partial \lambda}$ sont en général difficiles à calculer : on utilise la différentiation automatique (dérivée exacte).

Equation variationnelle et contrôle du pas



=> Meilleur numériquement.

La méthode de continuation

Intégrer :

$$(IVP)\begin{cases} \dot{c}(s) = k(S'(c(s))) \\ c(0) = (z^0, 0) \end{cases}$$

Jusqu'à $\lambda = 1$

- **dopri5** comme intégrateur.
- Sortie dense (interpolation d'Hermite de degré 3) pour détecter $\lambda = 1$.
- On utilise les **équations variationnelles** pour calculer $S'(c(s))$ et **Tapenade** pour la différentiation automatique.
- **dgeqrf** (librairie lapack) pour la factorisation QR.

Calculer $\dot{c}(s) = k(S'(c(s)))$

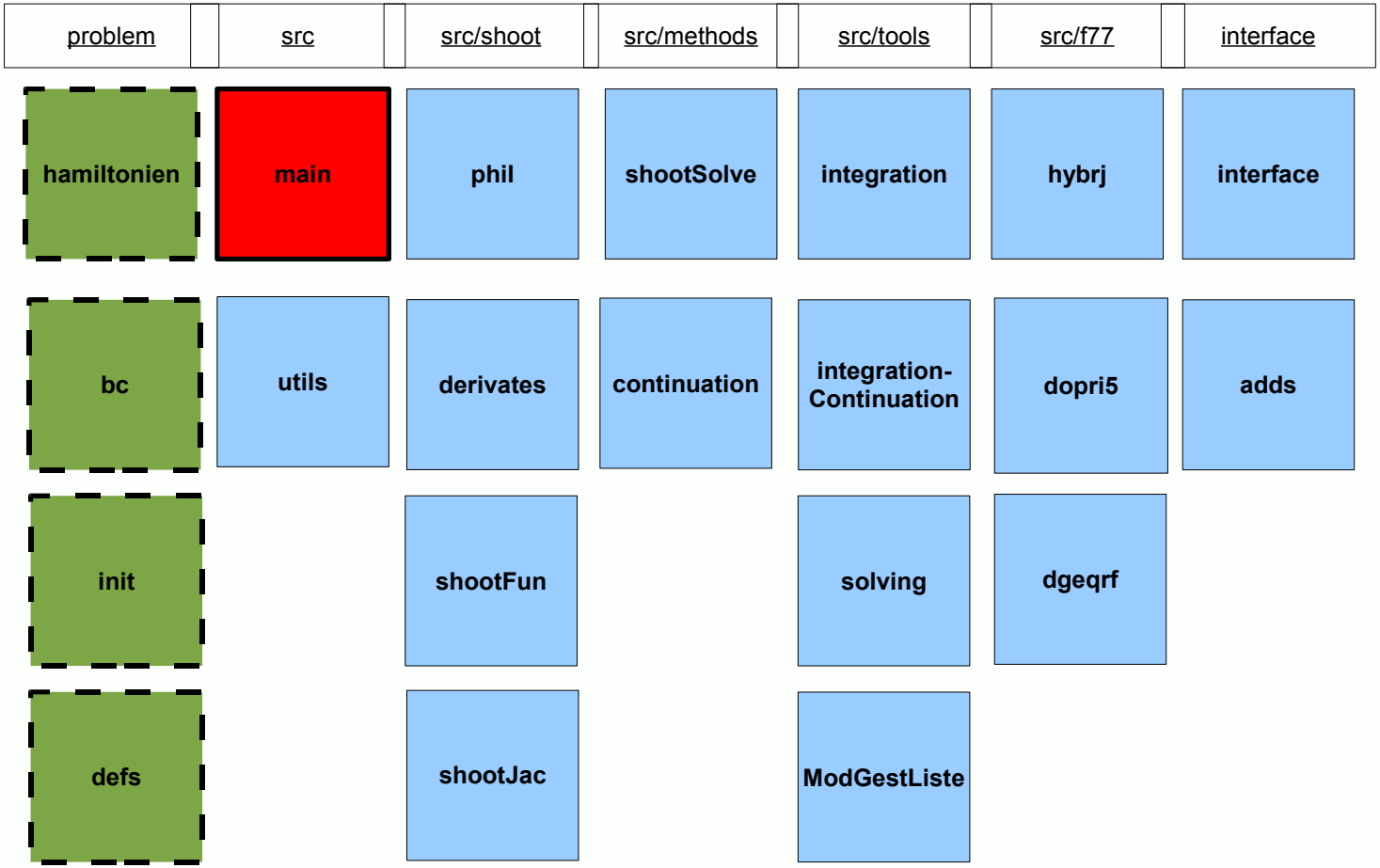
On sait que $S'(c(s))\dot{c}(s) = 0$ et $|\dot{c}(s)| = 1$

- factorisation QR : ${}^tS'(c(s)) = QR$
- $\dot{c}(s) = +/-$ dernière colonne de Q.

Enfin $\det \begin{pmatrix} S'(c(s)) \\ {}^t\dot{c}(s) \end{pmatrix} = \det Q \det R$ est de signe constant

- $\det Q = (-1)^p$ où p est le nombre de réflexions de Householder qui interviennent dans la factorisation.
- $\det R =$ le produit de ses éléments diagonaux (R est triangulaire).

Les modules ...


problem
src
src/shoot
src/methods
src/tools
src/f77
interface
hamiltonien
main
phil
shootSolve
integration
hybrj
interface
bc
utils
derivates
continuation
integration-
Continuation
dopri5
adds
init
shootFun
solving
dgeqrf
defs
shootJac
ModGestListe

... et ce qu'ils représentent

- **hamiltonien** :	fonction hamiltonienne
- **bc** :	conditions initiales et finales
- **init** :	paramètres du problème
- **defs** :	variables globales pour la résolution numérique
- **main** :	programme principal
- **utils** :	variables globales nécessaires au programme
- **phil** :	fonction phi (dérivée de l'hamiltonien)
- **derivates** :	dérivées partielles de phi (la plupart calculée par diff auto)
- **shootFun** :	fonction de tir
- **shootJac** :	dérivées de la fonction de tir
- **shootSolve** :	méthode de tir simple
- **continuation** :	méthode de suivi de chemin
- **integration** :	interface - méthodes d'intégration pour shootFun
- **integrationContinuation** :	interface - méthodes d'intégration pour la continuation
- **solving** :	interface - méthode de résolution de systèmes non linéaires
- **ModGestListe** :	méthodes de gestion de liste
- **hybrj** :	méthode de résolution de systèmes non linéaires
- **dopri5** :	méthode d'intégration (Runge-Kutta d'ordre 4(5))
- **dgeqrf** :	factorisation QR
- **interface** :	méthodes servant à l'affichage des résultats
- **adds** :	méthodes servant à la gestion des résultats

Transfert orbital

$$(P)\begin{cases} \min \int_0^{t_f} (|u(t)| - (1-\lambda)(ln|u(t)| + ln(1-|u(t)|)))dt \\ \dot{x} = f_0(x) + \frac{T_{max}}{m} \sum_{i=1}^{3} u_i f_i(x) \\ |u| \leq 1 \\ \text{Conditions initiales et finales} \\ t_f \text{ ou } L_f \text{ fixé} \end{cases}$$

avec $x = (P, e_x, e_y, h_x, h_y, L)$, $T_{max} = 10N$, t_f libre, L_f fixé, masse constante et comme conditions initiales et finales :

$P^0 = 11625km$	$P^f = 42165km$
$e_x^0 = 0.75$	$e_x^f = 0$
$e_y^0 = 0$	$e_y^f = 0$
$h_x^0 = 0.0612$	$h_x^f = 0$
$h_y^0 = 0$	$h_y^f = 0$
$L^0 = \pi$	$L^f = \text{fixé}$

Temps et précisision

- Temps de résolution : Un peu plus d'une heure. Le calcul du vecteur tangent est long et souvent fait (326 fois).
- Présicion : La norme de la fonction de tir est de l'ordre de 10^{-9} après la continuation puis 10^{-12} après un dernier tir.

Les bifurcations

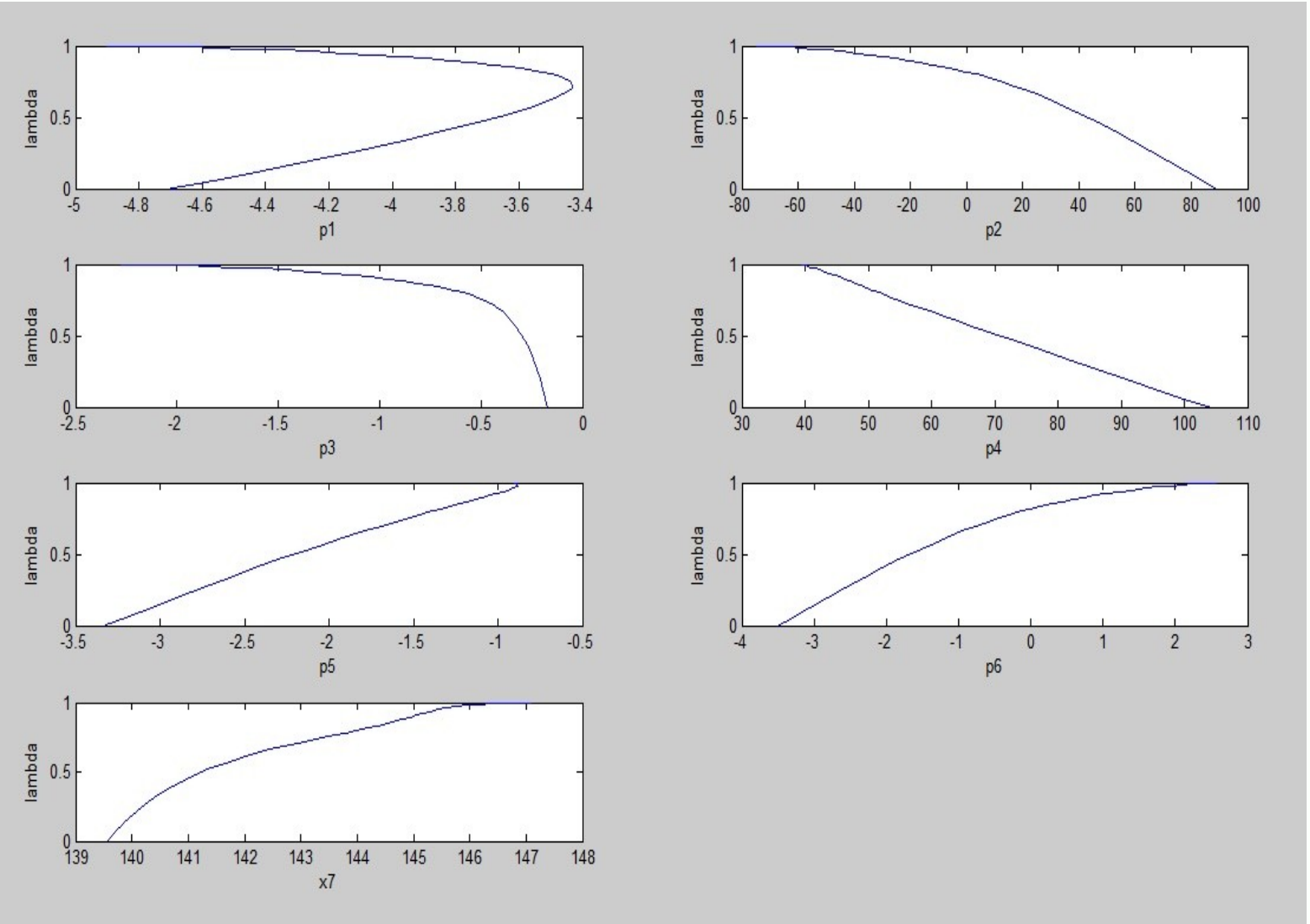
Pour avoir un chemin sans bifurcation le rang de S' doit être maximum en tout point. Donc une perte de rang indique une bifurcation.

Mais la probabilité de tomber sur le point de bifurcation est nulle, on ne peut pas utiliser ce critère.

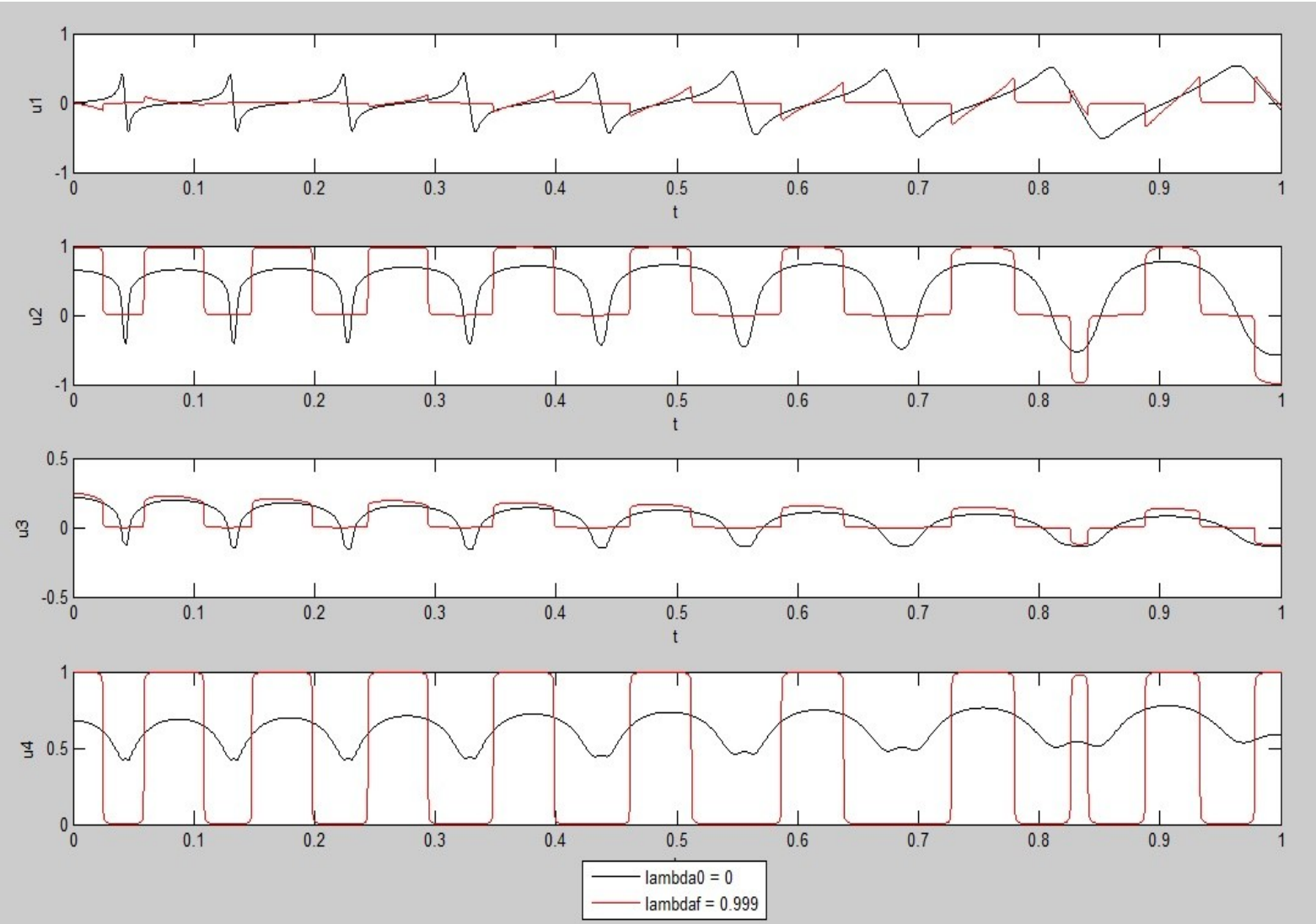
Or si le rang chute, $\det \begin{pmatrix} S'(c(s)) \\ {}^t\dot{c}(s) \end{pmatrix}$ change de signe.

$=>$ On détecte donc une bifurcation entre $\lambda = 0.8965$ et $\lambda = 0.9106$

Chemins



Contrôle



Conclusion

Evolution

- continuter les tests sur les deux problèmes ;
- interfaçage avec **Matlab** (pour l'instant en fortran 90) ;
- étude et comparaison des différentes methodes de suivi de chemin ;
- traiter les bifurcations éventuelles ;
- comparaison entre différences externes, équations variationnelles et autres (en terme de convergence ...) ;
- ...