

Modernisation de l'outil d'optimisation ENLSIP codé en Fortran77 avec le langage Julia

Pierre Borie

ENAC

15 septembre 2021

- 1 Présentation du projet
- 2 Étude des problèmes de moindres carrés
 - Moindres carrés pour la calibration de modèle
 - État de l'art
- 3 Travail d'implémentation
 - Stratégie d'implémentation et objectifs
 - Description de l'algorithme ENLSIP
- 4 Tests et résultats numériques
 - Comparaison avec Fortran
 - Comparaison avec un solveur moderne
- 5 Conclusion

Introduction

- Projet coopératif entre l'Université de Montréal (UdeM) et Hydro-Québec (HQ)
- Supervisé par :
 - ▶ Fabian Bastin, professeur au DIRO de l'UdeM et maître de stage
 - ▶ Stéphane Dellacherie, de l'unité Prévisions de contrôle du réseau d'Hydro-Québec TransÉnergie (HQT-PCR)
- Avec la collaboration d'Alain Marcotte et Olivier Milon (HQT-PCR)

- Société d'état québécoise (1944)
- Production d'électricité principalement via des barrages et centrales hydroélectriques
- Transport et distribution au Québec et vers des marchés d'exportation

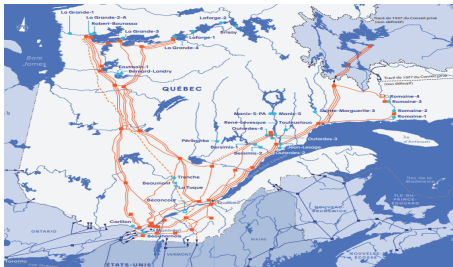


Figure – Réseau électrique d'Hydro-Québec

Source – <https://www.hydroquebec.com/transenergie/fr/>

Enjeux

Prédiction efficace nécessaire :

- Satisfaction de la demande
- Fiabilité
- Cadre sécuritaire du transport

Enjeux

Prédiction efficace nécessaire :

- Satisfaction de la demande
- Fiabilité
- Cadre sécuritaire du transport

→ Outils de prévision pour différents horizons temporels

Demande à court terme

- Modèle paramétrique
- Calibration à partir de données d'HQ (météorologiques, saisonnières, historiques de consommation...)
- Estimation réalisée par l'algorithme ENLSIP

ENLSIP

- Solveur de problèmes de moindres carrés sous contraintes d'égalité et d'inégalité
- Développé par Lindström et Wedin (1988)
- Codé en Fortran77
- Après 30 ans d'utilisation, maintenance et amélioration difficiles

Problématique

- Fiabilisation par l'implémentation dans un nouveau langage, Julia
- Modernisation en améliorant les techniques d'optimisation employées

- 1 Présentation du projet
- 2 Étude des problèmes de moindres carrés
 - Moindres carrés pour la calibration de modèle
 - État de l'art
- 3 Travail d'implémentation
 - Stratégie d'implémentation et objectifs
 - Description de l'algorithme ENLSIP
- 4 Tests et résultats numériques
 - Comparaison avec Fortran
 - Comparaison avec un solveur moderne
- 5 Conclusion

- 1 Présentation du projet
- 2 Étude des problèmes de moindres carrés
 - Moindres carrés pour la calibration de modèle
 - État de l'art
- 3 Travail d'implémentation
 - Stratégie d'implémentation et objectifs
 - Description de l'algorithme ENLSIP
- 4 Tests et résultats numériques
 - Comparaison avec Fortran
 - Comparaison avec un solveur moderne
- 5 Conclusion

Modélisation

- m données réelles (t_i, y_i)
- modèle $h(x, t)$ paramétré par $x \in \mathbb{R}^n$
- $r_i : x \longrightarrow |h(x, t_i) - y_i|$ écart entre prédiction et observation

Modélisation

- m données réelles (t_i, y_i)
- modèle $h(x, t)$ paramétré par $x \in \mathbb{R}^n$
- $r_i : x \longrightarrow |h(x, t_i) - y_i|$ écart entre prédiction et observation

Calibrer $h \iff$ minimiser $\sum_{i=1}^m r_i(x)^2$ par rapport à x

Modélisation avec contraintes

$$\left\{ \begin{array}{l} \min_{x \in \mathbb{R}^n} \frac{1}{2} \|r(x)\|^2 \\ \text{s.c.} \\ c_i(x) = 0, \quad \text{pour } i = 1, \dots, q \\ c_j(x) \geq 0, \quad \text{pour } j = q + 1, \dots, \ell \end{array} \right. \quad (1)$$

Avec $r = (r_1, \dots, r_m)^T$ multi-fonction des résidus

Le cas linéaire

$$\min_x \frac{1}{2} \|Ax - y\|^2 \quad (2)$$

Le cas linéaire

$$\min_x \frac{1}{2} \|Ax - y\|^2 \quad (2)$$

- Prépondérant en régression linéaire pour l'apprentissage statistique (Hastie *et al.*, 2009)

Le cas linéaire

$$\min_x \frac{1}{2} \|Ax - y\|^2 \quad (2)$$

- Prépondérant en régression linéaire pour l'apprentissage statistique (Hastie *et al.*, 2009)
- Structure avantageuse :
 - ▶ Convexité
 - ▶ Systèmes linéaires
 - ▶ Décompositions matricielles (QR, Cholesky)
 - ▶ Caractérisation de la solution (équations normales)

$$\boxed{A^T A x^* = A^T y} \quad (3)$$

Cas non linéaire

- Beaucoup de problèmes non linéaires
- Méthodes itératives avec calcul de direction de descente et longueur de pas :

$$x_{k+1} = x_k + \alpha_k p_k$$

- Approximation des résidus

Méthode de Gauss–Newton

Linéarisation des résidus :

$$r(x + p) \approx J(x)p + r(x) \quad (4)$$

Avec $J(x) = \left[\frac{\partial r_i}{\partial x_j}(x) \right]_{(i,j)}$ matrice jacobienne des résidus

Sous problème Gauss–Newton

À chaque itération on résout :

$$\min_p \frac{1}{2} \|J(x_k)p + r(x_k)\|^2 \quad (5)$$

- Se ramène au cas linéaire (2) avec $A = J(x_k)$, $x = p$ et $y = -r(x_k)$
- Bonne approximation en convergence

Autres familles de méthodes

- Levenberg–Marquardt :
 - ▶ Semblable à Gauss–Newton
 - ▶ Directions de descente potentielles restreintes à un certain ensemble
 $E_k = \{p, \|D_k p\| \leq \Delta_k\}$
 - ▶ D_k est une matrice diagonale,

Autres familles de méthodes

- Levenberg–Marquardt :
 - ▶ Semblable à Gauss–Newton
 - ▶ Directions de descente potentielles restreintes à un certain ensemble $E_k = \{p, \|D_k p\| \leq \Delta_k\}$
 - ▶ D_k est une matrice diagonale,
- Régions de confiance :
 - ▶ Approximation des résidus à chaque itération dans un voisinage de x_k
 - ▶ $\mathcal{B}_k = \{p \in \mathbb{R}^n, \|x_k + p\|_k \leq \Delta_k\}$
 - ▶ Δ_k réel positif
 - ▶ Revient à minimiser des fonctions plus simples (polynômes, fonctions linéaires)

Bilan de l'état de l'art

- Cas linéaire avantageux
- Approximation de la fonction objectif pour s'y ramener
- Peu de travaux sur des problèmes contraints (Gill *et al.*, 1985)

- 1 Présentation du projet
- 2 Étude des problèmes de moindres carrés
 - Moindres carrés pour la calibration de modèle
 - État de l'art
- 3 Travail d'implémentation
 - Stratégie d'implémentation et objectifs
 - Description de l'algorithme ENLSIP
- 4 Tests et résultats numériques
 - Comparaison avec Fortran
 - Comparaison avec un solveur moderne
- 5 Conclusion

Stratégie d'implémentation

- Développement incrémental des fonctionnalités
- Analyse du code source et du travail des auteurs (Lindström et Wedin, 1988)

Objectifs

- Implémenter ENLSIP en Julia intégralement
- Obtenir des résultats identiques en Julia et en Fortran
- Documenter la méthode de résolution utilisée
- Transmettre le travail dans l'infonuagique Hydro-Québec en vue de sa validation puis de son intégration en production

Algorithme itératif avec longueur de pas

Require: $x_0 :=$ point initial

$k \leftarrow 0$

Initialisation de l'ensemble actif

repeat

 Mise à jour des contraintes actives

 Calcul de la direction de descente p_k

 Calcul de la longueur de pas α_k

$x_{k+1} \leftarrow x_k + \alpha_k p_k$

$k \leftarrow k + 1$

until critère d'arrêt

return x_k

Gestion des contraintes

Inspirée de la stratégie Equality Quadratic Programming (Gill *et al.*, 1985) :

- Exploitation de la convergence vers un point KKT
- Estimation progressive de l'ensemble actif à la solution
- Restriction à des égalités pour calculer la direction de descente

Restriction du nombre de contraintes

Suite à estimation des multiplicateurs de Lagrange λ_k

- Évaluation des inégalités actives

Restriction du nombre de contraintes

Suite à estimation des multiplicateurs de Lagrange λ_k

- Évaluation des inégalités actives
- Contraintes d'inégalité actives \rightarrow contraintes d'égalité

Restriction du nombre de contraintes

Suite à estimation des multiplicateurs de Lagrange λ_k

- Évaluation des inégalités actives
- Contraintes d'inégalité actives \rightarrow contraintes d'égalité
- Contraintes d'inégalité inactives mises de côté pour calculer la direction de descente

Calcul de la direction de descente

Méthode de Gauss–Newton sous contraintes :

$$\left\{ \begin{array}{l} \min_{p \in \mathbb{R}^n} \frac{1}{2} \|r(x_k + p)\|^2 \\ \text{s.c.} \\ c(x_k + p) = 0 \end{array} \right. \quad (6)$$

Modélisation du sous-problème de l'itération k

Linéarisations des résidus et contraintes :

$$\begin{aligned} r(x_k + p) &\approx Jp + r(x_k) \\ c(x_k + p) &\approx Ap + c(x_k) \end{aligned} \tag{7}$$

J et A matrices jacobiennes de r et c en x_k .

Modélisation du sous-problème de l'itération k

Linéarisations des résidus et contraintes :

$$\begin{aligned} r(x_k + p) &\approx Jp + r(x_k) \\ c(x_k + p) &\approx Ap + c(x_k) \end{aligned} \tag{7}$$

J et A matrices jacobiennes de r et c en x_k .

$$\text{Problème initial (6)} \iff \begin{cases} \min_{p \in \mathbb{R}^n} \frac{1}{2} \|Jp + r(x_k)\|^2 \\ \text{s.c.} \\ Ap + c(x_k) = 0 \end{cases}$$

Résolution et implémentation

- Factorisations matricielles QR
- Résolution de systèmes linéaires triangulaires successifs
- Facilité par l'utilisation du Julia

Fonction de mérite pour le calcul du pas

Fonction de mérite :

$$\Psi : (x, w) \mapsto \frac{1}{2} \|r(x)\|^2 + \sum_{i \in \mathcal{W}} w_i c_i(x)^2 + \sum_{i \in \mathcal{I}} w_i \min(0, c_i(x))^2 \quad (8)$$

Avec \mathcal{W} prédiction de l'ensemble actif et \mathcal{I} contraintes inactives

$w \in \mathbb{R}^I$ vecteur des pénalités

- Calculées spécifiquement
- Orientent la recherche linéaire

Calcul du pas

Une fois p_k et w_k calculés :

$$\alpha_k = \min_{\alpha} \phi(\alpha) = \Psi(x_k + \alpha p_k, w_k) \quad (9)$$

- Minimisations successives d'approximations polynomiales de ϕ
- Ressemblance avec régions de confiance

Calcul du pas

Une fois p_k et w_k calculés :

$$\alpha_k = \min_{\alpha} \phi(\alpha) = \Psi(x_k + \alpha p_k, w_k) \quad (9)$$

- Minimisations successives d'approximations polynomiales de ϕ
- Ressemblance avec régions de confiance
- Fonctionnalité longue à implémenter
- Documentation restreinte au cas sans contraintes d'inégalité (Lindström et Wedin, 1984)
- *Reverse engineering* avec la version Fortran

Bilan sur l'implémentation

- Fidélité au code en Fortran
- Aspects théoriques difficiles à extraire
 - ▶ convergence
 - ▶ calcul des pénalités
 - ▶ présence d'heuristiques ou non
- Stratégie de travail payante

- 1 Présentation du projet
- 2 Étude des problèmes de moindres carrés
 - Moindres carrés pour la calibration de modèle
 - État de l'art
- 3 Travail d'implémentation
 - Stratégie d'implémentation et objectifs
 - Description de l'algorithme ENLSIP
- 4 Tests et résultats numériques
 - Comparaison avec Fortran
 - Comparaison avec un solveur moderne
- 5 Conclusion

Stratégie de objectifs

- Tests sur problèmes "jouet" documentés et issus de la littérature (Hock et Schittkowski, 1980)
- Vérifier la concordance des versions Fortran et Julia
- Comparer les performances le solveur IPOPT

Premier problème (n°65 de Hock et Schittkowski (1980))

$$\left\{ \begin{array}{l} \min_{x_1, x_2, x_3} (x_1 - x_2)^2 + \frac{(x_1 + x_2 - 10)^2}{9} + (x_3 - 5)^2 \\ \text{s.c.} \\ 48 - x_1^2 - x_2^2 - x_3^2 \geq 0 \\ -4.5 \leq x_i \leq 4.5, \text{ pour } i = 1, 2 \\ -5 \leq x_3 \leq 5 \end{array} \right. \quad (10)$$

Résultats

Point initial $x_0 = (-5, 5, 0)^T$

Solution attendue $x^* = (3.650461821, 3.65046168, 4.6204170507)^T$

Fonction objectif à la solution $f(x^*) = 0.9535288567$

Solveur	Itérations	Temps de calcul	Objectif
ENLSIP-Fortran77	11	–	9.53529000
ENLSIP-Julia	11	1.585 sec	9.53528856

Table – Comparaison des versions Julia et Fortran77 d'ENLSIP

- Concordance des résultats
- Même nombre d'itérations

Second problème (n°57 de Hock et Schittkowski (1980))

$$\left\{ \begin{array}{l} \min_{x_1, x_2} \sum_{i=1}^{44} f_i(x_1, x_2)^2 \\ \text{s.c.} \\ 0.49x_2 - x_1x_2 - 0.09 \geq 0 \\ x_1 \geq 0.4, \quad x_2 \geq -4 \end{array} \right. \quad (11)$$

- Avec $f_i(x_1, x_2) = b_i - x_1 - (0.49 - x_1)e^{-x_2(a_i-8)}$, ($i = 1, \dots, 44$)
- Couples (a_i, b_i) donnés
- S'apparente à la calibration du modèle :

$$h : (x_1, x_2, a) \longrightarrow x_1 + (0.49 - x_1)e^{-x_2(a-8)}$$

Résultats

Point initial $x_0 = (0.42, 5)^T$

Solution attendue $x^* = (0.419952675, 1.284845629)^T$

Fonction objectif à la solution $f(x^*) = 0.02845966972$

Solveur	Itérations	Temps de calcul	Objectif
ENLSIP-Julia	5	1.663 sec	2.845966972e-02
IPOPT	24	2.003 sec	2.845966907e-02

Table – Comparaison des performances d'ENLSIP-Julia avec IPOPT

Résultats

Point initial $x_0 = (0.42, 5)^T$

Solution attendue $x^* = (0.419952675, 1.284845629)^T$

Fonction objectif à la solution $f(x^*) = 0.02845966972$

Solveur	Itérations	Temps de calcul	Objectif
ENLSIP-Julia	5	1.663 sec	2.845966972e-02
IPOPT	24	2.003 sec	2.845966907e-02

Table – Comparaison des performances d'ENLSIP-Julia avec IPOPT

- Résultats comparables entre les deux solveurs
- ENLSIP plus adapté aux problèmes de calibration
- Temps d'exécution peu pertinent avec ces dimensions

Bilan des tests

- ENLSIP-Julia capable de résoudre des problèmes de moindres carrés contraints
- Résultats satisfaisants
- Très bonne correspondance avec la version Fortran

- 1 Présentation du projet
- 2 Étude des problèmes de moindres carrés
 - Moindres carrés pour la calibration de modèle
 - État de l'art
- 3 Travail d'implémentation
 - Stratégie d'implémentation et objectifs
 - Description de l'algorithme ENLSIP
- 4 Tests et résultats numériques
 - Comparaison avec Fortran
 - Comparaison avec un solveur moderne
- 5 Conclusion

Conclusion

Bilan du travail réalisé :

- Fiabilisation et compréhension d'ENLSIP bien avancées
- Résultats satisfaisants sur des problèmes simples

Conclusion

Bilan du travail réalisé :

- Fiabilisation et compréhension d'ENLSIP bien avancées
- Résultats satisfaisants sur des problèmes simples

Poursuite du projet :

- Compléter la documentation
- Complexification des tests
 - ▶ Élaboration d'une interface Fortran-Julia
 - ▶ Utilisation de données et modèles d'Hydro-Québec
- Modernisation de la méthode
 - ▶ Points intérieurs
 - ▶ Approche primal-dual

Bibliographie

- P.E. Gill, W. Murray, M.A. Saunders et M.H. Wright : Model building and practical aspects of non linear programming. *Computational Mathematical Programming*, F15:210–247, 1985.
- T. Hastie, R. Tibshirani et J. Friedman : *The Elements of Statistical Learning : Data Mining, Inference and Prediction*. Springer Series in Statistics. Springer, seconde édition, 2009.
- W. Hock et K. Schittkowski : *Test Examples for Nonlinear Programming Codes*, volume 187 de *Lecture Notes in Economics and Mathematical Systems*. Springer, seconde édition, 1980.
- P. Lindström et P.Å. Wedin : A new linesearch algorithm for nonlinear least squares problems. *Mathematical Programming*, Vol. 29:268–296, 1984.
- P. Lindström et P.Å. Wedin : Gauss-Newton based algorithms for constrained nonlinear least squares problems. *Institute of Information processing, University of Umeå, Sweden*, 1988.