



## SOLUTION ROBOTIQUE SOFTWARE



# ProxSuite

THE ADVANCED PROXIMAL OPTIMIZATION TOOLBOX

*Inria*

ProxSuite propose une famille de solveurs open source fondés sur des méthodes proximales primal-dual revisitées, couvrant LP et QP avec une attention particulière à la robustesse numérique et à l'efficacité.

Grâce à une API unifiée et à des backends denses, sparses et matrix-free, la bibliothèque exploite finement la structure des problèmes, offre des options avancées de warm-start et traite aussi des QP non convexes. ProxSuite propose aussi des solutions efficaces pour différencier à travers les problèmes QPs, et est facilement interfaçable avec les bibliothèques modernes d'apprentissage profond.

Testée de manière extensive, elle affiche des performances de pointe sur les benchmarks du domaine et s'intègre nativement avec CVXPY et CasADi (ProxQP).

### DOMAINES D'APPLICATION

Robotique de locomotion et de manipulation, Modélisation et simulation, Contrôle de systèmes robotiques, et bien plus.

#### A quoi cela sert ?

- Résoudre des LP/QP convexes et des QP non convexes avec robustesse numérique.
- Servir de sous-routine haute fréquence pour le contrôle optimal sous contrainte (MPC), la dynamique inverse et l'estimation.
- Étudier et prototyper des algorithmes proximaux primal-duaux pour LP/QP dans un cadre reproductible.
- Accélérer les résolutions séquentielles (warm-start) en boucles d'optimisation répétées.
- Fournir un solveur robuste pour des dimensions élevées (MPC) avec des exigences de précision variables.

#### Exemple de cas d'usage

- Robotique humanoïde et quadrupède: locomotion multi-contacts, suivi de trajectoires sous contraintes.
- Manipulation dextre: contrôle sous contraintes (limites articulaires/efforts/espace de travail), replanification rapide.
- Au-delà: contrôle embarqué temps réel (automobile, drones), finance (portefeuilles), production/énergie (QP opérationnels), apprentissage (SVM et sous-problèmes QP), problèmes inverses (imagerie).

### LES +

- **Pratique** : API unifiée permettant d'utiliser différents backend (denses, sparses, matrix-free)
- **Mise en oeuvre** : Réduction de la charge de réglage et robustesse numérique grâce à une approche basée sur Lagrangien augmenté et méthodes proximales primal-duales.
- **Efficace** : Exploitation de la structure creuse des matrices pour une meilleure efficacité.
- **Robuste** : Détection d'infaisabilité et calcul du QP réalisable le plus proche en contraintes pour diagnostics et reprise.

### Environnement & exigences techniques

- Header-only, thread-safe
- Logiciel C++, interfaçable avec Eigen et Python sous Licence BSD-2
- Multi-plateformes (Linux, MacOS, Windows) et plus sur demande
- Chaîne de build CMake / Pixi et intégration continue sur Github

### MOTS CLES

Solveurs numériques, Solveurs différentiables  
Optimisation convexe, LP, QP, Robotique,  
Planification de trajectoires sous contraintes

### MATURITE TECHNOLOGIQUE

- TRL 6-8
- Utilisée en R&D et en production
- Consortium Open-Source en cours de création

### ACCES OPEN-SOURCE

<https://github.com/Simple-Robotics/ProxSuite>

### OPPORTUNITES DE COLLABORATION

- ✓ Preuves de concept, adaptations sur-mesure
- ✓ Co-développement ou transfert technologique
- ✓ Collaboration via projets collaboratifs

### CONTACTS

Justin Carpentier  
Willow, Inria de Paris  
[justin.carpentier@inria.fr](mailto:justin.carpentier@inria.fr)

Pierre-Guillaume Raverdy  
Willow/SED, Inria de Paris  
[pierre-guillaume.raverdy@inria.fr](mailto:pierre-guillaume.raverdy@inria.fr)