

Régions de confiance avec Pas de Cauchy

Implémentation

1. Coder l'algorithme du pas de Cauchy dans le fichier `src/cauchy.jl`. La spécification de cet algorithme est donnée dans le fichier.
2. Ecrire des tests exhaustifs (qui testent tous les cas de figure possibles) pour votre algorithme du pas de Cauchy. Vous remplirez pour cela le fichier `test/tester_cauchy.jl` sur le modèle des autres fichiers de tests et vous exécuterez dans la cellule de code ci-après ces tests.

```
include("../src/cauchy.jl")           # votre algorithme
include("../test/tester_cauchy.jl")  # la fonction pour tester votre
algorithme

#
tester_cauchy(cauchy); # tester l'algorithme
```

1. Coder l'algorithme des régions de confiance (fichier `src/regions_de_confiance.jl`). Sa spécification est donnée dans le fichier.
2. Vérifier que les tests ci-dessous passent.

```
include("../src/regions_de_confiance.jl")
include("../test/tester_rc_cauchy.jl")

#
afficher = false # si true, alors affiche les résultats des
algorithmes

#
tester_rc_cauchy(regions_de_confiance, afficher);
```

Interprétation

1. Soit la fonction $f_1: R^3 \rightarrow R$ définie par

$$f_1(x_1, x_2, x_3) = 2(x_1 + x_2 + x_3 - 3)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^2$$

Quelle relation lie la fonction f_1 et son modèle de Taylor à l'ordre 2 ? Comparer alors les performances de l'algorithme de Newton et celui des régions de confiance avec le pas de Cauchy sur cette fonction.

2. Le rayon initial de la région de confiance est un paramètre important dans l'analyse de la performance de l'algorithme. Sur quel(s) autre(s) paramètre(s) peut-on jouer pour essayer d'améliorer cette performance ? Étudier l'influence d'au moins deux de ces paramètres. Pour cela vous ferez des tests numériques et donnerez les résultats sous forme de tableaux et de graphiques.

```
# Expérimentations numériques à faire ici
# Vous pouvez utiliser le package Plots pour les affichages de
# courbes: using Plots
```

Régions de confiance avec gradient conjugué tronqué

Implémentation

1. Implémenter l'algorithme du gradient conjugué tronqué (fichier `src/gct.jl`). Sa spécification est dans le fichier.
2. Vérifier que les tests ci-dessous passent.

```
include("../src/gct.jl")
include("../test/tester_gct.jl")

#
tester_gct(gct);
```

1. Intégrer l'algorithme du gradient conjugué tronqué dans le code des régions de confiance.
2. Vérifier que les tests ci-dessous passent.

```
include("../src/regions_de_confiance.jl")
include("../test/tester_rc_gct.jl")

#
afficher = false # si true, alors affiche les résultats des
algorithmes

#
tester_rc_gct(regions_de_confiance, afficher);
```

Interprétation

Nous proposons de comparer l'utilisation du pas de Cauchy avec celle du gradient conjugué tronqué dans l'algorithme des régions de confiance.

Remarques.

- Nous vous demandons de réaliser des expérimentations numériques pour les comparaisons demandées ci-après.
- Vous devez utiliser l'argument optionnel `max_iter_gct` et la sortie `xs` de l'algorithme des régions de confiance.
- Vous pouvez comparer l'écart en norme entre les itérés de l'algorithme et la solution du problème.

- Vous trouverez des choses utiles dans le fichier `test/fonctions_de_tests.jl`.
1. Comparer dans le cas où l'on force le gradient conjugué tronqué à ne faire qu'une seule itération. Que remarquez vous ?
 2. Comparer dans le cas général. Que remarquez vous ?
 3. Quels sont les avantages et inconvénients des deux approches ?

Expérimentations numériques à faire ici.
Vous pouvez utiliser le package Plots pour les affichages de courbes: using Plots