

# Notice fonctionnelle simplifiée de gpac

## Gradient à Pas Adaptatif avec Corrections

Groupe Problèmes Inverses

Laboratoire des Signaux et Systèmes (CNRS – Supélec – UPS)  
Supélec, Plateau de Moulon, 91192 Gif-sur-Yvette Cedex, France

L'algorithme décrit ci-après est un algorithme de descente du premier ordre à pas adaptatif. Il est particulièrement adapté à la minimisation de fonctions multivariées d'un grand nombre de variables. Il bénéficie de l'expérience du Groupe Problèmes Inverses (Laboratoire des Signaux et Systèmes, CNRS – Supélec – UPS) dans l'utilisation de tels algorithmes pour la résolution de problèmes inverses. Différentes directions de descente et techniques d'adaptation du pas sont disponibles.

### 1 Principes généraux

Le cadre est celui de l'optimisation numérique : il s'agit de trouver le vecteur  $\hat{\mathbf{x}}$  minimisant une fonction  $f$ , à valeur réelle, appelée critère :

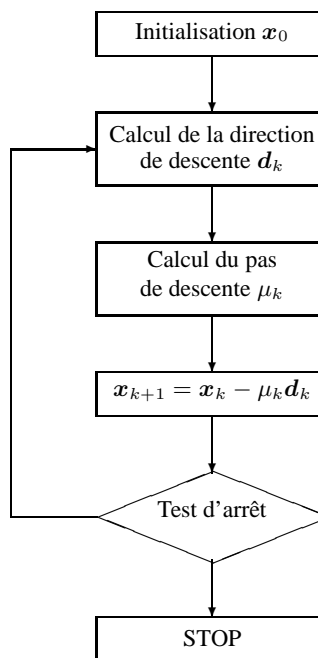
$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} f(\mathbf{x}).$$

L'algorithme retenu entre dans la classe des algorithmes de descente du premier ordre à pas adaptatif. A partir d'une solution initiale  $\mathbf{x}_0$ , il procède de façon itérative en exploitant la récurrence générique :

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mu_k \mathbf{d}_k$$

où  $\mathbf{d}_k$  est la direction de descente et  $\mu_k$  le pas de descente. Un organigramme de l'algorithme est représenté sur la figure ci-contre.

Le déroulement de l'algorithme repose en particulier sur (1) la direction de descente  $\mathbf{d}_k$ , (2) l'adaptation du pas  $\mu_k$  ainsi que (3) les conditions d'arrêt de l'algorithme. Tous ces choix sont réglables par l'intermédiaire d'un vecteur d'options passé au logiciel.



#### 1.1 Direction de descente

La direction de descente la plus simple est celle du gradient du critère : c'est la première stratégie proposée par le logiciel. Ce choix peut cependant s'avérer inefficace (notamment lorsque les iso-critères forment des vallées très étroites) et diverses corrections de direction ont été proposées dans la littérature [1, 2, 3]. Les corrections envisagées ici reposent sur des informations du premier ordre : elles s'appuient sur le gradient et sur les directions précédemment empruntées. Les corrections proposées dans ce logiciel sont :

1. la *correction de Vignes* (demi-somme du gradient et de la direction de descente précédente dans le cas d'un angle obtu entre ces directions) ;
2. la *correction de la bissectrice* (bissectrice du gradient et de la direction de descente précédente dans le cas d'un angle obtu entre ces directions) ;
3. le *gradient conjugué de Polak-Ribière*, conjugant le gradient et les directions de descente précédentes.

## 1.2 Adaptation du pas de descente

Le choix du pas de descente peut s'envisager par la minimisation du critère dans la direction de descente choisie :

$$\mu_k = \arg \min_{\mu} f(\mathbf{x}_k - \mu \mathbf{d}_k).$$

Le nouveau problème est alors un problème de minimisation monodimensionnel et plusieurs options sont possibles [1, Ch. 3]. La première option disponible dans le logiciel repose sur une technique de recherche par *dichotomie*. Deux autres options reposant sur des approximations polynomiales du critère sont également disponibles :

- a. une *approximation quadratique* à partir de la valeur du critère en deux points et de la valeur du gradient en l'un de ces deux points ;
- b. une *approximation cubique* qui nécessite la valeur du critère en un point supplémentaire.

La valeur du pas est alors choisie pour minimiser une de ces approximations. En pratique le logiciel effectue automatiquement l'adaptation du pas en combinant les deux approximations et la technique de dichotomie. En particulier, si les points sont trop proches les approximations polynomiales peuvent se révéler instables : le logiciel commute alors automatiquement sur la technique de dichotomie.

## 1.3 Conditions d'arrêt

Plusieurs conditions d'arrêt sont envisageables dans un tel algorithme itératif. Nous avons retenu une condition simultanée sur l'évolution du critère  $f(\mathbf{x}_k) - f(\mathbf{x}_{k-1})$  et sur la distance parcourue  $\|\mathbf{x}_k - \mathbf{x}_{k-1}\|$  (différents types de norme sont disponibles).

## 2 Codage *Matlab* et syntaxe d'appel

Le codage est réalisé à l'aide du logiciel de calcul *Matlab*, qui fonctionne sur diverses plates-formes (Linux, Unix, DOS, OS2, Mac). Le code contient environ 500 lignes réparties en trois fonctions *Matlab*.

1. `gpac` : c'est la fonction principale qui initialise l'algorithme et pilote les itérations jusqu'à ce que la condition d'arrêt soit vérifiée. Elle fait appel aux fonctions de calcul de direction et du pas.
2. `ChoixDir` définit la nouvelle direction de descente avec les éventuelles corrections ;
3. `Min1D` calcule les pas dans la direction choisie.

Dans un souci de compatibilité et de transparence, les syntaxes d'appel de la fonction `gpac` sont fortement inspirées des syntaxes d'appel des fonctions de l'*Optimization Toolbox* de *Matlab*. La forme d'appel standard est la suivante :

```
[X,Options,Histo] = gpac('fun',X0,Options,'grad',p1,p2,...)
```

La fonction possède une aide en ligne accessible par la commande `help gpac`.

## Références

- [1] J. Nocedal et S. J. Wright, *Numerical Optimization*, Series in Operations Research. Springer Verlag, New York, 2000.
- [2] D. P. Bertsekas, *Nonlinear programming*, Athena Scientific, Belmont, MA, USA, 2nd edition, 1999.
- [3] E. Walter et L. Pronzato, *Identification de modèles paramétriques à partir de données expérimentales*, Masson, Paris, 1994.