FACULTE DES SCIENCES BEN M'SICK
UNIVERSITE HASSAN II DE CASABLANCA

Algorithmes d’optimisation

Pr. Faouzia Benabbou (faouzia.benabbou@univh2c.ma)
Département de mathématiques et Informatique

Master Data Science & Big Data

2024-2025

Plan du Module: Algorithmes d’optimisation

- 01 Introduction
- 02 Rappels mathématiques
- 03 Algorithmes d’Optimisation Sans Contraintes
- 04 Algorithmes d’Optimisation avec Contraintes
- 05 Programmation Linéaire et non Linéaire
- 06 Optimisation combinatoire et globale
- 07 Méthodes heuristiques et métaheuristiques

Les algorithmes d’optimisation

Classes des algorithmes

Optimisation

Sans contraintes

Méthodes Analytiques

Méthodes numériques

- Méthode sans gradient
- Méthode de descente de gradient:
  - ✓ Méthode de gradient à pas fixe et optimal
  - ✓ Méthode de gradient conjugué
  - ✓ Méthode de Newton
  - ✓ Méthodes du gradient quasi-newton

Sous contrainte

- Méthodes générales de résolutions de PCE, PCI
- Programmation linéaire et non linéaire
- Optimisation combinatoire et globale
- Méthodes heuristiques et métaheuristiques
- Algorithmes multi-objectifs

PO sous Contraintes d'inégalité ou mixte

- **Méthodes de Pénalisation**
  - Les méthodes de pénalité classiques sont une famille d'algorithmes d'optimisation conçus pour résoudre des problèmes d'optimisation contrainte en les transformant en une séquence de problèmes d'optimisation sans contrainte.
  - L'idée fondamentale est d'ajouter un terme de "**pénalité**" à la fonction objective originale pour chaque contrainte violée.
  - L'ampleur de cette pénalité augmente à mesure que la violation de la contrainte devient plus importante.

PO sous Contraintes d’inégalité ou mixte

- **Méthodes de Pénalisation**

Soit le problème (P) $\begin{cases} \min\limits_{x\in\mathbb{R}^n} f(\mathrm{x}) \,, f \textit{ une fonction objective}. \\ \mathrm{g(x)} \quad 0 \\ \mathrm{h(x)} = 0 \end{cases}$

- où h : $\mathbb{R}^n \to \mathbb{R}^p$ est de classe **C¹**, la fonction de contrainte d'inégalité.
- g : $\mathbb{R}^n \to \mathbb{R}^m$, est de classe **C¹**, la fonction de contrainte d'égalité.
- x ∈ $\mathbb{R}^n$ est le vecteur des variables de décision.

On résout alors un **problème sans contraintes** (P ρ, μ) de la forme :

$$\min_{x\in\mathbb{R}^n} \varphi(x, \rho, \mu) \qquad \text{où} \;\; \varphi(x,\rho) = f(x) + \rho\, Pi(x) + \mu\, \mathrm{Pe(x)}$$

ρ, μ>0 paramètres de pénalisation pris assez grand et Pi et Pe les fonctions de pénalisation d’inégalité et égalité.

PO sous Contraintes d'inégalité ou mixte

- **Méthodes de Pénalisation**

  - Il existe principalement deux types de méthodes de pénalité classiques :
    - ✓ méthode des pénalités extérieures.
    - ✓ méthode des pénalités intérieurs.

PO sous Contraintes d'inégalité ou mixte

- **Méthodes des pénalités extérieures**
  - La méthode des pénalités extérieures (non strictes) est une technique d'optimisation sous contraintes qui transforme un problème avec contraintes en une suite de problèmes sans contraintes en ajoutant un **terme de pénalité** à la fonction objective.
  - Il s'agit d'autoriser les solutions non admissibles (qui ne respectent pas les contraintes), mais on les pénalise.
  - On commence à **l'extérieur** du domaine admissible, et la solution est poussée de manière itérative vers l'intérieur en **augmentant** la **pénalité**.

PO sous Contraintes d'inégalité ou mixte

- **Méthodes des pénalités extérieures**
  - La question qui se pose alors est : si résoudre le problème (P ρ, μ) permettra de résoudre le problème (P)?.
  - Ou autrement dit, si les ensembles de solutions des deux problèmes coïncident.
  - **Définition.** Une pénalisation est dite **exacte** si toute solution du problème (P) initial est solution du problème pénalisé (P ρ, μ), et **inexacte** dans le cas contraire.

PO sous Contraintes d'inégalité ou mixte

- **Méthodes des pénalités extérieures**
  - Les formes courantes de fonctions de pénalité**,** pour les contraintes d'**inégalité** Pi (**g_i(x)≤0**) :
    - ✓ Fonction de pénalité quadratique : PI_{i}(x)= $\boldsymbol{max\left(0, \mathrm{g}_{\mathrm{i}}(x)\right)^2}$
    - ✓ Fonction de pénalité de **Heaviside** (non différentiable) :

$$Pi_{\mathrm{j}}(\mathrm{x}) = \begin{cases} 0 \ si \ g_j(x) \leq 0 \\ 1 \ \ si \ g_j(x) > 0 \end{cases}$$

  - Les formes courantes de fonctions de pénalité, pour les contraintes d'**égalité** Pe(**h_j(x)=0**) :
    - ✓ Fonction de pénalité **quadratique : Pe_i(x)= h_i(x)^2**
    - ✓ Fonction de pénalité de **valeur absolue : Pe_i(x)=| h_i(x) |**
  - La fonction de pénalité totale est généralement la somme des pénalités pour chaque contrainte.

PO sous Contraintes d'inégalité ou mixte

▪ Méthodes des pénalités extérieures

- La fonction de pénalité totale est la somme des pénalités pour chaque contrainte.
- Par exemple pour un problème mixte le problème P peut se formuler comme suit :

$$P\rho,\mu \left\{ \min_{x \in \mathbb{R}^n} f(x) + \rho \sum_{j=1}^{j=m} \mathbf{P}\boldsymbol{i}_j(x) + \mu \sum_{i=1}^{i=p} \mathbf{P}\boldsymbol{e}_i(x) \right.$$

PO sous Contraintes d'inégalité ou mixte

- **Méthodes des pénalités extérieures**

Algorithme 11. Pénalisation classique (externe)

1-Initialisation **:** $\boldsymbol{x_0}, \boldsymbol{\varepsilon}, \mathbf{k = 0}, \mathbf{max\_iter}, \boldsymbol{f}, \boldsymbol{Pi}, \boldsymbol{Pe}: \mathbb{R}^{\boldsymbol{n}} \to \mathbb{R},$

$\mathbf{Pi}, \mathbf{Pe}$ **fonction de pénalité d'inégalité et égalité, paramètre de pénalité ρ_0, μ_0**

2-Répéter :

a)Résoudre le problème sans contrainte :

$$\min_{\mathbf{x}} \left[\mathrm{f(x)} + \rho_{\mathrm{k}} \sum_{j=1}^{j=m} \mathbf{P}\boldsymbol{i}_j\,(\mathrm{x}) + \mu_{\mathrm{k}} \sum_{i=1}^{i=p} \mathbf{P}\boldsymbol{e}_i\,(\mathrm{x}) \right]$$

b) Mise à jour : x_{k+1} est la solution approchée de a)

c) Augmenter **la pénalité :** $\rho_{k+1}=\beta\,\rho_k$, $\mu_{k+1}=\beta\,\mu_k$ où $\beta>1$.

d) k++

3-Jusqu'à $\| f(x_{k+1}) - f(x_k) \| < \varepsilon$ *ou max_iter atteint.*

4-Retourner x_k

PO sous Contraintes d'inégalité ou mixte

- **Méthodes des pénalités extérieures**

Exemple. 1 $$\begin{cases} \min_{x\in\mathbb{R}^n} f(\mathrm{x,y}) = \mathrm{x}^2 + (\mathrm{y}-2)^2 \,, f\ une\ fonction\ objective. \\ \mathrm{h(x,y)} = \mathrm{x} - \mathrm{y} = 0 \\ \mathrm{g(x,y)} = x + y - 1 \leq 0\,. \end{cases}$$

Utiliser la pénalisation quadratique, pour la fonction pénalisée: $\boldsymbol{max\left(0, \mathrm{g_i}(x)\right)^2}$, $\boldsymbol{h_i(\mathrm{x})^2}$

```python
# Fonction de pénalisation
def fonction_penalisee(xk, rho, mu, contraintes_inegalite, contraintes_egalite):
    x, y = xk
    penalite_inegalite = sum(rho * max(0, g(x, y))**2 for g in contraintes_inegalite)
    penalite_egalite = sum(mu * h(x, y)**2 for h in contraintes_egalite)
    return f(x, y) + penalite_inegalite + penalite_egalite
```

```python
# Méthode de pénalisation externe classique
def penalisation_externe_classique(fonction_objective, contraintes_inegalite, contraintes_egalite, x0, y0, rho0, mu0, beta, tolerance, options=None):
    xk = np.array([x0, y0])
    rho = rho0
    mu = mu0
    fk_prev = np.inf
    trajectory = [np.array([x0, y0])]  # Initialiser la trajectoire avec le point de départ
    total_iterations = 0  # Compteur total d'itérations

    while True:
        #résoudre le sous problème de minimisation
        res = minimize(
            fonction_penalisee,
            xk,
            args=(rho, mu, contraintes_inegalite, contraintes_egalite),
            #utiliser la méthode quasi-newton BFGS
            method='BFGS',
            options=options
        )
        xk_new = res.x
        fk_new = fonction_objective(*xk_new)
        total_iterations += res.nit  # Ajout des itérations de cette étape
        trajectory.append(xk_new)  # Ajouter le nouveau point à la trajectoire
        if np.abs(fk_new - fk_prev) < tolerance:
            break
        xk = xk_new
        rho *= beta
        mu *= beta
        fk_prev = fk_new
    return xk_new, total_iterations, np.array(trajectory)

# Point initial
x0, y0 = 0.0, 0.0
rho_initial = 1.0
mu_initial = 10.0
beta = 5.0  # Facteur d'augmentation unique pour rho et mu
tolerance_convergence = 1e-6
```

PO sous Contraintes d’inégalité ou mixte

- ## Méthodes des pénalités extérieures

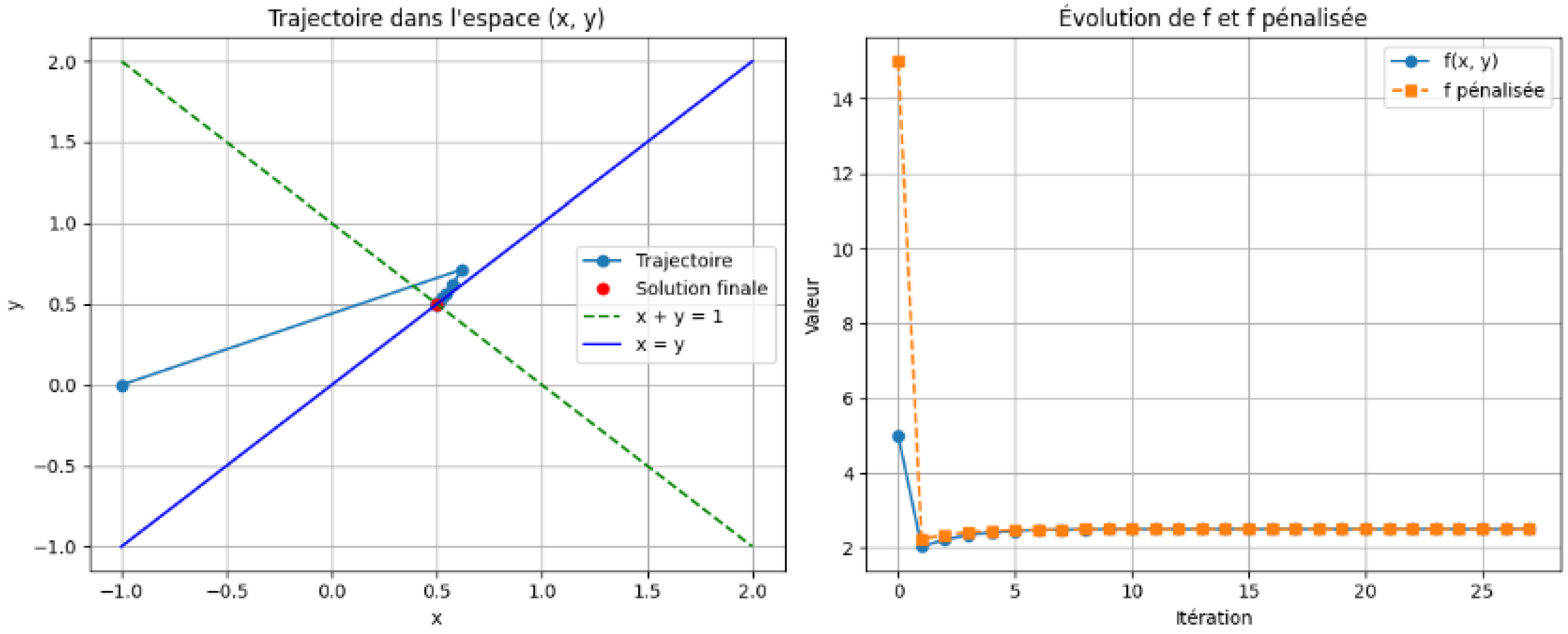
Exemple. 1 $\begin{cases} \min\limits_{x \in \mathbb{R}^n} f(x,y) = x^2 + (y-2)^2, f \text{ une fonction objective.} \\ h(x,y) = x - y = 0 \\ g(x,y) = x + y - 1 \leq 0. \end{cases}$

```python
# Point initial
x0, y0 = 0.0, 0.0
rho_initial = 1.0
mu_initial = 10.0
beta = 5.0  # Facteur d'augmentation unique pour rho et mu
tolerance_convergence = 1e-6

# Résolution
solution, iterations, trajectory = penalisation_externe_classique(
    fonction_objective=f,
    contraintes_inegalite=[g1],
    contraintes_egalite=[h1],
    x0=x0,
    y0=y0,
    rho0=rho_initial,
    mu0=mu_initial,
    beta=beta,
    tolerance=tolerance_convergence,
    options={'disp': False, 'maxiter': 100}
)
```

PO sous Contraintes d'inégalité ou mixte

- ## Méthodes des pénalités extérieures

```
Solution trouvée par la méthode de pénalisation externe classique:
x = 0.5000, y = 0.5000
f(x, y) = 2.5000
g1(x, y) = 0.0000
h1(x, y) = -0.0000
Nombre total d'itérations : 84
```



PO sous Contraintes d'iné

- **Méthodes des pénalités exté**

Exemple 2.
$$\begin{cases} \min\limits_{x\in\mathbb{R}^n} f(x,y) = (x^2 + y - 11)^2 + (x + y^2 - 7) \\ h(x,y) = y - x - 1 = 0 \\ g(x,y) = (x^2 + y^2 - 20) \leq 0\,. \end{cases}$$

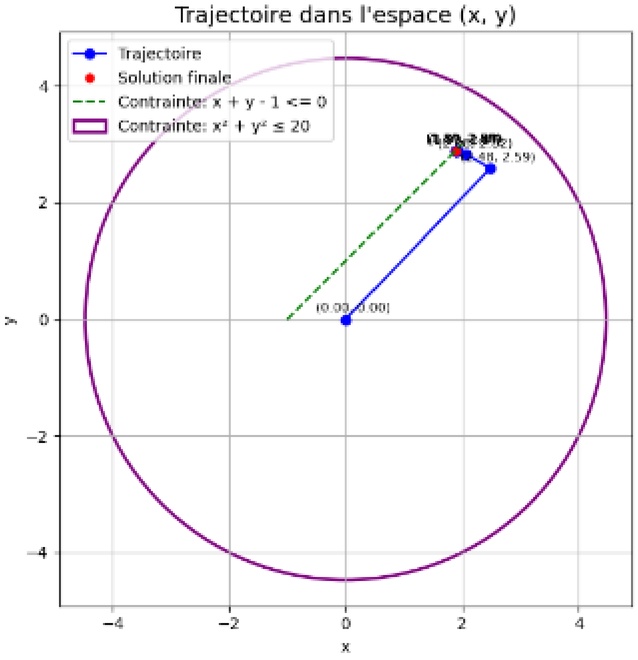
```
Solution trouvée par la méthode de pénalisation externe classique:
x = 1.8856, y = 2.8856
f(x, y) = 31.1025
g1(x, y) = -8.1178
h1(x, y) = 0.0000
Nombre total d'itérations : 39
```

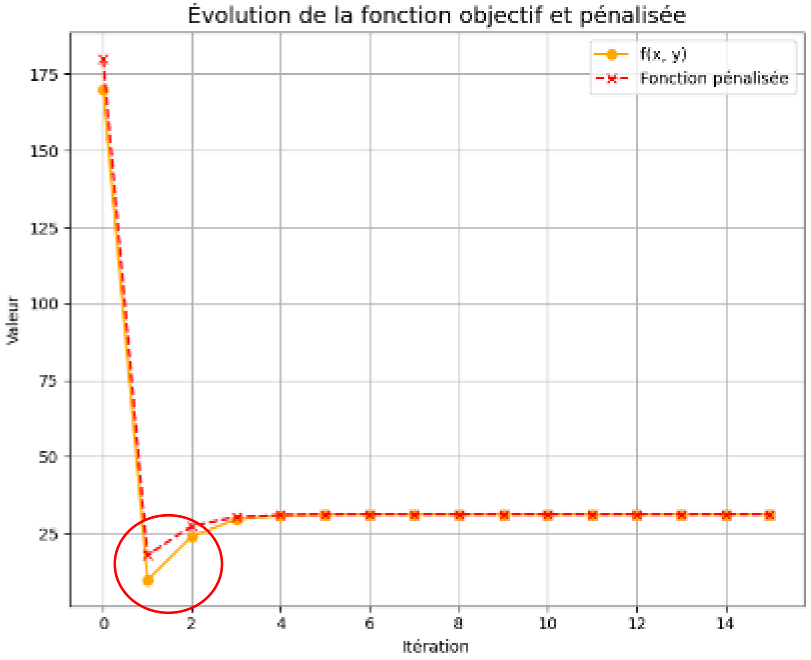
```
Solution à iteration  : 10
x = 2.4839, y = 2.5858
f(x, y) = 9.7490
Solution à iteration  : 16
x = 2.0773, y = 2.8160
f(x, y) = 24.0100
Solution à iteration  : 21
x = 1.9275, y = 2.8721
f(x, y) = 29.5631
Solution à iteration  : 25
x = 1.8941, y = 2.8829
f(x, y) = 30.7907
Solution à iteration  : 28
x = 1.8873, y = 2.8851
f(x, y) = 31.0400
Solution à iteration  : 30
x = 1.8859, y = 2.8855
f(x, y) = 31.0900
Solution à iteration  : 32
x = 1.8856, y = 2.8856
f(x, y) = 31.1000
Solution à iteration  : 33
x = 1.8856, y = 2.8856
f(x, y) = 31.1020
Solution à iteration  : 34
x = 1.8856, y = 2.8856
f(x, y) = 31.1024
Solution à iteration  : 35
x = 1.8856, y = 2.8856
f(x, y) = 31.1025
Solution à iteration  : 36
x = 1.8856, y = 2.8856
f(x, y) = 31.1025
Solution à iteration  : 37
x = 1.8856, y = 2.8856
f(x, y) = 31.1025
Solution à iteration  : 38
x = 1.8856, y = 2.8856
f(x, y) = 31.1025
Solution à iteration  : 39
x = 1.8856, y = 2.8856
f(x, y) = 31.1025
Solution à iteration  : 39
x = 1.8856, y = 2.8856
f(x, y) = 31.1025

Solution trouvée par la méthode de pénalisation externe classique:
x = 1.8856, y = 2.8856
```

PO sous Contraintes d'inégalité ou mixte

- **Méthodes des pénalités extérieures**

```
Solution trouvée par la méthode de pénalisation externe classique:
x = 1.8856, y = 2.8856
f(x, y) = 31.1025
g1(x, y) = -8.1178
h1(x, y) = 0.0000
Nombre total d'itérations : 39
```





Si les contraintes sont violées, les pénalités associées augmentent, ce qui peut entraîner une augmentation temporaire de la fonction pénalisée

PO sous Contraintes d'inégalité ou mixte

- **Méthodes des pénalités extérieures**
  - **Avantages :**
    - ✓ La méthode transforme un problème contraint en une suite de problèmes **non contraints**, permettant d'utiliser des **algorithmes classiques** d'optimisation (comme le gradient ou quasi-Newton).
    - ✓ Peut être appliquée à une grande variété de problèmes avec contraintes d'égalité et d'inégalité
  - **Inconvénients :**
    - ✓ Pour respecter exactement les contraintes, il faut que les paramètres de pénalisation (ρ_k ou μ_k) deviennent très grands, ce qui peut rendre la **fonction objective mal conditionnée, c**ela peut **ralentir ou bloquer la convergence**.
    - ✓ La **solution du problème pénalisé** ne satisfait pas toujours exactement les contraintes du problème original — la **réalisabilité** n'est atteinte qu'à la **limite**.
    - ✓ Le **choix des paramètres de pénalisation** influence fortement la performance et la précision,

PO sous Contraintes d’inégalité ou mixte

- **Méthodes des pénalités intérieures (barrière)**
  - les méthodes de pénalité intérieure classiques sont conçues spécifiquement pour traiter les contraintes **d'inégalité** de la forme $gj(x) \leq 0$.
  - Dans certains cas, il n’est pas envisageable que les itérés n’appartiennent pas au domaine *S* des contraintes, en particulier quand la fonction objective n’est pas définie en dehors de *S*.
  - Le principe fondamental des méthodes de pénalité intérieure est de créer une "barrière" qui empêche la solution de quitter l'intérieur de la région admissible définie par les contraintes **d'inégalité.**

PO sous Contraintes d'inégalité ou mixte

- **Méthodes des pénalités intérieures (barrière)**
  - **Le problème traité est sous forme :**

$$\begin{cases} \min_{x \in \mathbb{R}^n} f(x)\,, f \text{ une fonction objective.} \\ g(x) \quad 0\,. \end{cases}$$

g : $\mathbb{R}^n \rightarrow \mathbb{R}^m$, la fonction de contrainte d'égalité est différentiable.

Soit T={x| g_i(x)<0,∀i}, l'optimisation est effectuée uniquement dans cette région.

PO sous Contraintes d'inégalité ou mixte

- **Méthodes des pénalités intérieures (barrière)**
  - **Le problème devient après introduction d'une fonction barrière :**

    $(P\rho)\ \min_{x\in \mathbb{R}^n} f(x) + \rho B(x)$, ou ρ tend vers 0.
  - Les formes courantes de B qui permettent de respecter cette contrainte lorsque g(x)<0 sont :
    - ✓ **Fonction barrière inverse :** $B(x) = -\sum_{i=1}^{i=p} \frac{1}{g_i(x)}$
    - ✓ **Fonction barrière logarithmique :** $B(x) = -\sum_{i=1}^{i=p} \ln(-g_i(x))$, on appelle ca la Méthode de la barrière logarithmique

PO sous Contraintes d'inégalité ou mixte

- **Méthodes des pénalités intérieures (barrière)**

Algorithme 12. Pénalisation classique (interne)

1. Initialisation : $x_0, \varepsilon, \mathrm{k} = 0, \max\_\mathrm{iter}$,

f, Pi: $\mathbb{R}^n \to \mathbb{R}$, Bj fonction de $barrière$, paramètre de pénalité ρ_0

2. Répéter :

a)Résoudre le problème sans contrainte :

$$\min_{\mathbf{x}} \left[\mathrm{f(x)} + \rho_k \sum_{j=1}^{j=m} B_j\,(\mathrm{x}) \right]$$

b) Mise à jour: x_{k+1} est la solution approchée de a)

c) Diminuer la pénalité : $\rho_{k+1}=\beta\ \rho_{k}$, où $0< \beta <1$.

d)k++

3. Jusqu'à ‖ f(x_{k+1})- f(x_k)‖)< ε $\boldsymbol{ou\ max\_iter}$.

4. Retourner $\mathbf{x_k}$

PO sous Contraintes d'inégalité ou mixte

- **Méthodes des pénalités intérieures (barrière)**

Exemple 1.

$$\begin{cases} \min_{x\in\mathbb{R}^n} f(x) = x^2+1 \text{ , } f \textit{ une fonction objective.} \\ \\ g(x)=x-2 \leq 0 \text{ .} \end{cases}$$

On va transformer ce problème contraint en une séquence de problèmes non contraints en ajoutant un terme de barrière pour la violation de la contrainte

$B(x) = -\log(2 - x)$, on a une seule contrainte

Le problème non contraint pénalisé devient :

$$\min_{x\in\mathbb{R}^n} \varphi(x, \rho) \textit{ ou } \varphi(x, \rho) = f(x) + \rho B(x = f(x) - \rho \log(2 - x)$$

$$\varphi(x, \rho) = x^2+1 - \rho \log(2 - x)$$

PO sous Contraintes d'inégalité ou mixte

- **Méthodes des pénalités intérieures (barrière)**

Exemple 1.

$$\begin{cases}(\min\limits_{x\in\mathbb{R}^n} f(x) = x^2+1 \text{ , } f \text{ une fonction objective} \\ \qquad\qquad g(x) = x - 2 \ \leq \ 0)\end{cases}$$

```python
# Fonction barrière : f(x) - mu * log(2 - x)
def f_barriere(x, mu):
    if 2 - x <= 0:
        return np.inf  # Invalide hors domaine
    return f(x) - mu * np.log(2 - x)
```

PO sous Contraintes d'inégalité ou mixte

- **Méthodes des pénalités intérieures (barrière)**

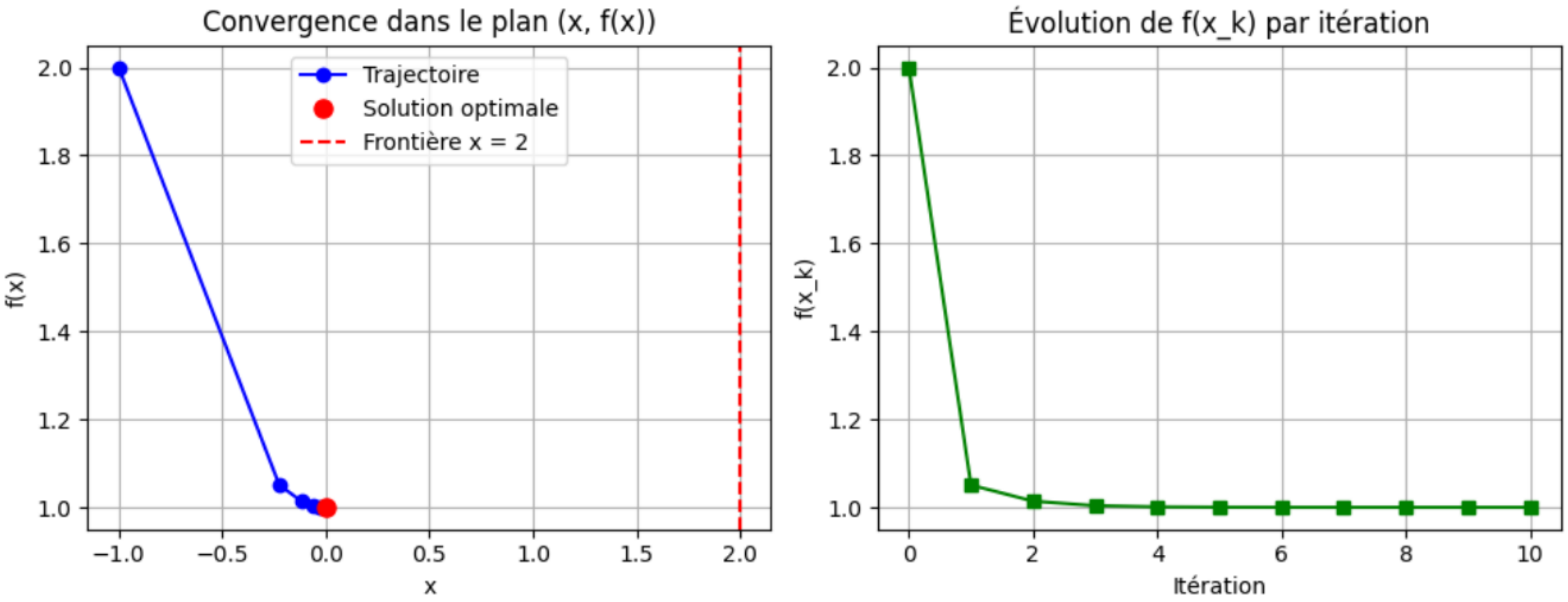
Exemple 1. $\begin{cases} (\min\limits_{x\in\mathbb{R}^n} f(x) = x^2+1 \text{ , } f \textit{ une fonction objective} \\ \qquad\qquad g(x) = x - 2 \le 0) \end{cases}$

```python
# Méthode de barrière avec critère d'arrêt basé sur la variation de f
def methode_penalisation_interne(x0, mu0=1.0, beta=0.5, epsilon=1e-10, max_iter=50):
    xk = x0
    mu = mu0
    trajectory = [xk]
    f_prec = f(xk)

    for k in range(max_iter):
        res = minimize(lambda x: f_penalisation(x, mu), [xk], method='BFGS')
        xk = res.x[0]
        trajectory.append(xk)
        f_courant = f(xk)
        critere = abs(f_courant - f_prec)
        if critere < epsilon:
            break

        f_prec = f_courant
        mu *= beta

    return xk, np.array(trajectory), k + 1  # +1 car k commence à 0
```

PO sous Contraintes d'inégalité ou mixte

- ## Méthodes des pénalités intérieures (barrière)

```
Solution optimale approchée : x = -0.000488
Valeur de f(x) = 1.000000
Nombre d'itérations : 10
```

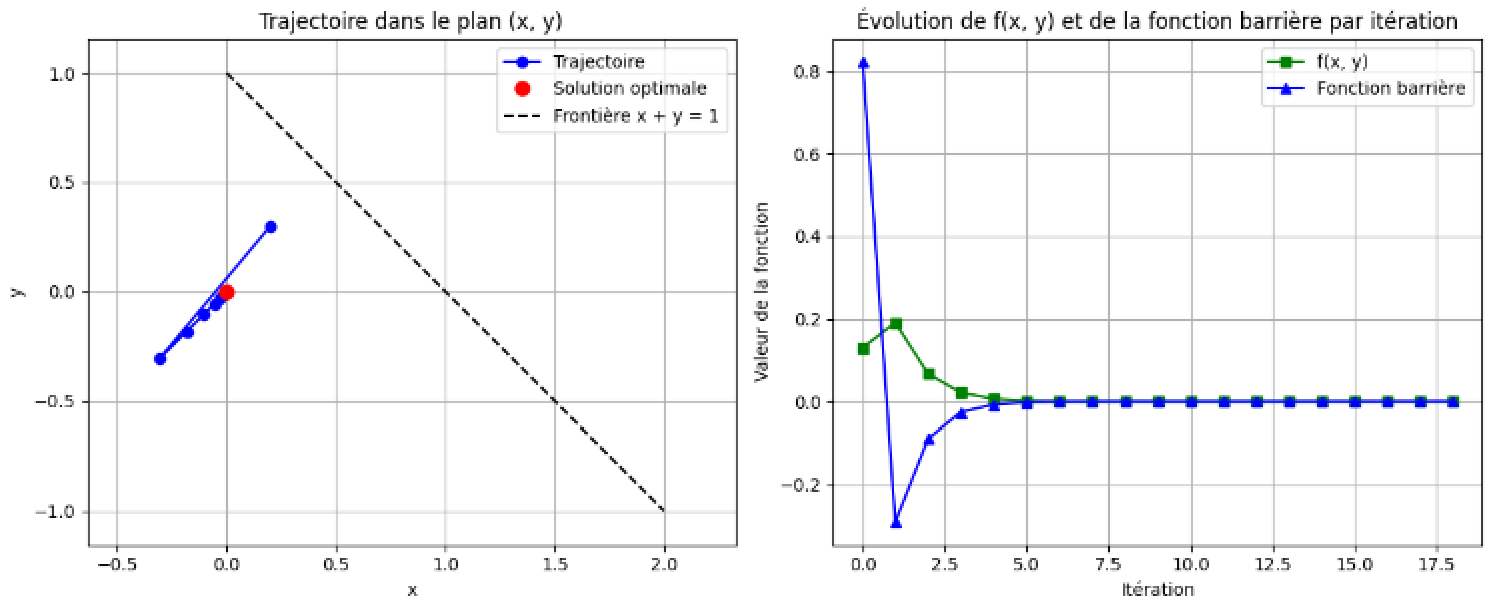


PO sous Contraintes d’inégalité ou mixte

- **Méthodes des pénalités intérieures (barrière)**

Exemple 2. $\begin{cases} \min\limits_{x \in \mathbb{R}^n} f(\mathrm{x},\mathrm{y}) = x^2 + y^2, f \; la \; fonction \; objective. \\ g(x) = x + y - 1 \leq 0 \end{cases}$

```
Solution optimale approchée : x = (-0.000008, -0.000008)
Valeur de f(x, y) = 0.000000
Nombre d'itérations : 18
```



PO sous Contraintes d’inégalité ou mixte

- **Méthodes des pénalités intérieures (barrière)**
  - Avantages
    - ✓ Contrairement aux méthodes de pénalisation externe, les itérés **restent toujours à l'intérieur du domaine admissible** (satisfont gi(x)<0).
    - ✓ **Utile pour les problèmes**, où les contraintes ne doivent jamais être violées.
    - ✓ Contrairement aux pénalisations externes (qui deviennent mal conditionnées pour des **pénalités élevées**), la barrière logarithmique reste mieux conditionnée.
  - Limites
    - ✓ Nécessite un point initial strictement **admissible**, Trouver un tel point peut être difficile, voire impossible pour certains problèmes complexes.
    - ✓ Inadaptée aux contraintes **d'égalité**, il faut utiliser une solutions hybrides (ex : Lagrangien augmenté + barrière).
    - ✓ **Sensibilité au choix de ρ**, si ρ décroît trop vite on a un risque de divergence, sinon si ρ décroît trop lentement → Convergence lente.