

# Méthode de descente

---

## Idée

1. Trouver une direction de descente  $d_k$ , c'est-à-dire telle que  $\nabla f(x_k)^T d_k < 0$ .
2. Trouver un pas  $\alpha_k$  tel que  $f(x_k + \alpha_k d_k) < f(x_k)$ .
3. Calculer  $x_{k+1} = x_k + \alpha_k d_k$ .

# Plus forte pente

---

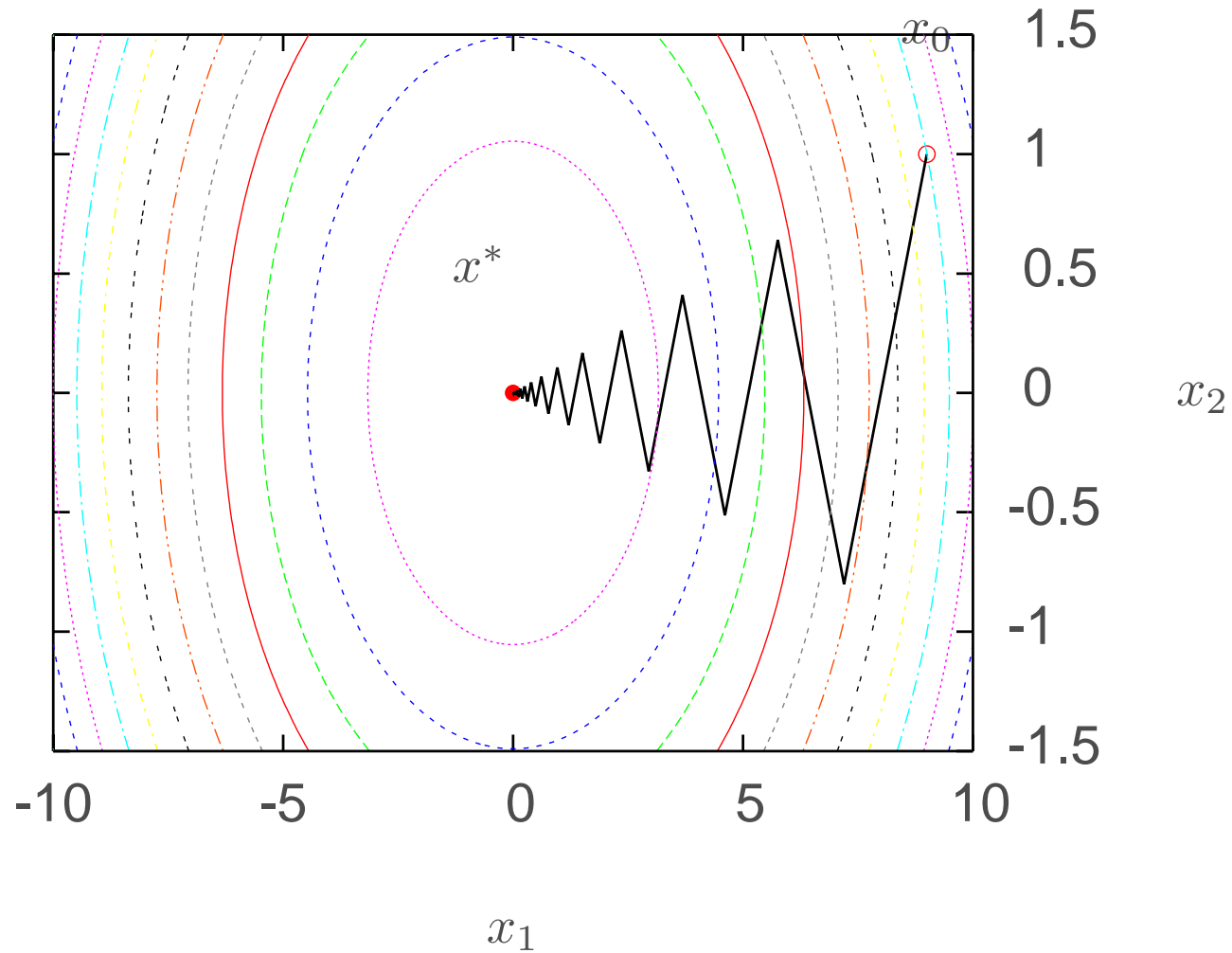
- Choix intuitif de la direction :  $d_k = -\nabla f(x_k)$
- Choix du pas

$$\alpha_k = \operatorname{argmin}_{\alpha \in \mathbb{R}_0^+} f(x_k + \alpha d_k).$$

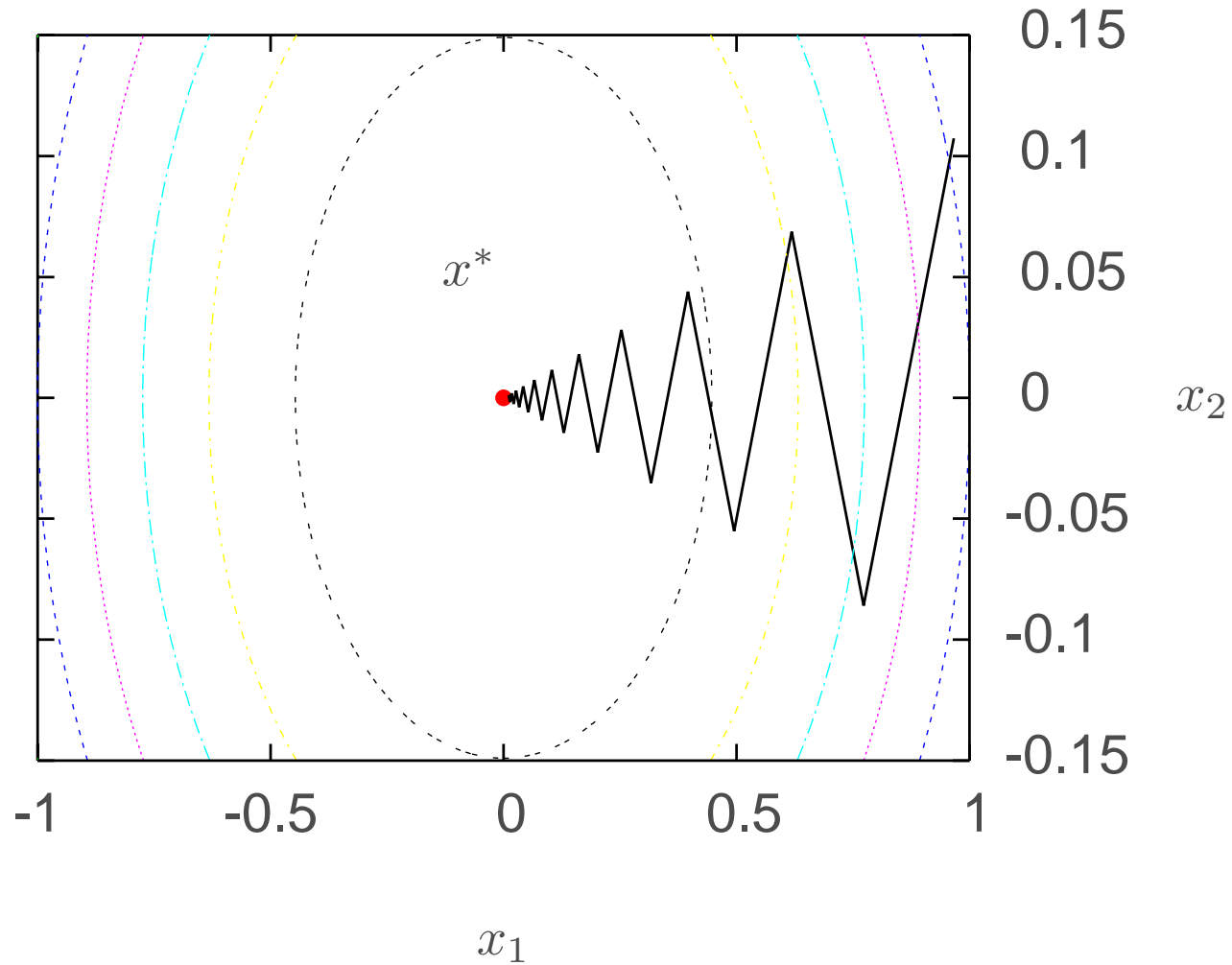
Exemple :

$$f(x) = \frac{1}{2}x_1^2 + \frac{9}{2}x_2^2$$

# Plus forte pente



# Plus forte pente



$(x_k)_1$	$(x_k)_2$	$\nabla f(x_k)_1$	$\nabla f(x_k)_2$	$\alpha_k$	$f(x_k)$
+9.000000E+00	+1.000000E+00	+9.000000E+00	+9.000000E+00	0.2	+4.500000E+01
+7.200000E+00	-8.000000E-01	+7.200000E+00	-7.200000E+00	0.2	+2.880000E+01
+5.760000E+00	+6.400000E-01	+5.760000E+00	+5.760000E+00	0.2	+1.843200E+01
+4.608000E+00	-5.120000E-01	+4.608000E+00	-4.608000E+00	0.2	+1.179648E+01
+3.686400E+00	+4.096000E-01	+3.686400E+00	+3.686400E+00	0.2	+7.549747E+00
+2.949120E+00	-3.276800E-01	+2.949120E+00	-2.949120E+00	0.2	+4.831838E+00
+1.037629E-01	+1.152922E-02	+1.037629E-01	+1.037629E-01	0.2	+5.981526E-03
+8.301035E-02	-9.223372E-03	+8.301035E-02	-8.301035E-02	0.2	+3.828177E-03
+6.640828E-02	+7.378698E-03	+6.640828E-02	+6.640828E-02	0.2	+2.450033E-03
+5.312662E-02	-5.902958E-03	+5.312662E-02	-5.312662E-02	0.2	+1.568021E-03
+4.250130E-02	+4.722366E-03	+4.250130E-02	+4.250130E-02	0.2	+1.003534E-03
+3.400104E-02	-3.777893E-03	+3.400104E-02	-3.400104E-02	0.2	+6.422615E-04
+1.284523E-04	+1.427248E-05	+1.284523E-04	+1.284523E-04	0.2	+9.166662E-09
+1.027618E-04	-1.141798E-05	+1.027618E-04	-1.027618E-04	0.2	+5.866664E-09
+8.220947E-05	+9.134385E-06	+8.220947E-05	+8.220947E-05	0.2	+3.754665E-09
+6.576757E-05	-7.307508E-06	+6.576757E-05	-6.576757E-05	0.2	+2.402985E-09
+5.261406E-05	+5.846007E-06	+5.261406E-05	+5.261406E-05	0.2	+1.537911E-09
+4.209125E-05	-4.676805E-06	+4.209125E-05	-4.209125E-05	0.2	+9.842628E-10

# Plus forte pente préconditionnée

---

$$f(x) = \frac{1}{2}x_1^2 + \frac{9}{2}x_2^2$$

Changement de variable :

$$\begin{aligned}x'_1 &= x_1 \\x'_2 &= 3x_2\end{aligned}$$

et

$$\tilde{f}(x') = \frac{1}{2}x_1'^2 + \frac{9}{2}\left(\frac{1}{3}x_2'\right)^2 = \frac{1}{2}x_1'^2 + \frac{1}{2}x_2'^2.$$

# Plus forte pente préconditionnée

---

$$\tilde{f}(x') = \frac{1}{2}x_1'^2 + \frac{1}{2}x_2'^2.$$

Direction :

$$d = -\nabla \tilde{f}(x') = \begin{pmatrix} -x_1' \\ -x_2' \end{pmatrix}.$$

Pas :

$$\begin{aligned} & \operatorname{argmin}_{\alpha} f(x' - \alpha \nabla f(x')) = \\ & \min_{\alpha} \frac{1}{2}(x_1' - \alpha x_1')^2 + \frac{1}{2}(x_2' - \alpha x_2')^2, \end{aligned}$$

# Plus forte pente préconditionnée

---

Solution :  $\alpha = 1$

$$\begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix} + \begin{pmatrix} -x'_1 \\ -x'_2 \end{pmatrix} = 0,$$



# Plus forte pente préconditionnée

---

- Après conditionnement, la méthode de la plus forte pente converge en une seule itération sur cet exemple
- D'une manière générale, un pré-conditionnement peut significativement accélérer la méthode
- Analysons l'impact d'un changement de variables sur la méthode.

(p. 261)

# Algorithme : Plus forte pente préconditionnée

---

## Objectif

Trouver une approximation de la solution du problème

$$\min_{x \in \mathbb{R}^n} f(x).$$

## Input

- La fonction  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  différentiable;
- Le gradient de la fonction  $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ ;
- Une famille de préconditionneurs  $(D_k)_k$  telle que  $D_k$  est définie positive pour tout  $k$ ;
- $x_0 \in \mathbb{R}^n$ ;
- La précision demandée  $\varepsilon \in \mathbb{R}, \varepsilon > 0$ .

# Algorithme : Plus forte pente préconditionnée

---

## Output

Une approximation de la solution  $x^* \in \mathbb{R}$

## Initialisation

$$k = 0$$

## Itérations

1.  $d_k = -D_k \nabla f(x_k)$ ,
2. Déterminer  $\alpha_k$ , par exemple  $\alpha_k = \operatorname{argmin}_{\alpha \geq 0} f(x_k + \alpha d_k)$ ,
3.  $x_{k+1} = x_k + \alpha_k d_k$ ,
4.  $k = k + 1$ .

**Critère d'arrêt** Si  $\|\nabla f(x_k)\| \leq \varepsilon$ , alors  $x^* = x_k$ .

# Plus forte pente préconditionnée

---

Il reste à préciser

- comment choisir  $D_k$
- comment choisir  $\alpha_k$

et il reste à s'assurer que cela fonctionne...

# Choix du pas

---

- Résolution de

$$\alpha_k = \operatorname{argmin}_{\alpha \in \mathbb{R}_0^+} f(x_k + \alpha d_k).$$

trop coûteuse

- Travail inutile si la direction n'est pas bonne
- Prenons n'importe quel  $\alpha$  tel que

$$f(x_k + \alpha d_k) < f(x_k)$$

- Malheureusement, cela ne suffit pas...

# Choix du pas

---

- Exemple :  $f(x) = x^2$
- Appliquons l'algorithme avec  $x_0 = 2$ , et

$$\begin{aligned}D_k &= 1/2|x_k| = \text{sgn}(x_k)/2x_k \\ \alpha_k &= 2 + 3(2^{-k-1}).\end{aligned}$$

- $D_k$  est bien (défini) positif pour tout  $k$ .
- $\nabla f(x_k) = 2x_k \Rightarrow d_k = -D_k \nabla f(x_k) = -\text{sgn}(x_k)$
- La méthode s'écrit

$$x_{k+1} = \begin{cases} x_k - 2 - 3(2^{-k-1}) & \text{si } x_k \geq 0, \\ x_k + 2 + 3(2^{-k-1}) & \text{si } x_k < 0, \end{cases}$$

# Choix du pas

---

Nous avons que

$$x_k = (-1)^k (1 + 2^{-k})$$

et

$$|x_{k+1}| < |x_k|.$$

(p. 264)

Dès lors

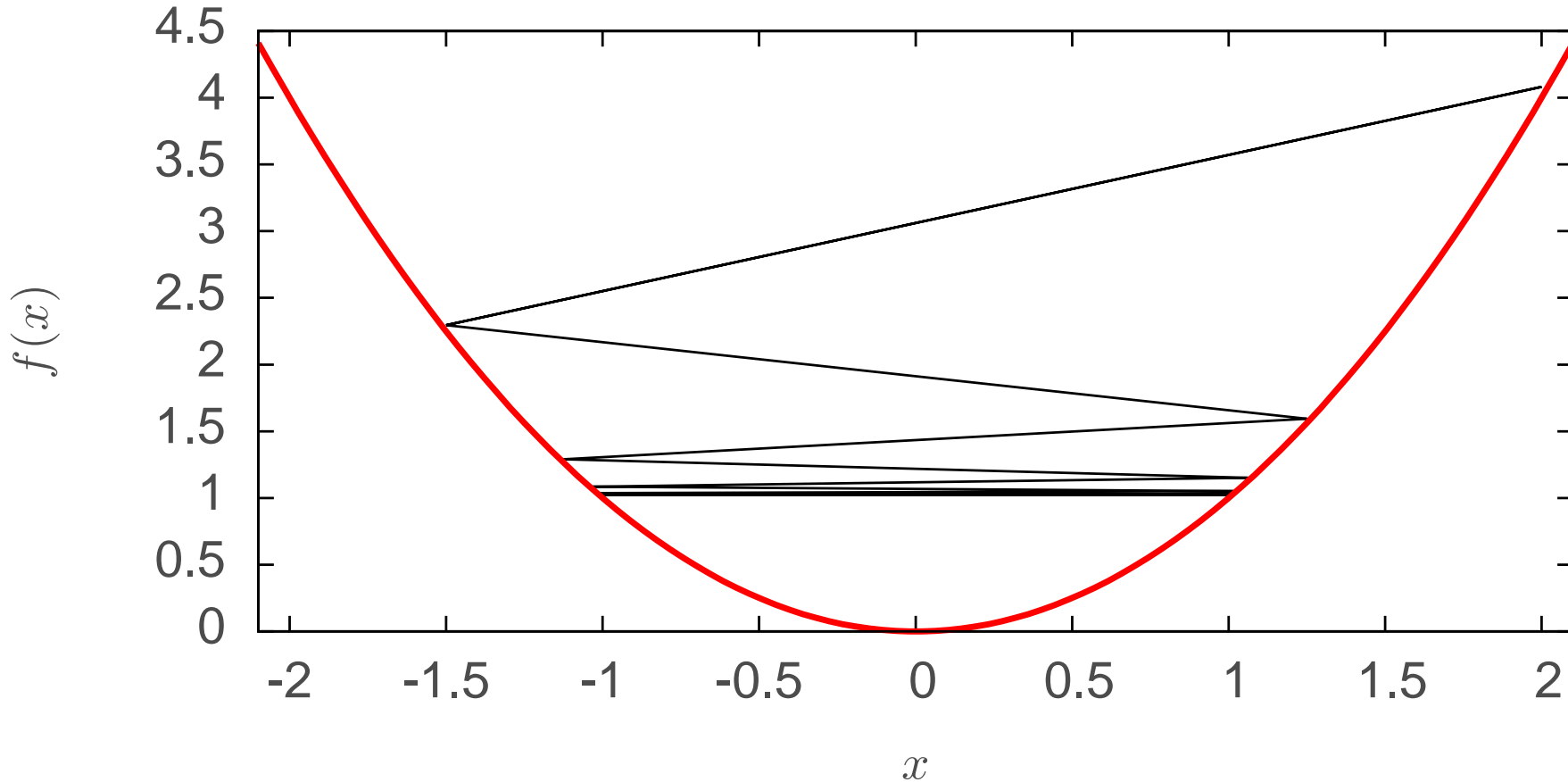
$$f(x_{k+1}) < f(x_k)$$

Cependant, la suite  $x_k$  a deux points d'accumulation: -1 et 1

$k$	$x_k$	$d_k$	$\alpha_k$
0	+2.000000e+00	-1	+3.500000e+00
1	-1.500000e+00	1	+2.750000e+00
2	+1.250000e+00	-1	+2.375000e+00
3	-1.125000e+00	1	+2.187500e+00
4	+1.062500e+00	-1	+2.093750e+00
5	-1.031250e+00	1	+2.046875e+00
:			
46	+1.000000e+00	-1	+2.000000e+00
47	-1.000000e+00	1	+2.000000e+00
48	+1.000000e+00	-1	+2.000000e+00
49	-1.000000e+00	1	+2.000000e+00
50	+1.000000e+00	-1	+2.000000e+00



# Choix du pas



# Choix du pas

---

Pourquoi cela ne fonctionne pas ?

- Origine théorique : théorème de Taylor
- Théorie locale
- Ici, pas trop longs
- Le fait que  $f(x_{k+1}) < f(x_k)$  est du à la chance et non au fait que  $d^T \nabla f(x_k) < 0$
- Les pas sont trop longs par rapport au bénéfice obtenu

Notion de diminution suffisante

# Diminution suffisante

---

Soit  $\gamma > 0$ . On veut

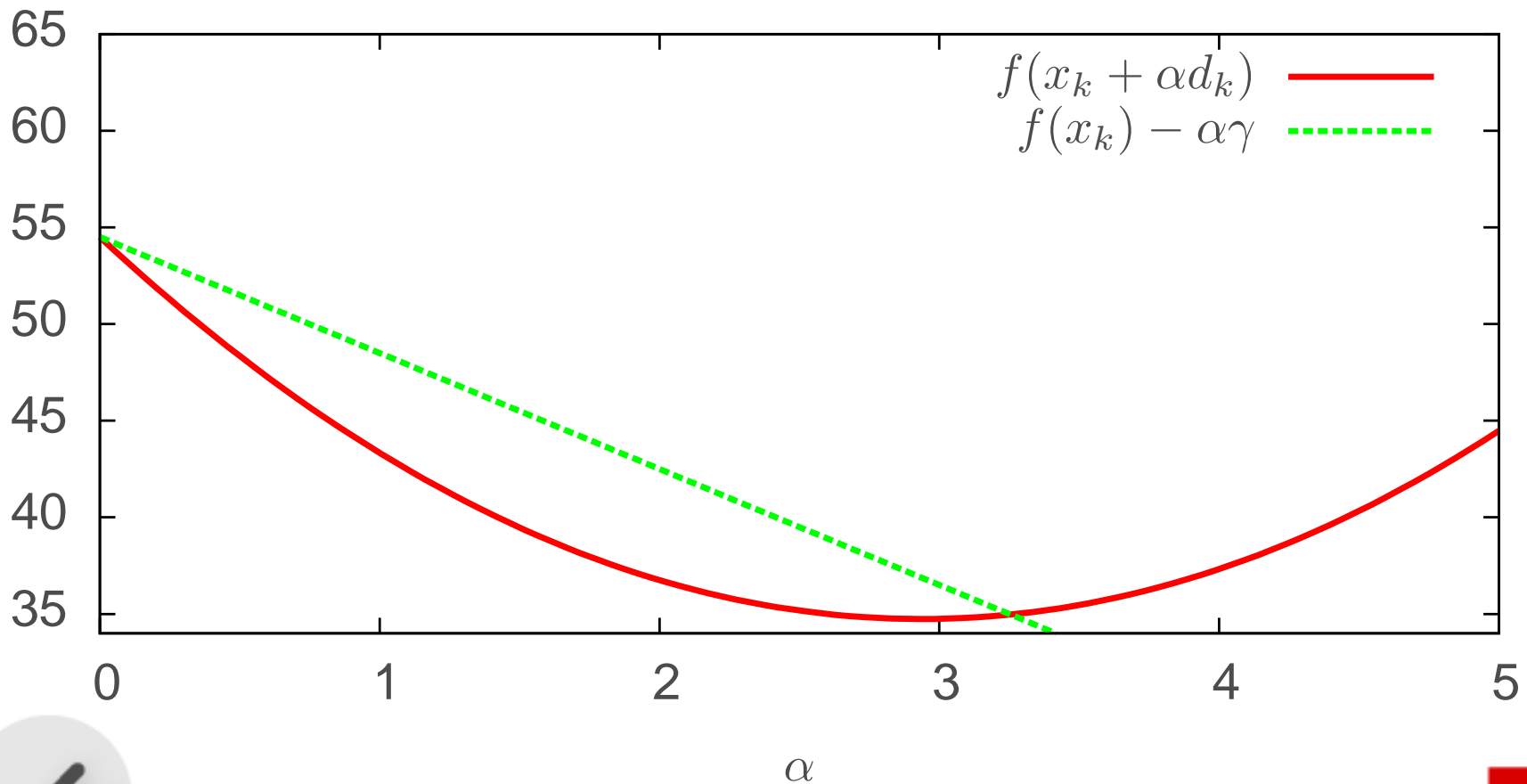
$$f(x_k) - f(x_k + \alpha_k d_k) \geq \alpha_k \gamma,$$

ou encore

$$f(x_k + \alpha_k d_k) \leq f(x_k) - \alpha_k \gamma.$$

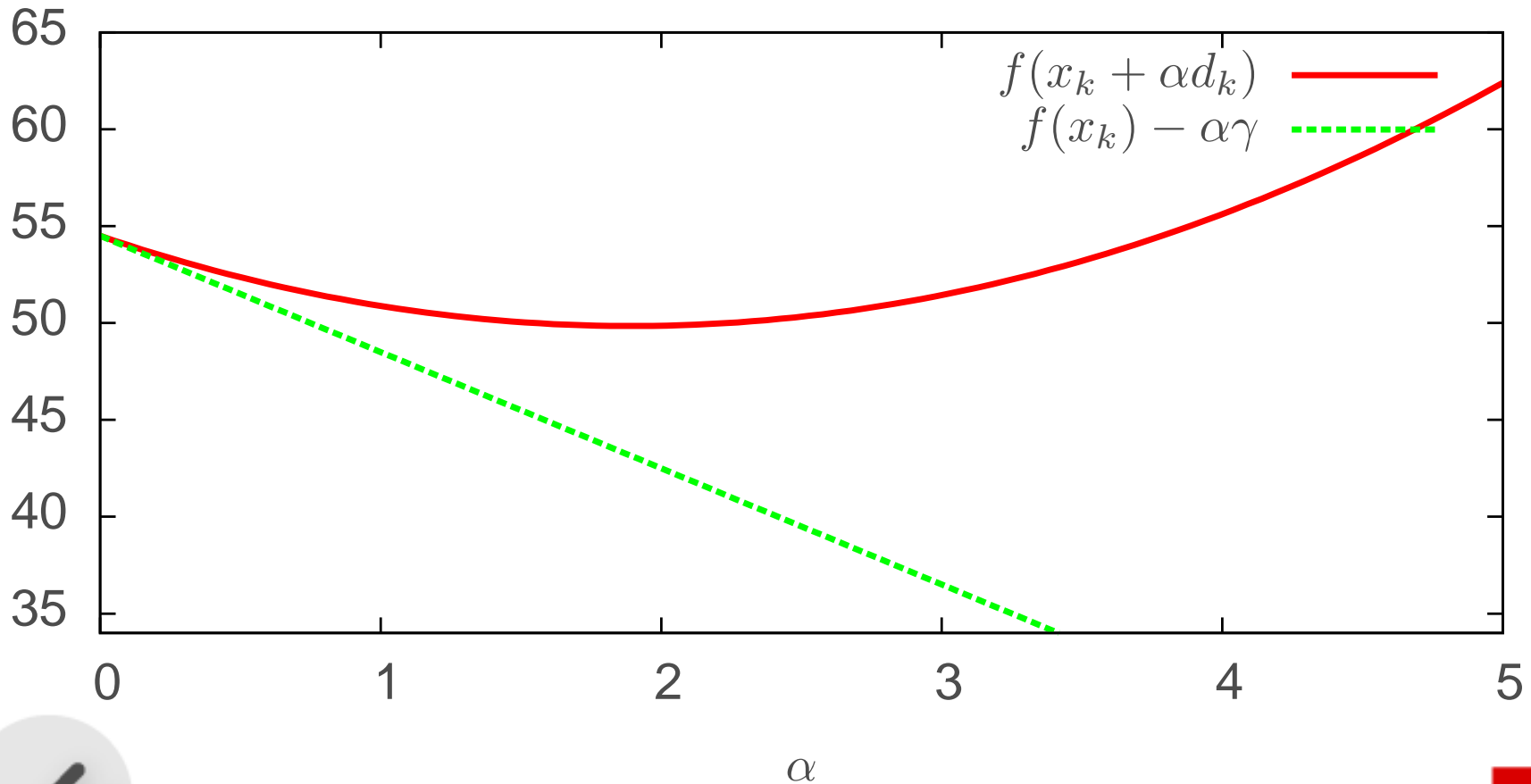
# Diminution suffisante

Exemple :  $f(x) = \frac{1}{2}x_1^2 + \frac{9}{2}x_2^2$ ,  $x_0 = \begin{pmatrix} 10 \\ 1 \end{pmatrix}$ ,  $d = \begin{pmatrix} \frac{-10}{\sqrt{181}} \\ \frac{-9}{\sqrt{181}} \end{pmatrix}$ ,  $\gamma = 6$ .



# Diminution suffisante

Exemple :  $f(x) = \frac{1}{2}x_1^2 + \frac{9}{2}x_2^2$ ,  $x_0 = \begin{pmatrix} 10 \\ 1 \end{pmatrix}$ ,  $d = \begin{pmatrix} \frac{-2}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} \end{pmatrix}$ ,  $\gamma = 6$ .



# Diminution suffisante

---

- $\gamma$  ne peut pas être constant
- Il doit dépendre de la direction
- Utilisons la théorie

# Rappel

---

**Direction de descente** Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction différentiable.

Soient  $x \in \mathbb{R}^n$  tel que  $\nabla f(x) \neq 0$  et  $d \in \mathbb{R}^n$ .

Si  $d$  est une direction de descente, alors il existe  $\eta > 0$  tel que

$$f(x + \alpha d) < f(x) \quad \forall 0 < \alpha \leq \eta.$$

De plus, pour tout  $\beta < 1$ , il existe  $\hat{\eta} > 0$  tel que

$$f(x + \alpha d) < f(x) + \alpha \beta \nabla f(x)^T d,$$

pour tout  $0 < \alpha \leq \hat{\eta}$ .

(voir début du cours et p. 36)

# Diminution suffisante

---

Choisissons

$$\gamma = -\beta \nabla f(x_k)^T d_k$$

avec  $0 < \beta < 1$ .



# Diminution suffisante

## Diminution suffisante : première condition de Wolfe

Soient  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction différentiable, un point  $x_k \in \mathbb{R}^n$ , une direction (de descente)  $d_k \in \mathbb{R}^n$  telle que  $\nabla f(x_k)^T d_k < 0$  et un pas  $\alpha_k \in \mathbb{R}$ ,  $\alpha_k > 0$ .

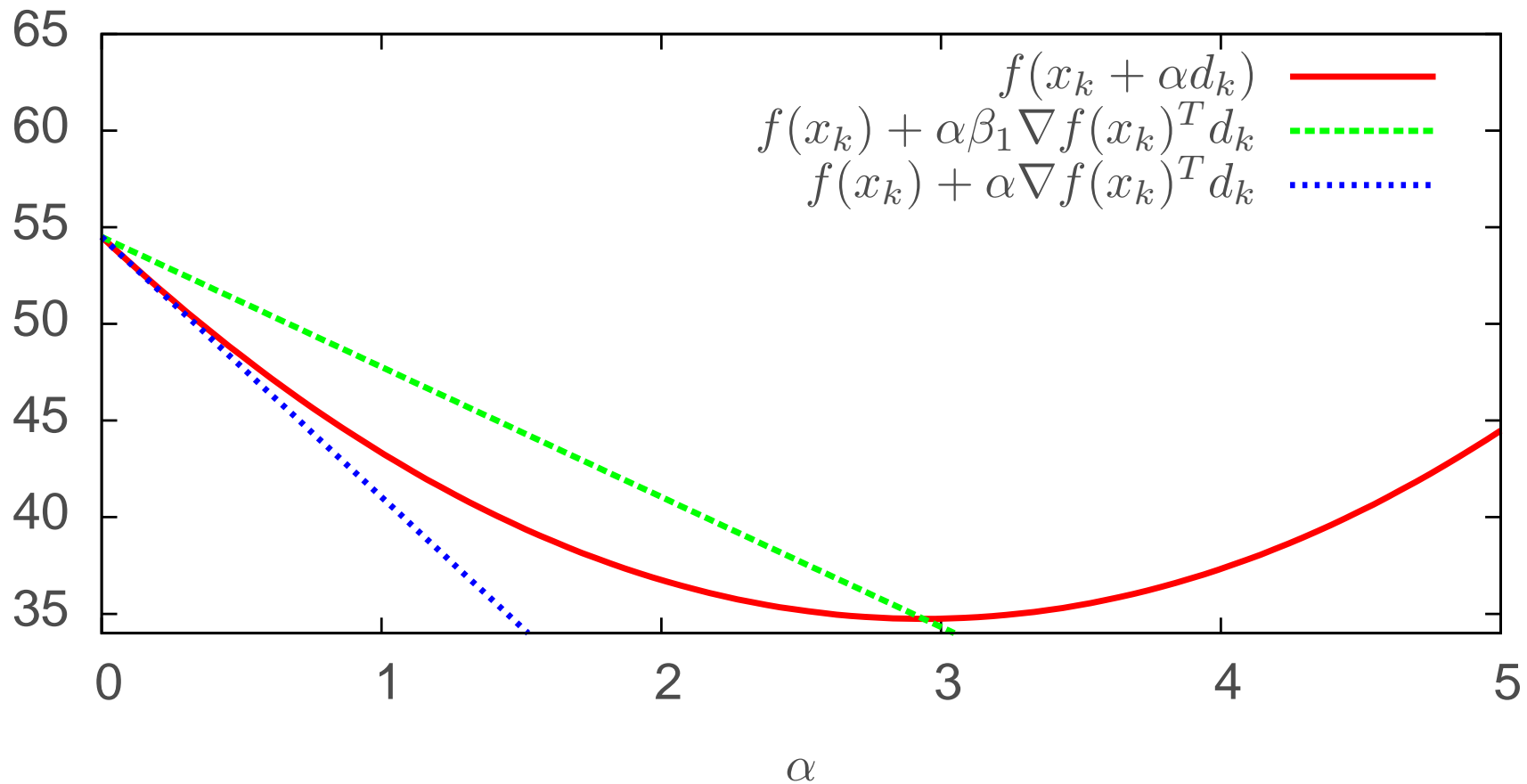
On dira que la fonction  $f$  diminue suffisamment en  $x_k + \alpha_k d_k$  par rapport à  $x_k$  si

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \alpha_k \beta_1 \nabla f(x_k)^T d_k,$$

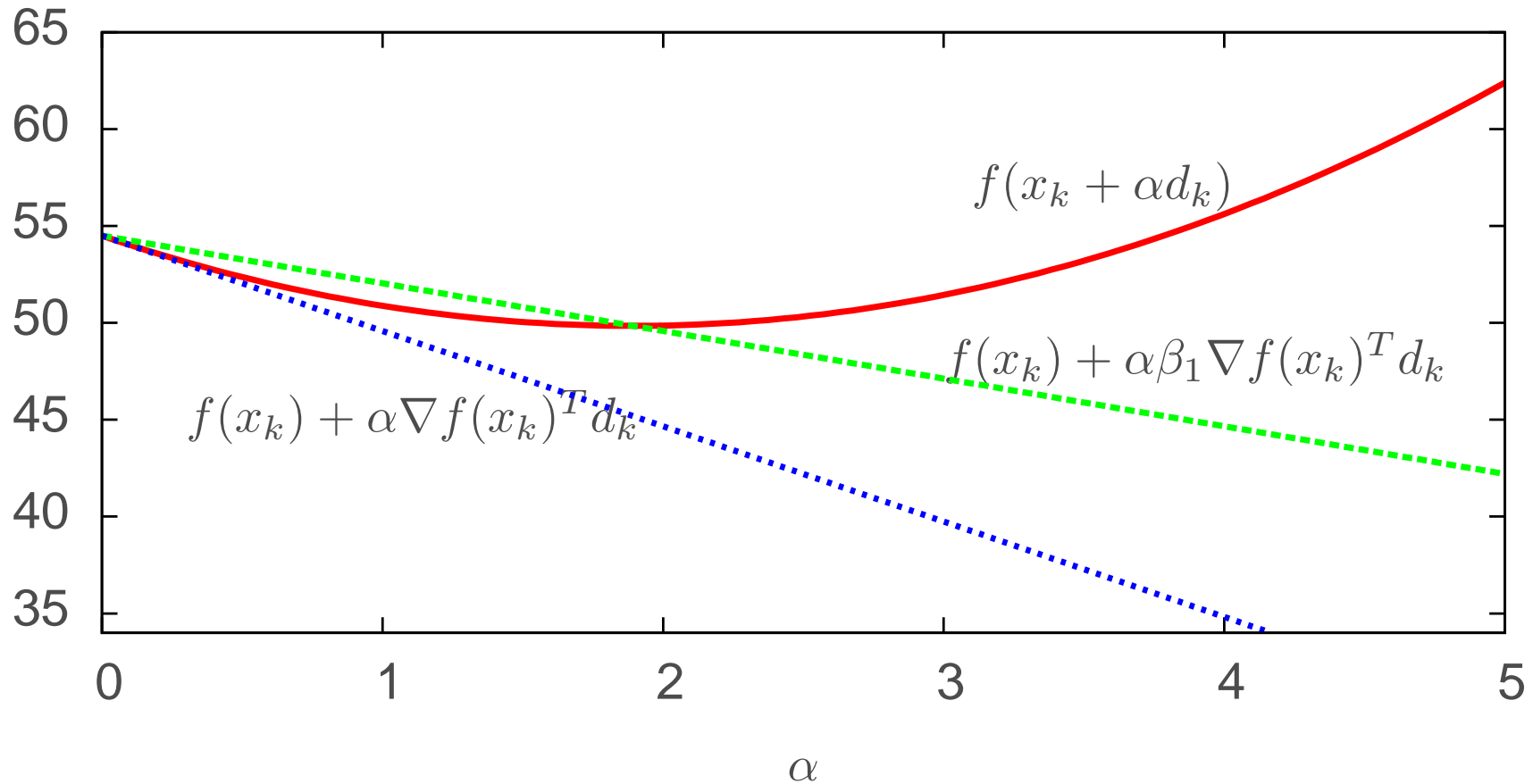
avec  $0 < \beta_1 < 1$ .

Cette condition s'appelle la première condition de Wolfe

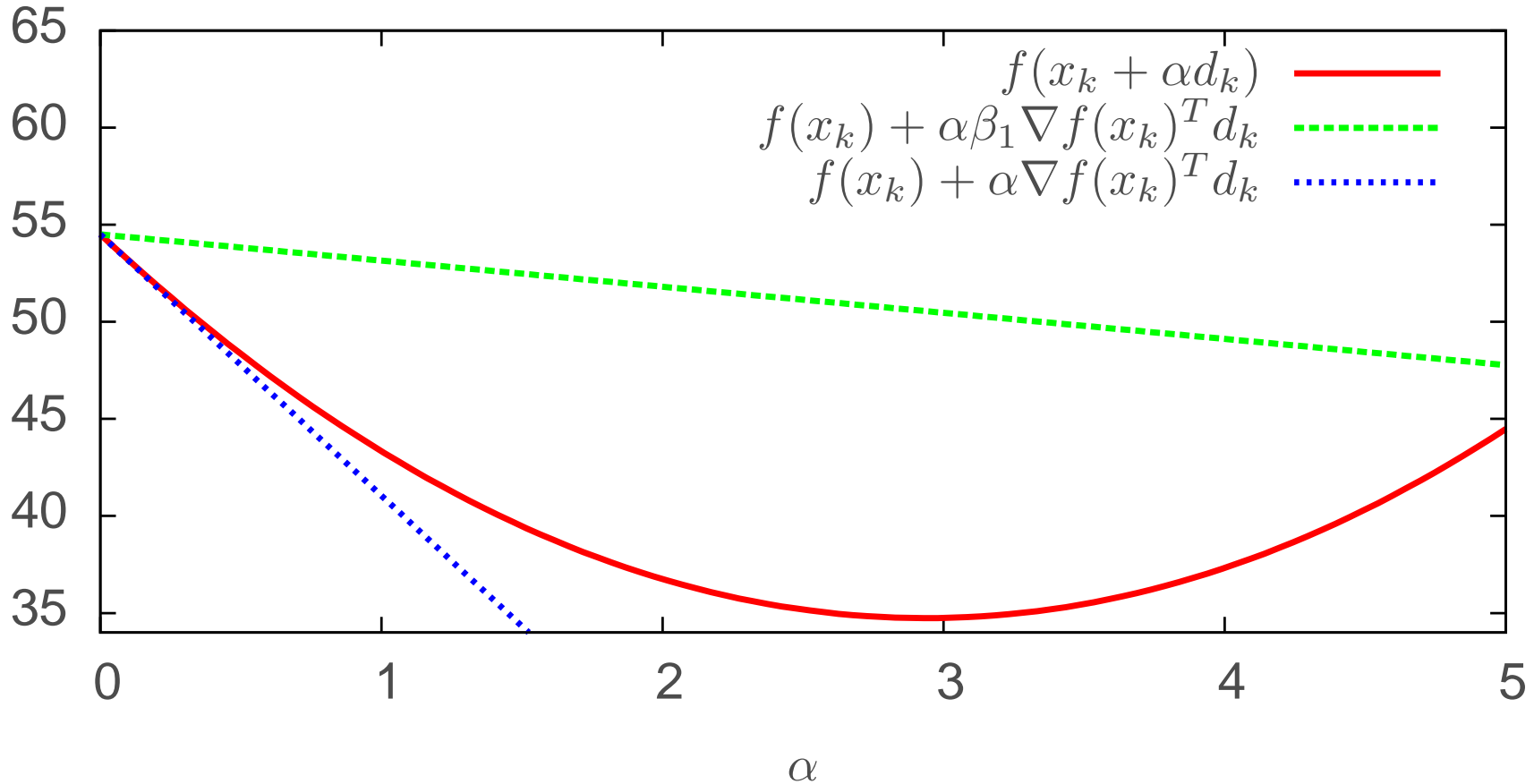
# Diminution suffisante $\beta_1 = 0.5$



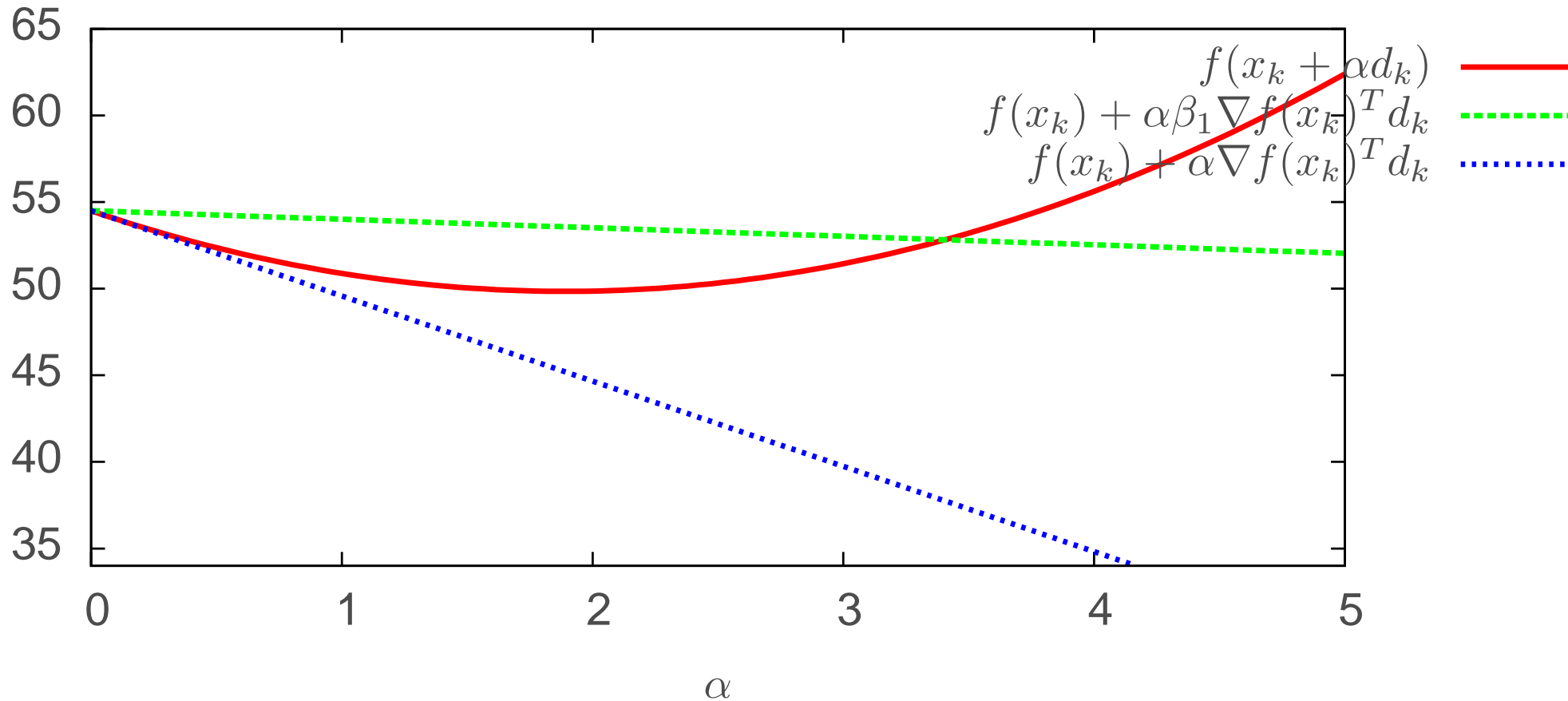
# Diminution suffisante $\beta_1 = 0.5$



# Diminution suffisante $\beta_1 = 0.1$



# Diminution suffisante $\beta_1 = 0.1$



# Choix du pas

---

- Exemple :  $f(x) = x^2$
- Appliquons l'algorithme avec  $x_0 = 2$ , et

$$\begin{aligned}D_k &= 1/2x_k \\ \alpha_k &= 2^{-k-1}.\end{aligned}$$

- $D_k$  est bien (défini) positif pour tout  $k$ .
- $\nabla f(x_k) = 2x_k \Rightarrow d_k = -D_k \nabla f(x_k) = -1$
- La méthode s'écrit

$$x_{k+1} = x_k - 2^{-k-1}$$

# Choix du pas

---

Nous avons que

$$x_k = 1 + 2^{-k}.$$

(p. 269)

Dès lors

$$f(x_{k+1}) < f(x_k)$$

Cependant,

$$\lim_{k \rightarrow \infty} x_k = 1 \neq 0$$

$k$	$x_k$	$d_k$	$\alpha_k$
0	+2.000000e+00	-1	+5.000000e-01
1	+1.500000e+00	-1	+2.500000e-01
2	+1.250000e+00	-1	+1.250000e-01
3	+1.125000e+00	-1	+6.250000e-02
4	+1.062500e+00	-1	+3.125000e-02
5	+1.031250e+00	-1	+1.562500e-02
⋮			
46	+1.000000e+00	-1	+7.105427e-15
47	+1.000000e+00	-1	+3.552714e-15
48	+1.000000e+00	-1	+1.776357e-15
49	+1.000000e+00	-1	+8.881784e-16
50	+1.000000e+00	-1	+4.440892e-16



# Choix du pas

---

Pourquoi cela ne fonctionne pas ?

- Dégénérescence
- Pas trop petits

Notion de progrès suffisant

- $\nabla f(x_k)^T d_k < 0$
- Si  $\alpha_k$  minimum dans la direction alors  $\nabla f(x_k + \alpha_k d_k)^T d_k = 0$
- La dérivée directionnelle **augmente**

# Choix du pas

## Progrès suffisant : seconde condition de Wolfe

Soient  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction différentiable, un point  $x_k \in \mathbb{R}^n$ , une direction (de descente)  $d_k \in \mathbb{R}^n$  telle que  $\nabla f(x_k)^T d_k < 0$  et un pas  $\alpha_k \in \mathbb{R}$ ,  $\alpha_k > 0$ .

On dira que le point  $x_k + \alpha_k d_k$  apporte un progrès suffisant par rapport à  $x_k$  si

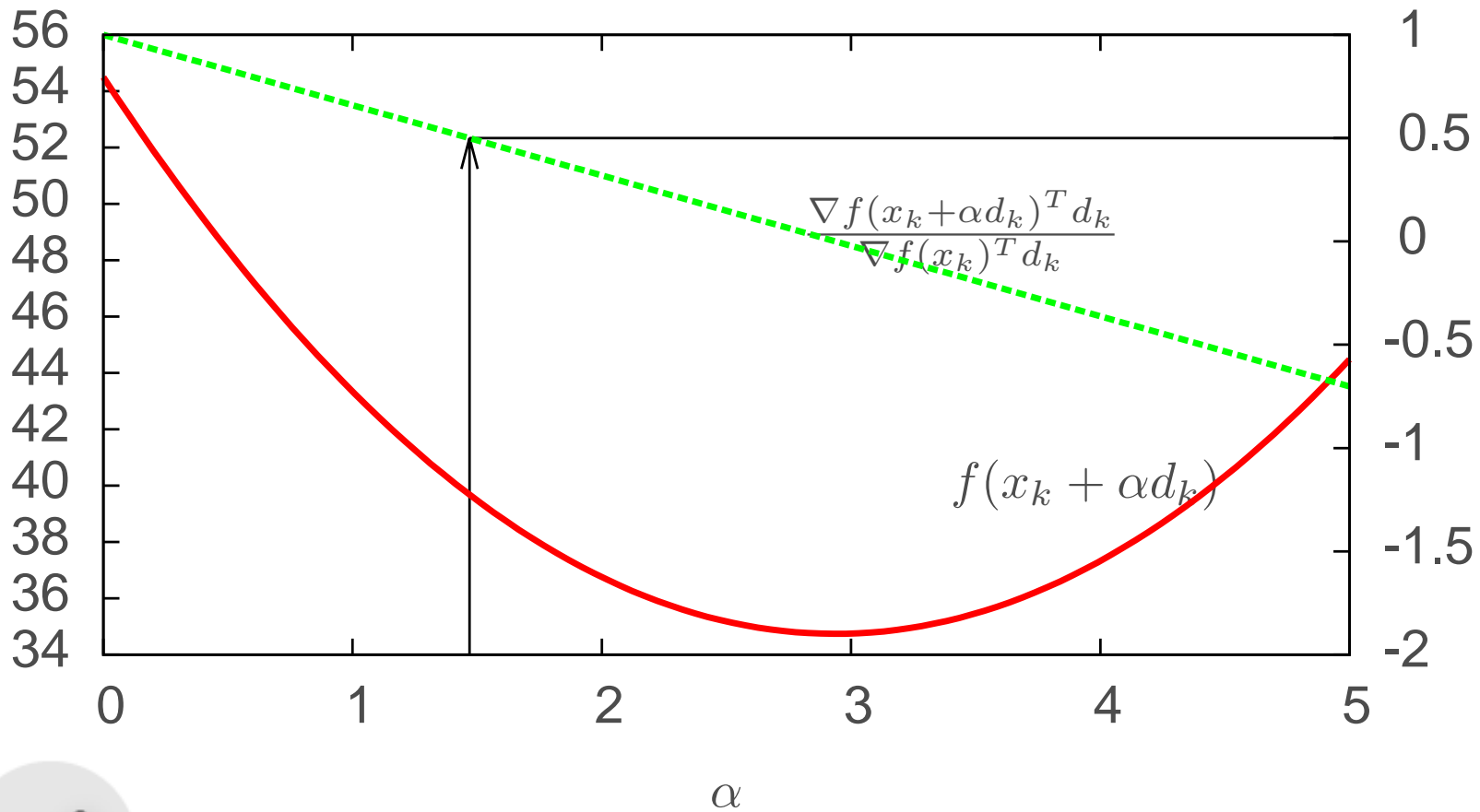
$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq \beta_2 \nabla f(x_k)^T d_k,$$

avec  $0 < \beta_2 < 1$ .

Cette condition s'appelle la seconde condition de Wolfe.

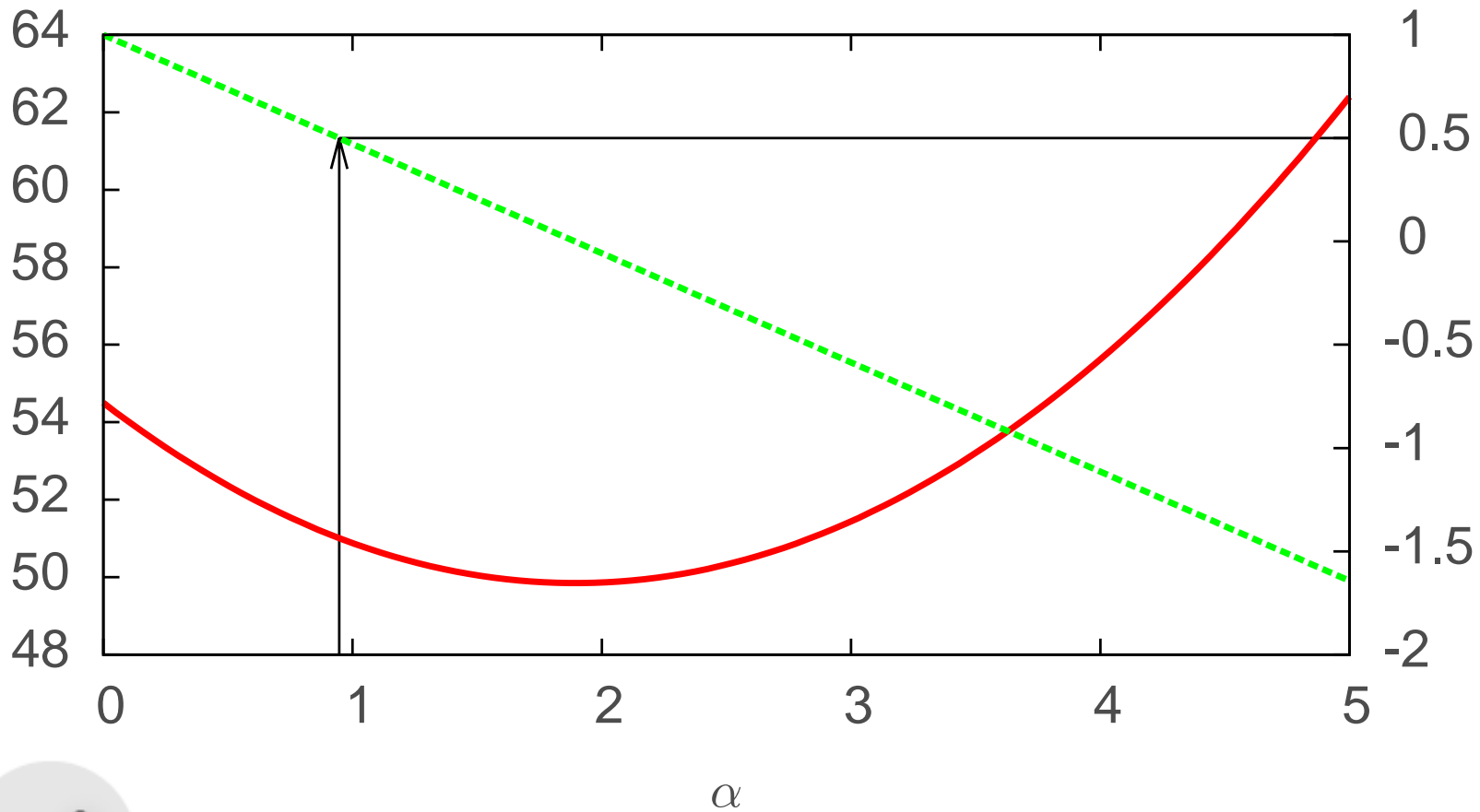
# Diminution suffisante $\beta_2 = 0.5$

$$d_k = (-10/\sqrt{181} \quad -9/\sqrt{181})^T \quad \alpha \geq 1.4687$$



# Diminution suffisante $\beta_2 = 0.5$

$$d_k = (-2/\sqrt{5} \quad 1/\sqrt{5})^T \quad \alpha \geq 0.94603$$



# Conditions de Wolfe

---

**Validité des conditions de Wolfe** Soient  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction différentiable, un point  $x_k \in \mathbb{R}^n$  et une direction (de descente)  $d_k \in \mathbb{R}^n$  telle que  $\nabla f(x_k)^T d_k < 0$  et  $f$  est bornée inférieurement dans la direction  $d_k$ , c'est-à-dire il existe  $f_0$  tel que  $f(x_k + \alpha d_k) \geq f_0$  pour tout  $\alpha \geq 0$ .

Si  $0 < \beta_1 < 1$ , il existe  $\eta$  tel que la première condition de Wolfe soit vérifiée pour tout  $\alpha_k \leq \eta$ . De plus, si  $0 < \beta_1 < \beta_2 < 1$ , il existe  $\alpha_2 > 0$  tel que les deux conditions de Wolfe soient toutes deux vérifiées.

(p. 271)

# Rappel

---

**Taylor au premier ordre** Soit  $f : \mathbb{R}^n \longrightarrow \mathbb{R}$  une fonction différentiable sur une sphère ouverte  $S$  centrée en  $x$ . Alors,

- pour tout  $d$  tel que  $x + d \in S$ , on a

$$f(x + d) = f(x) + d^T \nabla f(x) + o(\|d\|),$$

- pour tout  $d$  tel que  $x + d \in S$ , il existe  $\alpha \in [0, 1]$  tel que

$$f(x + d) = f(x) + d^T \nabla f(x + \alpha d).$$

(sans preuve)

# Algorithme : Recherche linéaire

---

## Objectif

Trouver un pas  $\alpha^*$  tel que les conditions de Wolfe soient vérifiées.

## Input

- La fonction  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  différentiable;
- Le gradient de la fonction  $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ ;
- Un vecteur  $x \in \mathbb{R}^n$ ;
- Une direction de descente  $d$  telle que  $\nabla f(x)^T d < 0$ ;
- Une première approximation de la solution  $\alpha_0 > 0$ .

# Algorithme : Recherche linéaire

---

## Input (suite)

- Des paramètres  $\beta_1$  et  $\beta_2$  tels que  $0 < \beta_1 < \beta_2 < 1$ .
- Un paramètre  $\lambda > 1$ .

## Output

Un pas  $\alpha^*$  tel que les conditions de Wolfe soient vérifiées.



# Algorithme : Recherche linéaire

---

## Initialisation

$$i = 0, \alpha_\ell = 0, \alpha_r = +\infty.$$

## Itérations

- Si  $\alpha_i$  vérifie les conditions, alors  $\alpha^* = \alpha_i$ . STOP.
- Si  $\alpha_i$  viole Wolfe-1, i.e.  
 $f(x_k + \alpha_k d_k) > f(x_k) + \alpha_k \beta_1 \nabla f(x_k)^T d_k$ , alors le pas est trop long et

$$\begin{aligned}\alpha_r &= \alpha_i \\ \alpha_{i+1} &= \frac{\alpha_\ell + \alpha_r}{2}\end{aligned}$$

# Algorithme : Recherche linéaire

## Itérations

- Si  $\alpha_i$  ne viole pas Wolfe-1 et viole Wolfe-2, i.e.

$$\nabla f(x + \alpha_i d)^T d < \beta_2 \nabla f(x)^T d$$

alors le pas est trop court et

$$\alpha_{i+1} = \begin{cases} \alpha_i & \text{si } \alpha_r < +\infty \\ \lambda \alpha_i & \text{sinon} \end{cases}$$

- $i = i+1$

# Algorithme : Recherche linéaire

$$f(x) = \frac{1}{2}x_1^2 + \frac{9}{2}x_2^2 \quad x = \begin{pmatrix} 10 \\ 1 \end{pmatrix} \quad d = \begin{pmatrix} \frac{-2}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} \end{pmatrix}$$

$$\alpha_0 = 10^{-3} \quad \beta_1 = 0.3 \quad \beta_2 = 0.7 \quad \lambda = 20.$$

$\alpha_i$	$\alpha_\ell$	$\alpha_r$	Cond. violée
1.000000000e-03	0.000000000e+00	9.999990000e+05	Wolfe-2
2.000000000e-02	1.000000000e-03	9.999990000e+05	Wolfe-2
4.000000000e-01	2.000000000e-02	9.999990000e+05	Wolfe-2
8.000000000e+00	4.000000000e-01	9.999990000e+05	Wolfe-1
4.200000000e+00	4.000000000e-01	8.000000000e+00	Wolfe-1
2.300000000e+00	4.000000000e-01	4.200000000e+00	—

# Plus forte pente

---

- En général, cet algorithme ne devrait pas être utilisé
- Décrivons-le pour pouvoir le comparer par la suite aux autres
- Utilisons la recherche linéaire

# Algorithme : Plus forte pente

---

## Objectif

Trouver une approximation de la solution du problème

$$\min_{x \in \mathbb{R}^n} f(x).$$

## Input

- La fonction  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  différentiable;
- Le gradient de la fonction  $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ ;
- Une première approximation de la solution  $x_0 \in \mathbb{R}^n$ ;
- La précision demandée  $\varepsilon \in \mathbb{R}, \varepsilon > 0$ .

# Algorithme : Plus forte pente

---

## Output

Une approximation de la solution  $x^* \in \mathbb{R}$

## Initialisation

$$k = 0$$

# Algorithme : Plus forte pente

---

## Itérations

1.  $d_k = -\nabla f(x_k)$ ,
2. Déterminer  $\alpha_k$  en appliquant la recherche linéaire avec  $\alpha_0 = 1$ .
3.  $x_{k+1} = x_k + \alpha_k d_k$ ,
4.  $k = k + 1$ .

## Critère d'arrêt

Si  $\|\nabla f(x_k)\| \leq \varepsilon$ , alors  $x^* = x_k$ .

# Méthode de Newton

---

- Combiner les idées de
  1. plus forte pente préconditionnée
  2. Newton
  3. recherche linéaire
- Itération de Newton pure

$$x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k),$$

- Itération de plus forte pente préconditionnée

$$x_{k+1} = x_k - \alpha_k D_k \nabla f(x_k),$$

- Si  $\nabla^2 f(x_k)^{-1}$  déf. positive, et  $\alpha_k = 1$  acceptable, itérations équivalentes.



# Méthode de Newton

---

- Si  $\alpha_k = 1$  non acceptable, **algorithme de recherche linéaire**
- Si  $\nabla^2 f(x_k)^{-1}$  non définie positive, définir

$$D_k = (\nabla^2 f(x_k) + E)^{-1}$$

avec  $E$  telle que  $D_k$  soit définie positive.

# Algorithme : Cholesky modifiée

---

## Objectif

Modifier une matrice afin de la rendre définie positive.

## Input

Une matrice symétrique  $A \in \mathbb{R}^{n \times n}$

## Output

Une matrice triangulaire inférieure  $L$  et  $\tau \geq 0$  tels que  $A + \tau I = LL^T$ .

## Initialisation

1.  $k=0$ ;
2. Si  $\min_i a_{ii} > 0$ , alors  $\tau_k = 0$ . Sinon,  $\tau_k = \frac{1}{2} \|A\|_F$ ;

# Algorithme : Cholesky modifiée

---

## Itérations

1. Calculer la factorisation de Cholesky de  $LL^T = A + \tau I$ .
2. Si factorisation réussie, STOP.
3. Sinon,  $\tau_{k+1} = \max(2\tau_k, \frac{1}{2}\|A\|_F)$
4.  $k = k + 1$ .

# Algorithme : Newton avec recherche linéaire

---

## Objectif

Trouver une approximation d'un minimum local du problème

$$\min_{x \in \mathbb{R}^n} f(x).$$

## Input

- La fonction  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  différentiable;
- Le gradient de la fonction  $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ ;
- Le hessien de la fonction  $\nabla^2 f : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ ;
- Une première approximation de la solution  $x_0 \in \mathbb{R}^n$ ;
- La précision demandée  $\varepsilon \in \mathbb{R}, \varepsilon > 0$ .

# Algorithme : Newton avec recherche linéaire

---

## Output

Une approximation de la solution  $x^* \in \mathbb{R}$

## Initialisation

$$k = 0$$

# Algorithme : Newton avec recherche linéaire

---

## Itérations

- Calculer une matrice triangulaire inférieure  $L_k$  et  $\tau$  tels que

$$L_k L_k^T = \nabla^2 f(x_k) + \tau I,$$

en utilisant l'algorithme précédent

- Trouver  $z_k$  en résolvant le système triangulaire  $L_k z_k = \nabla f(x_k)$ .
- Trouver  $d_k$  en résolvant le système triangulaire  $L_k^T d_k = -z_k$ .

# Algorithme : Newton avec recherche linéaire

---

## Itérations (suite)

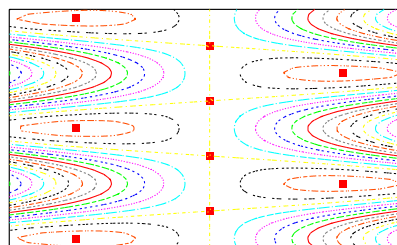
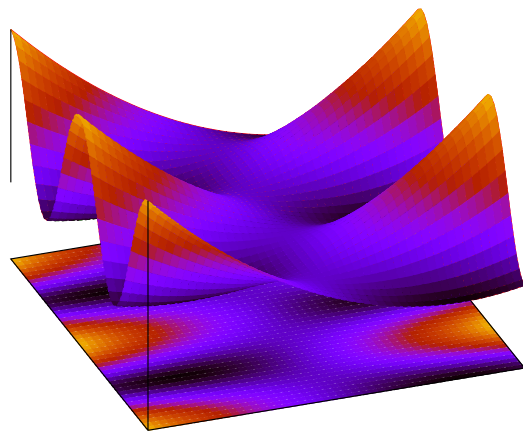
- Déterminer  $\alpha_k$  en appliquant la recherche linéaire avec  $\alpha_0 = 1$ .
- $x_{k+1} = x_k + \alpha_k d_k$ .
- $k = k + 1$ .

## Critère d'arrêt

Si  $\|\nabla f(x_k)\| \leq \varepsilon$ , alors  $x^* = x_k$ .

# Newton avec recherche linéaire

$$\min f(x_1, x_2) = \frac{1}{2}x_1^2 + x_1 \cos x_2,$$



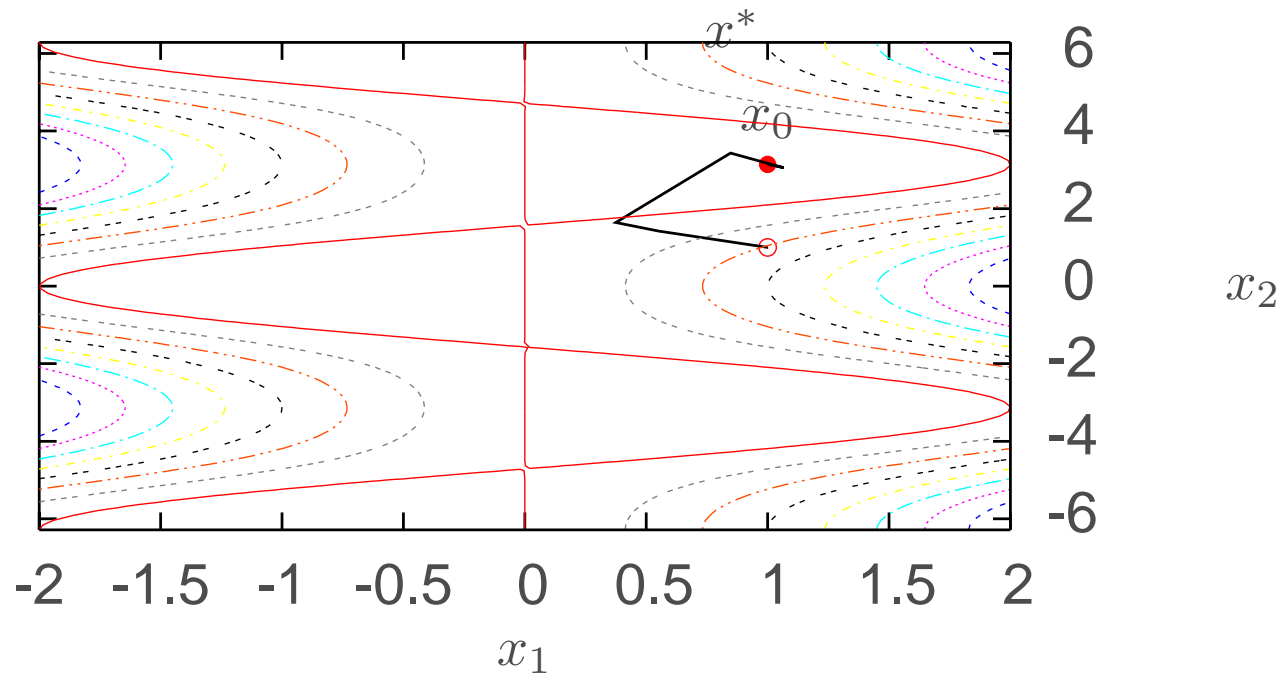
Point de départ  $x_0 = (1 \ 1)^T$ .



# Newton avec recherche linéaire

Solution:

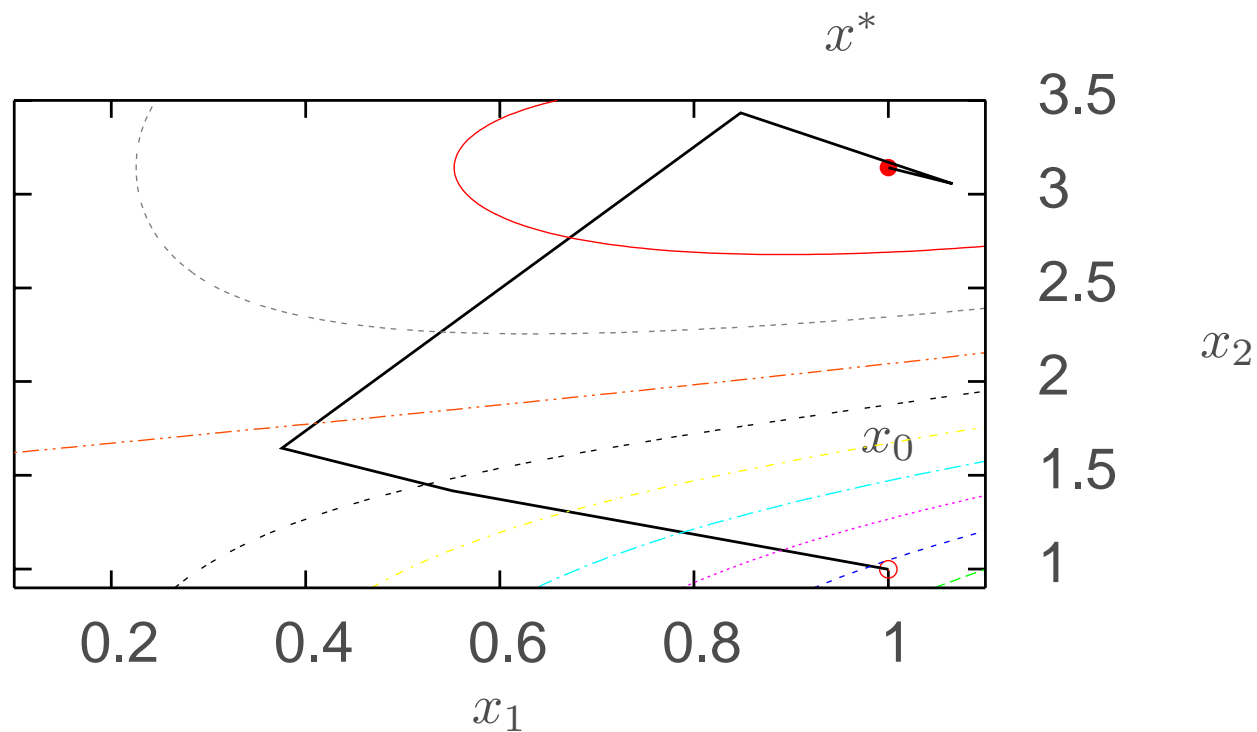
$$x^* = \begin{pmatrix} 1 \\ \pi \end{pmatrix} \quad \nabla f(x^*) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \nabla^2 f(x^*) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$



# Newton avec recherche linéaire

Solution:

$$x^* = \begin{pmatrix} 1 \\ \pi \end{pmatrix} \quad \nabla f(x^*) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \nabla^2 f(x^*) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$



# Newton avec recherche linéaire

$k$	$f(x_k)$	$\ \nabla f(x_k)\ $	$\alpha_k$	$\tau$
0	1.04030231e+00	1.75516512e+00		
1	2.34942031e-01	8.88574897e-01	1	1.64562250e+00
2	4.21849003e-02	4.80063696e-01	1	1.72091923e+00
3	-4.52738278e-01	2.67168927e-01	3	8.64490594e-01
4	-4.93913638e-01	1.14762780e-01	1	0.00000000e+00
5	-4.99982955e-01	5.85174623e-03	1	0.00000000e+00
6	-5.00000000e-01	1.94633135e-05	1	0.00000000e+00
7	-5.00000000e-01	2.18521663e-10	1	0.00000000e+00
8	-5.00000000e-01	1.22460635e-16	1	0.00000000e+00