Introduction à l'optimisation

Stéphane Canu
stephane.canu@litislab.eu

ASI 4 - Programmation linéaire et le simplexe

October 8, 2014

Plan

1. Programmation linéaire et le simplexe
   - Formulation Standard de la programmation linéaire
     - Exemples de programmation linéaire
     - Formulation Standard de la programmation linéaire
     - Domaine réalisable
     - Solution de base
     - Domaine réalisable et solution de base
   - Le tableau de Simplexe
   - L'algorithme du simplexe
     - illustration sur un exemple
     - L'algorithme
     - L'initialisation du simplexe
2. comment trouver la forme canonique d'un problème
   - la régession $L1$
3. Dual d'un programme linéaire

Un exemple de programmation linéaire

Calculez le mélange le moins cher à donner à nos cobayes

Afin d'assurer une bonne santé de cobayes, il faut les nourrir en leur donnant un minimum de 24 grammes de lipides, 36 grammes de glucides et 4 grammes de protéines par jour. Il ne faut pas leur donner plus de 5 onces de nourriture.

Nous disposions de deux sources d'alimentation. Les croquettes royal cobaye qui contiennent 8 grammes de lipides, 12 grammes de glucides et 2 grammes de protéines par once et qui coutent 3 euros les 10 onces, et les boulettes « vite se casse » qui contiennent 12 grammes de lipides, 12 grammes de glucides et 1 grammes de protéines par once et qui coutent 2 euros les 10 onces.

Un exemple de programmation linéaire

Calculez le mélange le moins cher à donner à nos cobayes

Afin d'assurer une bonne santé de cobayes, il faut les nourrir en leur donnant un minimum de 24 grammes de lipides, 36 grammes de glucides et 4 grammes de protéines par jour. Il ne faut pas leur donner plus de 5 onces de nourriture.
Nous disposions de deux sources d'alimentation. Les croquettes royal cobaye qui contiennent 8 grammes de lipides, 12 grammes de glucides et 2 grammes de protéines par once et qui coutent 3 euros les 10 onces, et les boulettes « vite se casse » qui contiennent 12 grammes de lipides, 12 grammes de glucides et 1 grammes de protéines par once et qui coutent 2 euros les 10 onces.

inconnues
- quantité de royal cobaye x_1 en once
- quantité de « vite se casse » x_2 en once

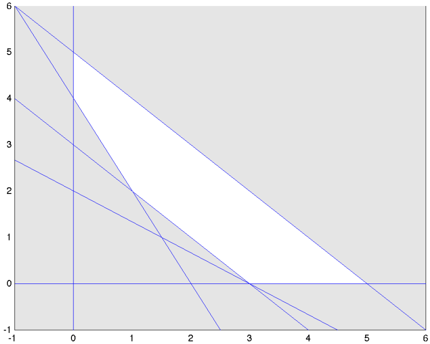
cout minimiser par rapport à $\mathbf{x} = (x_1, x_2)$ la fonction cout $J(\mathbf{x}) = 30x_1 + 20x_2$

contraintes
- lipides $8x_1 + 12x_2 \geq 24$
- glucides $12x_1 + 12x_2 \geq 36$
- protéines $2x_1 + x_2 \geq 4$
- diététique $x_1 + x_2 \leq 5$
- ces quantités sont positives : $x_1 \geq 0$ et $x_2 \geq 0$

Domaine admissible

Contraintes :

- lipides $8x_1 + 12x_2 \geq 24$
- glucides $12x_1 + 12x_2 \geq 36$
- protéines $2x_1 + x_2 \geq 4$
- diététique $x_1 + x_2 \leq 5$
- ces quantités sont positives : $x_1 \geq 0$ et $x_2 \geq 0$



Cout

$$J(\mathbf{x}) = 30x_1 + 20x_2$$

Programmation linéaire avec Matlab

Contraintes :

- lipides $8x_1 + 12x_2 \geq 24$
- glucides $12x_1 + 12x_2 \geq 36$
- protéines $2x_1 + x_2 \geq 4$
- diététique $x_1 + x_2 \leq 5$
- ces quantités sont positives :

  $x_1 \geq 0$ et $x_2 \geq 0$

Cout

$$J(\mathbf{x}) = 30x_1 + 20x_2$$

CVX: Matlab Software for Disciplined Convex Programming
Version 2.0 (beta), November 2012, Build 883

cvxr.com/cvx/

```
cvx_begin
    variables x_1 x_2
    dual variables lip glu pro die p1 p2;
    minimize( 30*x_1 +  20*x_2 )
    subject to
         lip : 8 *x_1 + 12* x_2 >= 24;
         glu : 12 *x_1 + 12 *x_2 >= 36;
         pro : 2 *x_1 +  x_2 >= 4;
         die : x_1 +  x_2 <= 5;
         p1 : x_1 >=0;
         p2 : x_2 >=0;
cvx_end
```

Programmation linéaire - version matricielle

Cout avec

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \text{ et } \mathbf{f} = \begin{pmatrix} 30 \\ 20 \end{pmatrix}$$

$$\begin{aligned} J(\mathbf{x}) &= 30x_1 + 20x_2 \\ &= \mathbf{f}^\top \mathbf{x} \end{aligned}$$

Contraintes avec

$$\mathbf{b} = \begin{pmatrix} 24 \\ 36 \\ 4 \\ -5 \end{pmatrix} \text{ et } A = \begin{pmatrix} 8 & 12 \\ 12 & 12 \\ 2 & 1 \\ -1 & -1 \end{pmatrix}$$

lipides $8x_1 + 12x_2 \geq 24$
glucides $12x_1 + 12x_2 \geq 36$
protéines $2x_1 + x_2 \geq 4$
diététique $x_1 + x_2 \leq 5$

$\rightarrow A\mathbf{x} \geq \mathbf{b}$

```
cvx_begin
    variables x_1 x_2
    dual variables lip glu pro die p1 p2;
    minimize( 30*x_1 +  20*x_2 )
    subject to
         lip : 8 *x_1 + 12* x_2 >= 24;
         glu : 12 *x_1 + 12 *x_2 >= 36;
         pro : 2 *x_1 +  x_2 >= 4;
         die : x_1 +  x_2 <= 5;
         p1 : x_1 >=0;
         p2 : x_2 >=0;
cvx_end

f = [30 ; 20];
b = [24;36;4;-5];
A = [8 12 ; 12 12 ; 2 1 ; -1 -1];
cvx_begin
    variable x(2)
    dual variables a p;
    minimize( f'*x )
    subject to
         a : A*x >= b
         p : x >= 0;
cvx_end
```

Programmation linéaire forme standard

Matlab Optimization toolbox

```
X = LINPROG(f,A,b) attempts
to solve the linear programming problem:

min f'*x    subject to:   A*x <= b
  x

X = LINPROG(f,A,b,Aeq,beq) solves
the problem above while additionally
satisfying the equality constraints
Aeq*x = beq.

X = LINPROG(f,A,b,Aeq,beq,LB,UB)
defines a set of lower and upper
bounds on the design variables, X
...

f = [30 ; 20];
b = [24;36;4;5];
A = [8 12 ; 12 12 ; 2 1 ; 1 1];
[x c e o p] = linprog(f,-A,-b,[],[],0*f);
```

CVX: Matlab Software for Disciplined Convex Programming

Version 2.0 (beta), November 2012, Build 883

cvxr.com/cvx/

```
cvx_begin
    variable x(n)
    dual variables a p;
    minimize( f'*x  )
    subject to
        a : A*x == b;
        p : x >= 0;
cvx_end
```

Un exemple de programmation linéaire

- trois usines de fabrication (A, B et C)
- quatre magasins (Wini (W), Xana (X), Yerville (Y) et Zanzibar (Z)).
- Comment minimiser ses couts de transports ?

Usine	Production
A	400
B	1500
C	900
Total	2800

Magasins	Demande
W	700
X	600
Y	1000
Z	500
Total	2800

.

Nous avons aussi la matrice des couts de transports :

	W	X	Y	Z
A	20	40	70	50
B	100	60	90	80
C	10	110	30	200

Un exemple de programmation linéaire

Le problème de transport :

- d ports de départ avec chacun une quantité $a_i, i = 1, d$ à envoyer
- r ports d'arrivée avec chacun une quantité $b_j, j = 1, r$ à recevoir
- c_{ij} le cout de transport d'un port a_i vers un port b_j est connu
- trouver comment affréter les bateaux pour un cout minimal

Un exemple de programmation linéaire

Le problème de transport :

- d ports de départ avec chacun une quantité $a_i, i = 1, d$ à envoyer
- r ports d'arrivée avec chacun une quantité $b_j, j = 1, r$ à recevoir
- c_{ij} le cout de transport d'un port a_i vers un port b_j est connu
- trouver comment affréter les bateaux pour un cout minimal

On introduit les variables x_{ij} : la quantité de produit transporté de i vers j.

$$\left\{\begin{array}{ll} \displaystyle\min_{\mathbf{z}\in\mathbb{R}^{d\times r}} & \displaystyle\sum_{i=1}^{d}\sum_{j=1}^{r} c_{ij}x_{ij} \\ \text{avec} & \displaystyle\sum_{j} x_{ij} = a_i; \quad i = 1, d \\ & \displaystyle\sum_{i} x_{ij} = b_j; \quad j = 1, r \\ \text{et} & x_{ij} \geq 0 \end{array}\right.$$

Example : a vous

L'entreprise Cannes United dispose de 3 usines de fabrication (A,B et C). Il souhaite minimiser ses couts de transports entre les sites de fabrication et ses 4 magasins (W,X,Y et Z).

Usine	Production
A	400
B	1500
C	900
Total	2800

Magasins	Demande
W	700
X	600
Y	1000
Z	500
Total	2800

.

Nous avons aussi la matrice des couts de transports :

	W	X	Y	Z
A	20	40	30	50
B	10	20	20	10
C	10	10	30	20

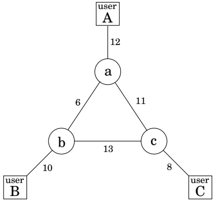
Par exemple, le cout de transport vers Y d'une canne fabriquée à A est de 30 euros.

1. formuler le problème comme un programme linéaire,
2. donner la solution du problème
3. donner la formulation duale

Un exemple de programmation linéaire

Maximisation d'un flot :

- prix : A-B 3 €, B-C 2€ et A-C 4€
- par contrat, chaque connexion doit au moins recevoir 2 unités
- trouver comment maximiser les revenus sur le réseau

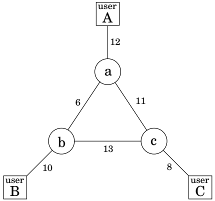


voir Algorithms, S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, 2006, page 195

Un exemple de programmation linéaire

Maximisation d'un flot :

- prix : A-B 3 €, B-C 2€ et A-C 4€
- par contrat, chaque connexion doit au moins recevoir 2 unités
- trouver comment maximiser les revenus sur le réseau



On introduit les variables x_i : x_1 (AB direct),x_2 (AB long), x_3 (BC direct), x_4 (BC long), x_5 (AC direct), x_6 (AC long)

$$
\begin{cases}
\max\limits_{\mathbf{x}\in\mathbb{R}^6} & 3x_1 + 3x_2 + 2x_3 + 2x_4 + 4x_5 + 4x_6 & \\
\text{avec} & x_1 + x_2 + x_3 + x_4 \leq 10 & \text{arrête B-b} \\
 & x_1 + x_4 + x_6 \leq 6 & \text{arrête a-b} \\
 & x_1 + x_4 + x_6 \leq 6 & \text{arrête a-b} \\
 & x_1 + x_2 \geq 2 & \text{traffic minimum} \\
 & \dots & \\
\text{et} & x_i \geq 0 &
\end{cases}
$$

voir Algorithms, S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, 2006, page 195

Programmation linéaire : formulation Standard

PL : forme standard

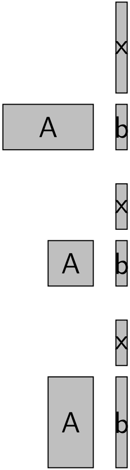
$$\begin{cases} \min\limits_{\mathbf{z}\in\mathbb{R}^m} & \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} = \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0 \end{cases}$$

- cout linéaire,
- contraintes linéaires
- $A \in \mathcal{M}(p, m)$

- m **variables (inconnues)** $\mathbf{z} = (z_1, z_2, \ldots, z_m)^\top$
- vecteur des couts (connu) $\mathbf{c} = (c_1, c_2, \ldots, c_m)^\top$
- la fonction objectif (de perte) $f(\mathbf{z}) = \mathbf{c}^\top \mathbf{z}$
- p **contraintes** : $p < m$ (moins de contraintes que d'inconnues)
- A une matrice p lignes (contraintes) et m colonnes (inconnues)
- $\mathbf{b} = (b_1, b_2, \ldots, b_p)^\top$ vecteur des *«second membres »*

Programmation linéaire : la nature du problème

$$\left\{\begin{array}{ll} \min\limits_{\mathbf{z}\in\mathbb{R}^m} & \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} = \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0 \end{array}\right.$$



Sous déterminé : il y a plus d'inconnues que d'équations

il y a autant d'équtions que d'inconnues

sur déterminé : il y a plus d'équations que d'inconnues

$A \in \mathcal{M}(p, m)$

Dans le cas qui nous intéresse le système est sous déterminé et la matrice A à plus de colonnes que de lignes : $p < m$

PL : réduction à la forme Standard

forme standard : cout linéaire, contraintes linéaires

$$\begin{cases} \min\limits_{\mathbf{z}\in\mathbb{R}^m} & \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} = \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0 \end{cases} \qquad A \in \mathcal{M}(p, m)$$

un exemple simple $m = 2, p = 1$

$$\begin{cases} \min\limits_{\mathbf{z}\in\mathbb{R}^2} & az_1 + bz_2 \\ \text{avec} & z_1 + z_2 \leq 1 \\ \text{et} & z_1 \geq 0, z_2 \geq 0 \end{cases}$$

il y a trois solutions selon les valeurs de (a, b).

$$\begin{cases} \min\limits_{\mathbf{z}\in\mathbb{R}^2} & az_1 + bz_2 + 0z_3 \\ \text{avec} & z_1 + z_2 + z_3 = 1 \\ \text{et} & z_1 \geq 0, z_2 \geq 0, z_3 \geq 0 \end{cases}$$

PL : réduction à la forme Standard

forme standard : cout linéaire, contraintes linéaires

$$\begin{cases} \min\limits_{\mathbf{z}\in\mathbb{R}^m} & \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} = \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0 \end{cases} \qquad A \in \mathcal{M}(p, m)$$

un exemple simple $m = 2, p = 1$

$$\begin{cases} \min\limits_{\mathbf{z}\in\mathbb{R}^2} & az_1 + bz_2 \\ \text{avec} & z_1 + z_2 \leq 1 \\ \text{et} & z_1 \geq 0, z_2 \geq 0 \end{cases}$$

il y a trois solutions selon les valeurs de (a, b).

$$\begin{cases} \min\limits_{\mathbf{z}\in\mathbb{R}^2} & az_1 + bz_2 + 0z_3 \\ \text{avec} & z_1 + z_2 + z_3 = 1 \\ \text{et} & z_1 \geq 0, z_2 \geq 0, z_3 \geq 0 \end{cases}$$

Formulation standard : des inégalités aux égalités

Introduction de nouvelles variables : des variables d'écart $\mathbf{e} \geq 0$

$$\mathbf{e} = (e_1, e_2, e_3, e_4)^\top$$

$$\begin{aligned} 8x_1 + 12x_2 &\geq 24 \\ 12x_1 + 12x_2 &\geq 36 \\ 2x_1 + x_2 &\geq 4 \\ x_1 + x_2 &\leq 5 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \end{aligned} \quad \Longrightarrow \quad \begin{aligned} 8x_1 + 12x_2 - e_1 &= 24 \\ 12x_1 + 12x_2 - e_2 &= 36 \\ 2x_1 + x_2 - e_3 &= 4 \\ x_1 + x_2 + e_4 &= 5 \\ x_1 \geq 0, e_1 \geq 0, e_2 &\geq 0 \\ x_2 \geq 0, e_3 \geq 0, e_4 &\geq 0 \end{aligned}$$

$$\left\{ \begin{array}{ll} \min\limits_{\mathbf{z} \in \mathbb{R}^m} & \mathbf{d}^\top \mathbf{z} \\ \text{avec} & B\mathbf{z} \leq \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0 \end{array} \right. \Longrightarrow \left\{ \begin{array}{ll} \min\limits_{\mathbf{z} \in \mathbb{R}^m, \mathbf{e} \in \mathbb{R}^p} & \mathbf{d}^\top \mathbf{z} \\ \text{avec} & B\mathbf{z} + \mathbf{e} = \mathbf{b} \\ \text{et} & \mathbf{z}, \mathbf{e} \geq 0 \end{array} \right. \Longrightarrow \left\{ \begin{array}{ll} \min\limits_{\mathbf{x} \in \mathbb{R}^{m+p}} & \mathbf{c}^\top \mathbf{x} \\ \text{avec} & A\mathbf{x} = \mathbf{b} \\ \text{et} & \mathbf{x} \geq 0 \end{array} \right.$$

$$\mathbf{x} = (\mathbf{z}; \mathbf{e}), \qquad \mathbf{c} = (\mathbf{d}; 0); \qquad A = \left[\begin{array}{c|c} B & I \end{array} \right]$$

PL : réduction à la forme Standard

La cas des variables non positives

$$\begin{cases} \min\limits_{x,y,z} & x + 3y + 4z \\ \text{avec} & x + 2y + z = 5 \\ & 2x + 3y + z = 6 \\ \text{et} & y \geq 0, z \geq 0 \end{cases}$$

Trois possibilités :

- on ne fait rien...
- on remplace x par la différence de deux variables contraintes : $x = x_p - x_m$ avec $x_p \geq 0$ et $x_m \geq 0$

$$\begin{cases} \min\limits_{x_p,x_m,y,z} & x_p - x_m + 3y + 4z \\ \text{avec} & x_p - x_m + 2y + z = 5 \\ & 2x_p - 2x_m + 3y + z = 6 \\ \text{et} & x_p \geq 0, x_m \geq 0, y \geq 0, z \geq 0 \end{cases}$$

- on élimine x à l'aide d'une contrainte

Example : Formulation duale

Construisons une combinaison linéaire des contraintes :

$$\forall y, z \geq 0 \qquad \left\{ \begin{array}{ccc} y^\top A x & = & y^\top b \\ z^\top x & \geq & 0 \end{array} \right.$$

$$y^\top A x + z^\top x \;\; \geq \;\; y^\top b$$

en choisant y et $z \geq 0$ tels que
$A^\top y + z = c$
on a $\qquad c^\top x \;\geq\; y^\top b$

```
cvx_begin
    variable x(n)
    dual variables y z;
    minimize( c'*x  )
    subject to
        y : A*x == b;
        z : x >= 0;
cvx_end

size(y),size(z)
[A'*y+z c]
b'*y
```

Définition : problème dual

$$\left\{ \begin{array}{ll} \displaystyle\max_{\mathbf{y} \in \mathbb{R}^p} & \mathbf{b}^\top \mathbf{y} \\ \text{avec} & A^\top \mathbf{y} \leq \mathbf{c} \end{array} \right.$$

Théorème de dualité

Définition : problème primal et problème dual

problème primal

$$\left\{\begin{array}{ll} \max\limits_{\mathbf{x}\in\mathbb{R}^m} & \mathbf{c}^\top\mathbf{x} \\ \text{avec} & A\mathbf{x} = \mathbf{b} \\ \text{et} & \mathbf{x} \geq 0 \end{array}\right.$$

problème dual .

$$\left\{\begin{array}{ll} \max\limits_{\mathbf{y}\in\mathbb{R}^p} & \mathbf{b}^\top\mathbf{y} \\ \text{avec} & A^\top\mathbf{y} \leq \mathbf{c} \end{array}\right.$$

Théorème de dualité

$$\mathbf{c}^\top\mathbf{x}^\star = \mathbf{b}^\top\mathbf{y}^\star$$

Programme linéaire dual

```
A = [8 12 ;12 12; 2 1 ;-1 -1 ];
b = [24;36;4;-5];
c = [30;20];

cvx_begin
   cvx_precision best
   variable x(2)
   dual variables de di diC
   minimize( c'*x )
   subject to
      de : A*x >= b;
      di : x >= 0;
cvx_end
```

```
A = [8 12 ;12 12; 2 1 ;-1 -1 ];
b = [24;36;4;-5];
c = [30;20];

cvx_begin
   cvx_precision best
   variable y(4)
   dual variables pe pi
   maximize( b'*y )
   subject to
      pe : A'*y <= c;
      pi : y >=0;
cvx_end
```

Plan

1. Programmation linéaire et le simplexe
   - Formulation Standard de la programmation linéaire
     - Exemples de programmation linéaire
     - Formulation Standard de la programmation linéaire
     - Domaine réalisable
     - Solution de base
     - Domaine réalisable et solution de base
   - Le tableau de Simplexe
   - L'algorithme du simplexe
     - illustration sur un exemple
     - L'algorithme
     - L'initialisation du simplexe
2. comment trouver la forme canonique d'un problème
   - la régession $L1$
3. Dual d'un programme linéaire

Domaine réalisable

$$\begin{cases} \min\limits_{\mathbf{z}\in\mathbb{R}^m} & \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} = \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0 \end{cases} \quad A \in \mathcal{M}(p, m)$$

Domaine réalisable

C'est l'ensemble des vecteurs vérifiant les contraintes

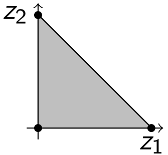
$$\mathcal{S}_a = \{\mathbf{z} \in \mathbb{R}^m | A\mathbf{z} = \mathbf{b};\ \mathbf{z} \geq 0\ \}$$

C'est un polytope (polyèdre) convexe

exemple :

$z_1 + z_2 \leq 1$
$z_1 \geq 0, z_2 \geq 0$

$z_1 + z_2 + z_3 = 1$
$z_1 \geq 0, z_2 \geq 0, z_3 \geq 0$



Solution de base

$$\begin{cases} \min\limits_{\mathbf{z}\in\mathbb{R}^m} & \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} = \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0 \end{cases} \qquad \begin{array}{l} A \in \mathcal{M}(p, m) \\ p < m \end{array}$$

Definition (Solution de Base : $\mathbf{z} = [\mathbf{z}_I ; \mathbf{z}_{\bar{I}}]$)

base ensemble I de p entiers distincts parmi $\{1, m\}$

matrice de base la matrice carrée $A_I = A(:, I)$ composée des p colonnes de A de la base I

solution de base un vecteur $\mathbf{z}_I \in \mathbb{R}^p$ vérifiant $A_I \mathbf{z}_I = \mathbf{b}$.
L'ensemble des solutions de base est noté $\mathcal{S}_I$

solution de base réalisable une solution de base dont toutes les composantes sont positive : $\mathbf{z}_I = A_I^{-1}\mathbf{b} \geq 0$

base réalisable une base dont la solution de base est réalisable

$$\boxed{\quad A \quad} = \boxed{\; A_I \;}\boxed{\; A_{\bar{I}} \;}$$

Exemples de Solutions de base

calculer les solution de base liées aux système suivant :

$$A\mathbf{z} = \mathbf{b} \qquad \left\{ \begin{array}{ll} z_1 + 2z_2 + z_3 + 2z_4 & = 3 \\ 2z_1 + z_2 + 5z_3 + 3z_4 & = 3 \\ 3z_1 + 4z_2 + 5z_3 + 7z_4 & = 7 \end{array} \right. \qquad A = \begin{pmatrix} 1 & 2 & 1 & 2 \\ 2 & 1 & 5 & 3 \\ 3 & 4 & 5 & 7 \end{pmatrix}$$

$$\mathcal{S}_l = \left\{ \begin{pmatrix} -2 \\ 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 4/3 \\ 1/3 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 4 \\ -1 \\ 0 \end{pmatrix} \right\}$$

L'ensemble des solutions de base réalisables :

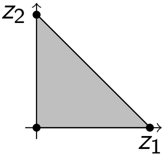
$$\mathcal{S}_r = \left\{ \begin{pmatrix} 4/3 \\ 1/3 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \right\}$$

Domaine réalisable et solution de base

exemple :

$z_1 + z_2 \leq 1$ $\qquad$ $z_1 + z_2 + z_3 = 1$

$z_1 \geq 0, z_2 \geq 0$ $\qquad$ $z_1 \geq 0, z_2 \geq 0, z_3 \geq 0$



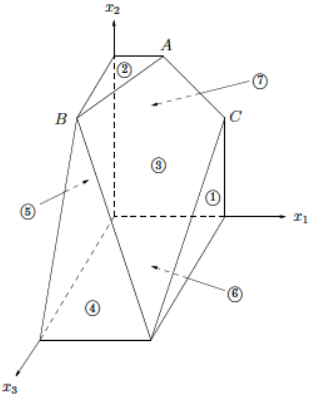
- Domaine réalisable est un sous espace vectoriel de taille potentiellement infinie

$$\mathrm{card}(\mathcal{S}_a) = \infty$$

- L'ensemble des solutions de Base est un ensemble fini

$$k = \mathrm{card}(\mathcal{S}_I) \leq \binom{p}{m} = \frac{m!}{p!(m-p)!}$$

Domaine réalisable et solution de base



$$\max \quad x_1 + 6x_2 + 13x_3$$

$$x_1 \leq 200 \qquad ①$$

$$x_2 \leq 300 \qquad ②$$

$$x_1 + x_2 + x_3 \leq 400 \qquad ③$$

$$x_2 + 3x_3 \leq 600 \qquad ④$$

$$x_1 \geq 0 \qquad ⑤$$

$$x_2 \geq 0 \qquad ⑥$$

$$x_3 \geq 0 \qquad ⑦$$

Algorithms 2006 S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, page 214

Théorème fondamental : l'optimum est une solution de base

$$\mathcal{P} \quad \begin{cases} \min\limits_{\mathbf{z}\in\mathbb{R}^m} & \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} = \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0 \end{cases} \qquad \begin{array}{l} A \in \mathcal{M}(p, m), p < m \\ rang(A) = p \end{array}$$

Theorem (Caractérisation de la solution du problème $\mathcal{P}$)

s'il existe une solution au problème $\mathcal{P}$ alors soit c'est une solution de base soit il existe une solution de base réalisable équivalente (de même cout)

<u>démonstration</u> : soit $\mathbf{z}^*$ la solution optimale vérifiant $A\mathbf{z} = \mathbf{b}$ et admettant ℓ composantes non nulle ($p \leq \ell \leq m$). Notons $\mathbf{z}_\ell$ le vecteur des composantes non nulles et A_ℓ la matrice des colonnes associée de la matrice A de sorte que $A\mathbf{z} = A_\ell \mathbf{z}_\ell = \mathbf{b}$ Les ℓ colonnes de la matrice A_ℓ sont linéairement liées et donc il existe des α_i tel que :

$$\alpha_1 A_1 + \alpha_2 A_2 + \cdots + \alpha_i A_i + \cdots + \alpha_\ell A_\ell = 0$$

Exemple

$$\left\{\begin{array}{ll} \min\limits_{\mathbf{z}\in\mathbb{R}^2} & -3z_1 - 2z_2 \\ \text{avec} & 4z_1 + 2z_2 - z_3 = 16 \\ & z_1 + 2z_2 - z_4 = 8 \\ & z_1 + z_2 - z_5 = 5 \end{array}\right.$$

si $I_b = \{1, 2, 5\}$, la solution du problème est la solution du système linéaire :

$$\begin{pmatrix} 4 & 2 & 0 \\ 1 & 2 & 0 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_5 \end{pmatrix} = \begin{pmatrix} 16 \\ 8 \\ 5 \end{pmatrix}$$

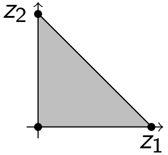
avec $z_3 = z_4 = 0$, soit :

$$\mathbf{z} = (2,67 \quad 2,67 \quad 0 \quad 0 \quad 1/3)$$

La question est donc de trouver les variables actives (l'ensemble I_b)

Domaine réalisable et solution de base

$$\begin{cases} \min_{\mathbf{z}\in\mathbb{R}^m} & \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} = \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0 \end{cases} \qquad A \in \mathcal{M}(p, m)$$



Theorem (Convexité et points extrèmes)

Si A est une matrice de rang p, Alors, à tout élément du domaine réalisable (vérifiant $A\mathbf{z} = \mathbf{b}$) peut s'écrire comme une combinaison convexe des solutions de base réalisables :

$$\mathbf{z} = \sum_{i=1}^{k} \lambda_i \mathbf{z}_{bi} \qquad \textit{avec} \ \sum_{i=1}^{k} \lambda_i = 1$$

Démonstration :

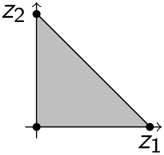
$$A\mathbf{z} = \mathbf{b} \Leftrightarrow \mathbf{a}_1 z_1 + \mathbf{a}_2 z_2 + \cdots + \mathbf{a}_p z_p + \mathbf{a}_{p+1} z_{p+1} + \cdots + \mathbf{a}_m z_m = \mathbf{b}$$

On réordonne les colonnes de A de sorte que $A_I = (\mathbf{a}_1,\ \mathbf{a}_2,\ \ldots,\ \mathbf{a}_p)$. Si les vecteurs $(\mathbf{a}_1,\ \mathbf{a}_2,\ \ldots,\ \mathbf{a}_p)$ sont linéairement indépendants,

$$\exists \alpha_{p+1,i}, i = 1, p \qquad \text{tels que} \qquad \mathbf{a}_{p+1} = \alpha_{p+1,1}\mathbf{a}_1 + \cdots + \alpha_i \mathbf{a}_{p+1,i} + \cdots + \alpha_{p+1,p}\mathbf{a}_p$$

Domaine réalisable et solution de base

$$\begin{cases} \min\limits_{\mathbf{z}\in\mathbb{R}^m} & \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} = \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0 \end{cases} \qquad A \in \mathcal{M}(p, m)$$



Theorem (Convexité et points extrèmes)

Si A est une matrice de rang p, Alors, à tout élément du domaine réalisable (vérifiant $A\mathbf{z} = \mathbf{b}$) peut s'écrire comme une combinaison convexe des solutions de base réalisables :

$$\mathbf{z} = \sum_{i=1}^{k} \lambda_i \mathbf{z}_{bi} \qquad \textit{avec} \ \sum_{i=1}^{k} \lambda_i = 1$$

<u>Démonstration</u> :

$$A\mathbf{z} = \mathbf{b} \Leftrightarrow \mathbf{a}_1 z_1 + \mathbf{a}_2 z_2 + \cdots + \mathbf{a}_p z_p + \mathbf{a}_{p+1} z_{p+1} + \cdots + \mathbf{a}_m z_m = \mathbf{b}$$

On réordonne les colonnes de A de sorte que $A_I = (\mathbf{a}_1,\ \mathbf{a}_2,\ \ldots,\ \mathbf{a}_p)$. Si les vecteurs $(\mathbf{a}_1,\ \mathbf{a}_2,\ \ldots,\ \mathbf{a}_p)$ sont linéairement indépendants,

$$\exists \alpha_{p+1,i}, i = 1, p \qquad \text{tels que} \qquad \mathbf{a}_{p+1} = \alpha_{p+1,1}\mathbf{a}_1 + \cdots + \alpha_i \mathbf{a}_{p+1,i} + \cdots + \alpha_{p+1,p}\mathbf{a}_p$$

Exemple

à la solution réalisable

$$z_1 = 0.6,\ z_2 = 0.3,\ z_3 = 0.1$$

on peut associer comme on veut

$$z_{b1} = 1 \text{ où } z_{b2} = 1 \text{ où encore } z_{b3} = 1$$

Et comme l'ensemble des solution réalisable est convexe tout point $\mathbf{z}$ réalisable peut s'écrire comme une combinaison convexe des solutions de base (qui sont les points extrêmes du polyèdre convexe)

Plan

Le principe du simplexe

- on va lister les éléments de $\mathcal{S}_I$ par ordre de cout décroissant
  1. trouver une solution initiale
  2. trouver comment passer d'une solution à une autre
     - faire sortir un point de $\mathcal{S}_I$
     - et faire entrer un nouveau point

- Le simplexe est l'enveloppe convexe d'un ensemble de points

  Un simplexe tire son nom du fait qu'il soit l'objet géométrique clos le plus "simple" qui a n dimensions, par exemple sur une droite (1 dimension) l'objet le plus simple à 1 dimension est le segment, alors que dans le plan (2 dimensions) l'objet le plus simple à 2 dimensions est le triangle, et dans l'espace (3 dimensions) l'objet le plus simple à 3 dimensions est le tétraèdre (pyramide à base triangulaire) (*cf* wikipedia).

- Pour s'y retrouver : on utilise le tableau du simplexe

Le tableau du Simplexe

Definition (le tableau du simplexe)

Pour un programme linéaire sous forme standard, le tableau du simplexe s'écrit :

$$\left\{\begin{array}{ll} \min\limits_{\mathbf{z}\in\mathbb{R}^m} & f = \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} = \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0 \end{array}\right. \qquad A \in \mathcal{M}(p, m)$$

$$\begin{array}{|cccccc|c|c|}
\hline
a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1m} & 0 & b_1 \\
a_{21} & a_{22} & \dots & a_{2j} & \dots & a_{2m} & 0 & b_2 \\
\vdots & \dots & \dots & \dots & \dots & \vdots & \vdots & \vdots \\
a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{im} & 0 & b_i \\
\vdots & \dots & \dots & \dots & \dots & \vdots & \vdots & \vdots \\
a_{p1} & a_{p2} & \dots & a_{pj} & \dots & a_{pm} & 0 & b_p \\
\hline
c_1 & c_2 & \dots & c_j & \dots & c_m & -1 & 0 \\
\hline
\end{array}$$

Justification par réécriture du problème :

$$\left\{\begin{array}{ll} \min\limits_{\mathbf{z}\in\mathbb{R}^m} & f = \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} = \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0 \end{array}\right. \rightarrow \left\{\begin{array}{lll} \min\limits_{\mathbf{z}\in\mathbb{R}^m} & f & \\ \text{avec} & A\mathbf{z} & = \mathbf{b} \\ \text{et} & \mathbf{c}^\top \mathbf{z} - f & = 0 \\ & \mathbf{z} \geq 0 & \end{array}\right. \rightarrow \begin{array}{|c|c|c|} \hline A & 0 & \mathbf{b} \\ \hline c^\top & -1 & 0 \\ \hline \end{array}$$

Le tableau du Simplexe : forme canonique

relative à la base I : $z =$ réordonancement de $[\mathbf{z}_N; \mathbf{z}_I]$ avec $\mathbf{z}_N = 0$

$$\left\{\begin{array}{ll} \min\limits_{\mathbf{z}\in\mathbb{R}^m} & f = \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} = \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0 \end{array}\right. \quad = \quad \left\{\begin{array}{ll} \min\limits_{\mathbf{z}\in\mathbb{R}^m} & f = \mathbf{c}_N^\top \mathbf{z}_N + \mathbf{c}_I^\top \mathbf{z}_I \\ \text{avec} & A_N z_N + A_I \mathbf{z}_I = \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0 \end{array}\right.$$

A_N	A_I	0	$\mathbf{b}$
$c_N^\top$	$c_I^\top$	-1	0

A_I^{-1}	0
$-c_I^\top A_I^{-1}$	1

B	I	0	$\mathbf{z}_I$
$d^\top$	0	-1	$-f$

Definition (Forme canonique du tableau du simplexe)

La forme canonique relative à la base I du tableau du simplexe est un tableau équivalent dans lequel on trouve la matrice identité et un vecteur de 0.

Avantages de la forme canonique

B	I	0	$\mathbf{z}_I$
$d^\top$	0	-1	$-f$

Le tableau résume tout de la solution associée à la base I :

1. la valeur des variables z_I (pour les variables hors base $\mathbf{z}_N = 0$).

$$\begin{aligned} A\mathbf{z} &= b \\ A_N\mathbf{z}_N + A_I\mathbf{z}_I &= b \\ A_I^{-1}A_N\mathbf{z}_N + A_I^{-1}A_I\mathbf{z}_I &= A_I^{-1}b \\ \mathbf{z}_I &= A_I^{-1}b \end{aligned}$$

2. la valeur du cout f

$$\begin{aligned} \mathbf{c}^\top\mathbf{z} &= f \\ \mathbf{c}_N^\top\mathbf{z}_N + \mathbf{c}_I^\top\mathbf{z}_I - f &= 0 \\ (\mathbf{c}_N^\top - \mathbf{c}_I^\top A_I^{-1}A_N)\mathbf{z}_N + (\mathbf{c}_I^\top - \mathbf{c}_I^\top A_I^{-1}A_I)\mathbf{z}_I - f &= -c_I^\top A_I^{-1}b \\ -f &= -c_I^\top A_I^{-1}b \end{aligned}$$

3. le cout éventuel de la réintroduction d'une variable hors base d

Exemple de tableau sous forme cannonique

dans le cas de contraintes d'inégalités, le tableau du simplexe s'écrit à l'aide de variables d'écart :

$$\begin{cases} \min\limits_{\mathbf{x}\in\mathbb{R}^m} & \mathbf{d}^\top \mathbf{x} \\ \text{avec} & B\mathbf{x} \leq \mathbf{b} \\ \text{et} & \mathbf{x} \geq 0 \end{cases} \qquad \begin{cases} \min\limits_{\mathbf{x}\in\mathbb{R}^m} & \mathbf{d}^\top \mathbf{x} \\ \text{avec} & B\mathbf{x} + \mathbf{e} = \mathbf{b} \\ \text{et} & \mathbf{x}, \mathbf{e} \geq 0 \end{cases}$$

B	I	0	b
$d^\top$	0	-1	0

$$\begin{array}{|cccccc|ccccc|c|c|}
\hline
b_{11} & b_{12} & \ldots & b_{1j} & \ldots & b_{1m} & 1 & 0 & 0 & 0 & 0 & 0 & b_1 \\
\vdots & \ldots & \ldots & \ldots & \ldots & \vdots & & \ddots & & & & \vdots & \vdots \\
b_{i1} & b_{i2} & \ldots & b_{ij} & \ldots & b_{im} & 0 & 0 & 1 & 0 & 0 & 0 & b_i \\
\vdots & \ldots & \ldots & \ldots & \ldots & \vdots & & & & \ddots & & \vdots & \vdots \\
b_{p1} & b_{p2} & \ldots & b_{pj} & \ldots & b_{pm} & 0 & 0 & 0 & 0 & 1 & 0 & b_p \\
\hline
d_1 & d_2 & \ldots & d_i & \ldots & d_m & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
\hline
\end{array}$$

$A = [B, I], c = [d; 0]$; Il y a maintenant $m + p$ variables (toujours p contraintes) et les variable de base (non nulles) sont $\{z_{m+1}, \ldots, z_{m+p}\}$

Exemple

$$\left\{\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^2} & -3x_1 - 2x_2 \\ \text{avec} & 4x_1 + 2x_2 \leq 16 \\ & x_1 + 2x_2 \leq 8 \\ & x_1 - x_2 \geq -5 \\ \text{et} & x_1, x_2 \geq 0 \end{array}\right. \qquad \left\{\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^2, \mathbf{e}\in\mathbb{R}^3} & -3x_1 - 2x_2 \\ \text{avec} & 4x_1 + 2x_2 + e_1 = 16 \\ & x_1 + 2x_2 + e_2 = 8 \\ & -x_1 + x_2 + e_3 = 5 \\ \text{et} & x_1, x_2, e_1, e_2, e_3 \geq 0 \end{array}\right.$$

on ajoute des variables d'écart. Attention au sens des inégalités

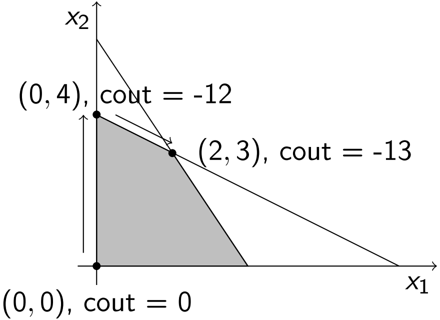
On retrouve la formulation standard avec :
$\mathbf{z} = (x_1, x_2, e_1, e_2, e_3) \in \mathbb{R}^5$
$\mathbf{c} = (-3,\ -2,\ 0,\ 0,\ 0)$

$$A = \begin{pmatrix} 4 & 2 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

L'algorithme du simplexe graphiquement

Le principe consiste à modifier l'ensemble des variables de base en faisant entrer et sortir une variable, en s'assurant, à chaque modification que le critère diminue.

$$\left\{\begin{array}{ll} \min\limits_{x_1, x_2 \in \mathbb{R}^2} & -2x_1 - 3x_2 \\ \text{avec} & x_1 + 2x_2 \leq 8 \\ & 3x_1 + 2x_2 \leq 12 \\ \text{et} & x_1, x_2 \geq 0 \end{array}\right.$$



L'algorithme du simplexe

Le principe consiste à modifier l'ensemble des variables de base en faisant entrer et sortir une variable, en s'assurant, à chaque modification que le critère diminue.

$$
\left\{\begin{array}{ll}
\max\limits_{x_1,x_2\in\mathbb{R}^2} & 2x_1+3x_2\\
\text{avec} & x_1+2x_2\leq 8\\
 & 3x_1+2x_2\leq 12\\
\text{et} & x_1,x_2\geq 0
\end{array}\right.
\rightarrow
\left\{\begin{array}{ll}
\min\limits_{x_1,x_2\in\mathbb{R}^2} & -2x_1-3x_2+0e_1+0e_1\\
\text{avec} & x_1+2x_2+e_1=8\\
 & 3x_1+2x_2+e_2=12\\
\text{et} & x_1,x_2,e_1,e_2\geq 0
\end{array}\right.
$$

z_1	z_2	z_3	z_4	cout	b	base
1	2	1	0	0	8	z_3
3	2	0	1	0	12	z_4
-2	-3	0	0	1	0	

Initialisation

z_1	z_2	z_3	z_4	cout	b	base
1	2	1	0	0	8	z_3
3	2	0	1	0	12	z_4
-2	-3	0	0	1	0	

$I_0 = \{3, 4\}$
$z_3 = 8, z_4 = 12$ cout $= 0$

dans le cas général l'initialisation peut s'avérer complexe

in : Si l'on fait entrer la variable z_2 dans la base, le cout va diminuer de $3z_2$, alors que les variables de base n'influencent pas le cout. On pourrait tout aussi bien faire entrer la variable z_1. Toute variable associé à un cout négatif ferait diminuer le cout et est donc éligible. Le choix se prote arbitrairement sur la variable associée au cout le plus négatif.

out : Avec l'introduction de z_2 dans la base, les deux équations deviennent

$$\begin{aligned} 2z_2 + z_3 &= 8 \\ 2z_2 + z_4 &= 12 \end{aligned}$$

Si c'est z_3 qui sort, $z_3 = 0$ et donc $z_2 = 4$ et $z_4 = 4$. Si c'est z_4 qui sort, $z_4 = 0$ et donc $z_2 = 6$ et $z_3 = -4$ ce qui est impossible car les variables doivent être positives.

new : Reste à adapter le tableau du simplexe aux nouvelles variables de base. Ce qui revient à remplacer la deuxième colonne du tableau du simplexe par le vecteur $\mathbf{e}_1$. Cela peut se faire en multipliant le tableau par une matrice de Gauss[1] convenable.

$$\begin{pmatrix} \frac{1}{2} & 0 & 0 \\ \frac{-2}{2} & 1 & 0 \\ \frac{3}{2} & 0 & 1 \end{pmatrix}$$

z_1	z_2	z_3	z_4	cout	b	base
1	2	1	0	0	8	z_3
3	2	0	1	0	12	z_4
-2	-3	0	0	1	0	

$=$

z_1	z_2	z_3	z_4	cout	b	base
$\frac{1}{2}$	1	$\frac{1}{2}$	0	0	4	z_2
2	0	-1	1	0	4	z_4
$-\frac{1}{2}$	0	$\frac{3}{2}$	0	1	12	

[1] pour la construction des matrices de Gauss voir le cours de calcul numérique.

Lors de l'itération suivante, c'est la première variable qui rentre, et la quatrième qui sort, et l'on obtient le tableau suivant :

z_1	z_2	z_3	z_4	cout	b	base
0	1	$\frac{3}{4}$	$-\frac{1}{4}$	0	3	z_2
1	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	2	z_1
0	0	$\frac{5}{4}$	$\frac{1}{4}$	1	13	

et l'otimum est $(z_1, z_2) = (2, 3)$ pour un cout de -13.

Plan

Forme générale de l'algorithme

in : on fait entrer dans la base une variable associée à un cout négatif. Par exemple celle qui a le cout le plus négatif. On la note z_j.

out : la variable sortante est la première à s'annuler : c'est celle pour laquelle le b_i/a_{ij} est le plus petit avec $a_{ij} > 0$. On la note z_i.

new : on multiplie le tableau du simplexe par la matrice de Gauss qui permet de transformer la colonne de la variable entrante en 0, sauf à la place de la variable sortante où l'on met un 1.

$$M_G = I - \tau_j \mathbf{e}_i^\top = \begin{pmatrix} 1 & 0 & \dots & 0 & -\frac{A(1,i)}{A(j,i)} & 0 & \dots & 0 \\ \vdots & 0 & \dots & 1 & \vdots & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & \frac{1}{A(j,i)} & 0 & \dots & 0 \\ \vdots & 0 & \dots & 0 & \vdots & 1 & \dots & \vdots \\ 0 & 0 & \dots & 0 & -\frac{A(p,i)}{A(j,i)} & 0 & \dots & 1 \end{pmatrix}$$

\+ initialisation et test d'arrêt.

Forme générale de l'algorithme

$$\begin{cases} \min\limits_{\mathbf{x}\in\mathbb{R}^m} & \mathbf{f} = \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} = \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0; \end{cases}$$

On décompose le système en distinguant les variables de base

$$\begin{cases} A = [N; B] \\ \mathbf{c} = [\mathbf{c}_n; \mathbf{c}_b] \\ \mathbf{z} = [\mathbf{z}_n; \mathbf{z}_b] \end{cases} \qquad \begin{cases} N\mathbf{z}_n & + B\mathbf{z}_b & = \mathbf{b} \\ f - \mathbf{c}_n^\top \mathbf{z}_n & - \mathbf{c}_b^\top \mathbf{z}_b & = 0 \end{cases}$$

$$B^{-1}N\mathbf{z}_n + \mathbf{z}_b = B^{-1}\mathbf{b} \qquad \Rightarrow \qquad \mathbf{z}_b = B^{-1}\mathbf{b} - B^{-1}N\mathbf{z}_n$$

La solution de base associée est la solution particulière obtenue en posant $\mathbf{z}_n = 0$. Dans ce cas $\mathbf{z}_b = B^{-1}\mathbf{b}$

réalisable et dégénérescence

- une solution de base est dite réalisable si toutes les composantes du vecteur $\mathbf{z}_b$ sont positives : $\mathbf{z}_b = B^{-1}\mathbf{b} \geq 0$.
- si une (où plusieurs) composante du vecteur $B^{-1}\mathbf{b}$ est nulle, la solution de base associée est dite dégénérée

et le cout ?

$$\left\{\begin{array}{rll} N\mathbf{z}_n & + B\mathbf{z}_b & = \mathbf{b} \\ f - \mathbf{c}_n^\top \mathbf{z}_n & - \mathbf{c}_b^\top \mathbf{z}_b & = 0 \end{array}\right. \qquad \Rightarrow \qquad \mathbf{z}_b \;=\; B^{-1}\mathbf{b} - B^{-1}N\mathbf{z}_n$$

Le cout se réécrit

$$f - \left(\mathbf{c}_n^\top - \mathbf{c}_b^\top B^{-1}N\right)\mathbf{z}_n \;=\; \mathbf{c}_b^\top B^{-1}\mathbf{b}$$

et le tableau du simplexe devient

$B^{-1}N$	I	0	$B^{-1}\mathbf{b}$
$\mathbf{c}_n^\top - \mathbf{c}_b^\top B^{-1}N$	0	1	$\mathbf{c}_b^\top B^{-1}\mathbf{b}$

Qui est B ?

Voici le tableau initial

$$
\begin{array}{|cccccc|ccccc|c|c|}
\hline
a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1m} & 1 & 0 & 0 & 0 & 0 & 0 & b_1 \\
\vdots & \dots & \dots & \dots & \dots & \vdots & & \ddots & & & & \vdots & \vdots \\
a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{im} & 0 & 0 & 1 & 0 & 0 & 0 & b_i \\
\vdots & \dots & \dots & \dots & \dots & \vdots & & & & \ddots & & \vdots & \vdots \\
a_{p1} & a_{p2} & \dots & a_{pj} & \dots & a_{pm} & 0 & 0 & 0 & 0 & 1 & 0 & b_p \\
\hline
c_1 & c_2 & \dots & c_i & \dots & c_m & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
\hline
\end{array}
$$

La $i^{\text{ème}}$ variable sort et la $j^{\text{ème}}$ rentre. Le tableau associé est donné en permutant les colonnes :

$$
\begin{array}{|cccccc|ccccc|c|c|}
\hline
a_{11} & a_{12} & \dots & 0 & \dots & a_{1m} & 1 & 0 & a_{1j} & 0 & 0 & 0 & b_1 \\
\vdots & \dots & \dots & \dots & \dots & \vdots & & \ddots & & & & \vdots & \vdots \\
a_{i1} & a_{i2} & \dots & 1 & \dots & a_{im} & 0 & 0 & a_{ij} & 0 & 0 & 0 & b_i \\
\vdots & \dots & \dots & \dots & \dots & \vdots & & & & \ddots & & \vdots & \vdots \\
a_{p1} & a_{p2} & \dots & 0 & \dots & a_{pm} & 0 & 0 & a_{pj} & 0 & 1 & 0 & b_p \\
\hline
c_1 & c_2 & \dots & 0 & \dots & c_m & 0 & 0 & c_i & 0 & 0 & 1 & 0 \\
\hline
\end{array}
$$

Qui est B ?

La $i^{\text{ème}}$ variable sort et la $j^{\text{ème}}$ rentre...

$$B = \begin{pmatrix} 1 & 0 & \dots & 0 & A(1,j) & 0 & \dots & 0 \\ \vdots & 0 & \dots & 1 & \vdots & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & A(i,j) & 0 & \dots & 0 \\ \vdots & 0 & \dots & 0 & \vdots & 1 & \dots & \vdots \\ 0 & 0 & \dots & 0 & A(p,j) & 0 & \dots & 1 \end{pmatrix}$$

L'inverse de cette matrice est donnée par la matrice de Gauss

$$B^{-1} = M_G = I - \tau_j \mathbf{e}_i^\top = \begin{pmatrix} 1 & 0 & \dots & 0 & -\frac{A(1,j)}{A(i,j)} & 0 & \dots & 0 \\ \vdots & 0 & \dots & 1 & \vdots & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & \frac{1}{A(i,j)} & 0 & \dots & 0 \\ \vdots & 0 & \dots & 0 & \vdots & 1 & \dots & \vdots \\ 0 & 0 & \dots & 0 & -\frac{A(p,j)}{A(i,j)} & 0 & \dots & 1 \end{pmatrix}$$

De plus, il est inutile de réordonner A

...et le cout ?

$$\mathbf{c}_b^\top = (0, 0, \dots, 0, \quad \underset{j}{c_j}, \quad 0, \dots, 0)$$

$$c_n^\top - \mathbf{c}_b^\top B^{-1} N = c_n^\top + (0, 0, \dots, 0, -\frac{c_i}{A(i,j)}, 0, \dots, 0)$$

$$\mathbf{c}_b^\top B^{-1} \mathbf{b} = \frac{c_i b_j}{A(i,j)}$$

Il suffit d'augmenter la matrice de Gauss d'une ligne et d'une colonne !
On obtient le cout au signe près

Théorème de terminaison

L'optimum est atteint lorsque toutes les composantes du vecteur des couts réduits $c_n^\top - \mathbf{c}_b^\top B^{-1} N$ sont positives.

Démonstration : on a toujours : $\mathbf{z}_b = B^{-1}\mathbf{b} - B^{-1}N\mathbf{z}_n$

$$\forall \mathbf{z}, \quad \mathbf{c}^\top \mathbf{z} = \mathbf{c}_n^\top \mathbf{z}_n + \mathbf{c}_b^\top \mathbf{z}_b = \underbrace{\mathbf{c}_b^\top B^{-1} \mathbf{b}}_{\text{cout minimum}} + \underbrace{(c_n^\top - \mathbf{c}_b^\top B^{-1} N)\mathbf{z}_n}_{\geq 0}$$

...et le cout ?

$$\mathbf{c}_b^\top = (0, 0, \ldots, 0, \underset{j}{c_j}, 0, \ldots, 0)$$

$$c_n^\top - \mathbf{c}_b^\top B^{-1} N = c_n^\top + (0, 0, \ldots, 0, -\frac{c_i}{A(i,j)}, 0, \ldots, 0)$$

$$\mathbf{c}_b^\top B^{-1} \mathbf{b} = \frac{c_i b_j}{A(i,j)}$$

Il suffit d'augmenter la matrice de Gauss d'une ligne et d'une colonne !
On obtient le cout au signe près

Théorème de terminaison

L'optimum est atteint lorsque toutes les composantes du vecteur des couts réduits $c_n^\top - \mathbf{c}_b^\top B^{-1} N$ sont positives.

Démonstration : on a toujours : $\mathbf{z}_b = B^{-1}\mathbf{b} - B^{-1}N\mathbf{z}_n$

$$\forall \mathbf{z}, \quad \mathbf{c}^\top \mathbf{z} = \mathbf{c}_n^\top \mathbf{z}_n + \mathbf{c}_b^\top \mathbf{z}_b = \underbrace{\mathbf{c}_b^\top B^{-1}\mathbf{b}}_{\text{cout minimum}} + \underbrace{(c_n^\top - \mathbf{c}_b^\top B^{-1} N)\mathbf{z}_n}_{\geq 0}$$

Forme générale de l'algorithme

in : on fait entrer dans la base une variable associée à un cout négatif. Par exemple celle qui a le cout le plus négatif. On la note z_j.

out : la variable sortante est la première à s'annuler : c'est celle pour laquelle le b_i/a_{ij} est le plus petit avec $a_{ij} > 0$. On la note z_i.

new : on multiplie le tableau du simplexe par la matrice de Gauss qui permet de transformer la colonne de la variable entrante en 0, sauf à la place de la variable sortante où l'on met un 1.

$$M_G = I - \tau_j \mathbf{e}_i^\top = \begin{pmatrix} 1 & 0 & \dots & 0 & -\frac{A(1,i)}{A(j,i)} & 0 & \dots & 0 \\ \vdots & 0 & \dots & 1 & \vdots & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & \frac{1}{A(j,i)} & 0 & \dots & 0 \\ \vdots & 0 & \dots & 0 & \vdots & 1 & \dots & \vdots \\ 0 & 0 & \dots & 0 & -\frac{A(p+1,i)}{A(j,i)} & 0 & \dots & 1 \end{pmatrix}$$

+ initialisation et test d'arrêt.

Le simplex en Matlab

```
A = [ -3 2 1 0 0 0;
      -1 2 0 1 0 0;
       1 1 0 0 1 0];
b = [2;4;5];
c = [-1 -2 0 0 0 1];

b = [b;0];  A = [A;c];

in = 2;
 b./A(:,in)
 out = 1;
 Mg = eye(4);
 Mg(:,out) = -A(:,in)/A(out,in);
 Mg(out,out) = 1/A(out,in); Mg
 A = Mg*A
 b = Mg*b

in = 1;
 b./A(:,in)
 out = 2;
 Mg = eye(4);
 Mg(:,out) = -A(:,in)/A(out,in);
 Mg(out,out) = 1/A(out,in); Mg
 A = Mg*A
 b = Mg*b

in = 3;
 b./A(:,in)
 out = 3;
  Mg = eye(4);
 Mg(:,out) = -A(:,in)/A(out,in);
 Mg(out,out) = 1/A(out,in); Mg
 A = Mg*A
 b = Mg*b
```

```
 b./A(:,in)  =
     1
     2
     5
     0

Mg = 0.5000         0         0         0
    -1.0000         1         0         0
    -0.5000         0         1         0
     1.0000         0         0         1

A =      -1.5000    1.0000    0.5000         0         0         0
          2.0000         0   -1.0000    1.0000         0         0
          2.5000         0   -0.5000         0    1.0000         0
         -4.0000         0    1.0000         0         0    1.0000

 b./A(:,in)  =
   -0.6667
    1.0000
    1.6000
   -0.5000


Mg =      1    0.7500         0         0
          0    0.5000         0         0
          0   -1.2500         1         0
          0    2.0000         0         1


A =       0    1.0000   -0.2500    0.7500         0         0
    1.0000         0   -0.5000    0.5000         0         0
          0         0    0.7500   -1.2500    1.0000         0
          0         0   -1.0000    2.0000         0    1.0000

 b./A(:,in)  =
   -10
    -2
     2
    -6

Mg =      1         0    0.3333         0
          0         1    0.6667         0
          0         0    1.3333         0
          0         0    1.3333         1

A =       0    1.0000         0    0.3333    0.3333         0
    1.0000         0         0   -0.3333    0.6667         0
          0         0    1.0000   -1.6667    1.3333         0
          0         0         0    0.3333    1.3333    1.0000
```

Plan

Principe de l'initialisation

Lorsque l'un des coefficient b_i du second membre b est négatif, il n'est plus possible d'initialiser le simplexe par $x_i = b_i$ car les variables x_i doivent être positives. C'est le cas par exemple lorsque s'une inégalité est stipulé dans le « mauvais » sens comme par exemple $x_1 + x_2 \geq 3$,
Il nous faut donc trouver une méthode générique permettant de déterminer le sommet initial du simplexe.

1. transformation en un programme équivalent avec $b \geq 0$

$$\text{pour tous les } b_i < 0 \Rightarrow (A(i,:), b_i) \text{ devient} (-A(i,:), -b_i)$$

2. construction du programme auxiliaire en introduisant là où c'est nécessaire des variables auxiliaires w_i.

$$\begin{cases} \min\limits_{\mathbf{z} \in \mathbb{R}^m} & \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} \leq \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0; \qquad \mathbf{b} \geq 0 \end{cases} \quad \text{devient} \quad \begin{cases} \min\limits_{\mathbf{z},\mathbf{w} \in \mathbb{R}^{m+p}} & \sum\limits_{i=1}^{p} w_i \\ \text{avec} & A\mathbf{z} + \mathbf{w} = \mathbf{b} \\ \text{et} & \mathbf{z}, \mathbf{w} \geq 0 \end{cases}$$

3. résoudre le programme auxiliaire en partant de $\mathbf{x} = 0$ et $\mathbf{w} = \mathbf{b}$. Si la solution obtenue vérifie $\mathbf{w} \neq 0$ le programme initial n'admet pas de solution réalisable. Si non, la solution obtenue $\mathbf{x}$ est une solution initiale réalisable du programme initial.

Exemple d'initialisation

$$\left\{\begin{array}{ll} \max\limits_{z_1, z_2 \in \mathbb{R}^2} & z_1 + z_2 \\ \text{avec} & 3z_1 + 4z_2 \geq 7 \\ & 2z_1 + z_2 \leq 12 \\ \text{et} & z_1, z_2 \geq 0 \end{array}\right. \rightarrow \left\{\begin{array}{ll} \max\limits_{\mathbf{z}, \mathbf{x} \in \mathbb{R}^4} & z_1 + z_2 \\ \text{avec} & 3z_1 + 4z_2 - x_3 = 7 \\ & 2z_1 + z_2 + x_4 = 12 \\ \text{et} & z_1, z_2, x_3, x_4 \geq 0 \end{array}\right.$$

$$\rightarrow \left\{\begin{array}{ll} \min\limits_{\mathbf{z}, \mathbf{x}, w_1 \in \mathbb{R}^5} & w_1 \\ \text{avec} & 3z_1 + 4z_2 - x_3 + w_1 = 7 \\ & 2z_1 + z_2 + x_4 = 12 \\ \text{et} & z_1, z_2, x_3, x_4, w_1 \geq 0 \end{array}\right.$$

Les variables de bases sont (x_4, w_1). Il faut maintenant écrire le cout en fonction des variables hors base. En utilisant la première équation on a :

$$\left\{\begin{array}{ll} \min & g = w_1 \\ & 3z_1 + 4z_2 - x_3 + w_1 = 7 \end{array}\right. \Leftrightarrow \left\{\begin{array}{ll} \min & g = -3z_1 - 4z_2 + x_3 \\ & w_1 = 7 - 3z_1 - 4z_2 + x_3 \end{array}\right.$$

Initialisation et tableau du simplexe

Tableau du simplexe initial:

z_1	z_2	z_3	z_4	z_5	cout	b	base
3	4	-1	0	1	0	7	$z_5 = w_1$
2	1	0	1	0	0	12	$z_4 = x_2$
-3	-4	1	0	0	1	0	

La solution de ce problème initial est $(x_2 = 7/4, x_4 = 41/4)$ et le tableau :

z_1	z_2	z_3	z_4	z_5	cout	b	base
$3/4$	1	$-1/4$	0	$1/4$	0	$7/4$	z_2
$5/4$	0	$1/4$	1	$-1/4$	0	$41/4$	$z_4 = x_2$
0	0	0	0	1	1	0	

Il nous reste à repartir avec la fonction cout initiale :

z_1	z_2	z_3	z_4	cout	b	base
$3/4$	1	$-1/4$	0	0	$7/4$	z_2
$5/4$	0	$1/4$	1	0	$41/4$	$z_4 = x_2$
$-1/4$	0	$-1/4$	0	1	$7/4$	

Plan

Réécriture d'un programme linéaire

- définition de la forme canonique

$$\begin{cases} \min\limits_{\mathbf{z}\in\mathbb{R}^m} & \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} = \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0 \end{cases}$$

- transformation des variables non positives :

$$\begin{cases} \min\limits_{\mathbf{z}\in\mathbb{R}^m} & \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} = \mathbf{b} \end{cases} \quad \Leftrightarrow \quad \begin{cases} \min\limits_{\mathbf{z}_+,\mathbf{z}_-\in\mathbb{R}^{2m}} & \mathbf{c}^\top (\mathbf{z}_+ - \mathbf{z}_-) \\ \text{avec} & A(\mathbf{z}_+ - \mathbf{z}_-) = \mathbf{b} \\ \text{et} & \mathbf{z}_+, \mathbf{z}_- \geq 0 \end{cases}$$

- transformation des inégalités en égalités

$$\begin{cases} \min\limits_{\mathbf{z}\in\mathbb{R}^m} & \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} \leq \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0 \end{cases} \quad \Leftrightarrow \quad \begin{cases} \min\limits_{\mathbf{z}\in\mathbf{s}\mathbb{R}^{m+n}} & \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} + \mathbf{s} = \mathbf{b} \\ \text{et} & \mathbf{z}, \mathbf{s} \geq 0 \end{cases}$$

exemple : la régression L1

la régession :

$$\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{i=1}^{n} |\mathbf{w}^\top \mathbf{x}_i - y_i|$$

n contraintes et n variables d'écart

$$\begin{cases} \displaystyle\min_{\mathbf{w}\in\mathbb{R}^d, \varepsilon\in\mathbb{R}^n} & \mathbf{e}^\top |\varepsilon| \\ \text{avec} & \varepsilon = X\mathbf{w} - \mathbf{y} \end{cases}$$

soit en posant $\varepsilon = \varepsilon_+ - \varepsilon_-$

$$\begin{cases} \displaystyle\min_{\mathbf{w},\varepsilon_+,\varepsilon_-} & \mathbf{e}^\top(\varepsilon_+ + \varepsilon_-) \\ \text{avec} & \varepsilon_+ - \varepsilon_- = X\mathbf{w} - \mathbf{y}; \qquad \varepsilon_+, \varepsilon_- \geq 0 \end{cases}$$

Forme canonique de la régression L1

$$\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{i=1}^{n} |\mathbf{w}^\top \mathbf{x}_i - y_i| \quad \Rightarrow \quad \left\{ \begin{array}{lll} \displaystyle\min_{\mathbf{w},\varepsilon_+,\varepsilon_-} & \mathbf{e}^\top(\varepsilon_+ + \varepsilon_-) & \\ \text{avec} & \varepsilon_+ - \varepsilon_- = X\mathbf{w} - \mathbf{y}; & \varepsilon_+, \varepsilon_- \geq 0 \end{array} \right.$$

On obtient alors la forme canoniques des programmes linéaires en posant $\mathbf{w} = \mathbf{w}_+ - \mathbf{w}_-$ et :

$$\begin{array}{rl} \mathbf{z} = & (\mathbf{w}_+; \mathbf{w}_-; \varepsilon_+; \varepsilon_-)^\top \\ \mathbf{b} = & \mathbf{y} \\ \mathbf{c} = & (0_{2d}; \mathbb{I}_{2n})^\top \\ A = & \left(\begin{array}{c|c|c|c} X & -X & -I & I \end{array} \right) \end{array} \qquad \left\{ \begin{array}{ll} \displaystyle\min_{\mathbf{z}\in\mathbb{R}^{2d+2n}} & \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} = \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0 \end{array} \right.$$

.

Le tableau du simplexe de la régression L1

Deux problèmes restent à régler.

start si on initialise $\mathbf{w} = 0$ les variables d'écart sont $\varepsilon_{i+} = b_i, \varepsilon_{i-} = 0$ pour les $b_i \geq 0$ et $\varepsilon_{i-} = -b_i, \varepsilon_{i+} = 0$ pour les $b_i < 0$ ou l'on multiplie par -1 les équations correspondant à $b_i < 0$ et l'on pose alors $\varepsilon_{i+} = 0, \varepsilon_{i-} = b_i$.

cout le cout doit ensuite être exprimé en fonction des variables hors base. Une fois les signes changées, on a $\varepsilon = \mathbf{b} - A\mathbf{w}$ et le cout peut s'écrire $\mathbb{1}^\top \varepsilon = \mathbb{1}^\top \mathbf{b} - \mathbb{1}^\top A\mathbf{w}$,

En posant $\mathbf{s} = A^\top \mathbb{1}$ on a le tableau suivant :

$\mathbf{w}_+$	$\mathbf{w}_-$	ε_+	ε_-	cout	b	base
X	$-X$	$-I$	I	0	$\widetilde{\mathbf{b}}$	ε
$\mathbf{s}$	$\mathbf{s}$	0	0	1	0	

LP et régression L1

Donc par construction nous avons n variables non nulles (les $2d + n$ autres étant nulles...). Si l'on considère que les d variables de $\mathbf{w}$ sont non nulles, cela revient à dire que d contraintes sont saturées et vérifient :

$$\varepsilon_+(\mathcal{A}) = \varepsilon_-(\mathcal{A}) = 0$$

où $\mathcal{A}$ est l'ensemble des indices correspondant aux variables non nulles. Ces d points sont ceux qui permettent de trouver $\mathbf{w}$ en résolvant le système linéaire suivant :

$$X(\mathcal{A}; \mathcal{A})\mathbf{w} - \mathbf{y}(\mathcal{A})$$

le problème étant de trouver $\mathcal{A}$. Il peut y en avoir moins ($\mathcal{C}(\mathcal{A}) \leq d$), mais pas plus.

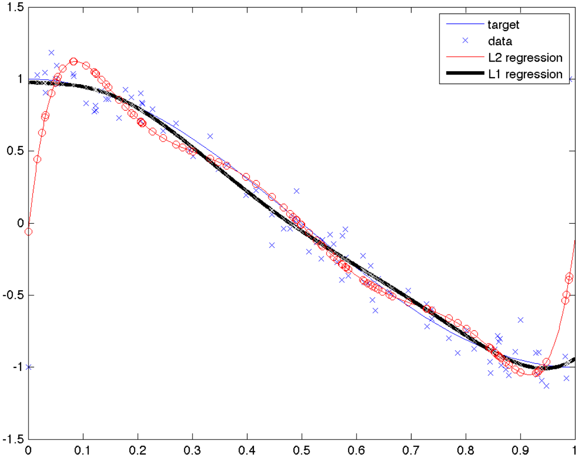
LP en matlab

$$\begin{aligned} \mathbf{z} &= (\mathbf{w}_+; \mathbf{w}_-; \varepsilon_+; \varepsilon_-)^\top \\ \mathbf{b} &= \mathbf{y} \\ \mathbf{c} &= (0_{2d}; \mathbb{1}_{2n})^\top \\ A &= \left(\; X \mid -X \mid -I \mid I \; \right) \end{aligned} \qquad \left\{ \begin{array}{ll} \min\limits_{\mathbf{z} \in \mathbb{R}^{2d+2n}} & \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} = \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0 \end{array} \right.$$

```
X = LINPROG(f,A,b) attempts to solve the linear programming problem:

         min f'*x    subject to:   A*x <= b
          x

 X = LINPROG(f,A,b,Aeq,beq) solves the problem above while additionally
 satisfying the equality constraints Aeq*x = beq.

 X = LINPROG(f,A,b,Aeq,beq,LB,UB) defines a set of lower and upper
 bounds on the design variables, X...
```

```
f = [zeros(size(a)) ;zeros(size(a)) ;ones(size(y)); ones(size(y))];
b = ya;
A = [X -X -eye(length(ya)) eye(length(ya))];
[a2 f1 e o l] = linprog(f,[],[],A,b,0*f);
a2 = [eye(size(X,2)) -eye(size(X,2)) zeros(size(X')) zeros(size(X')) ]*a2;
plot(xt,Xt*a2,'k');
```

Régression L1



Plan

Dual d'un programme linéaire

$$\begin{cases} \min\limits_{\mathbf{z}\in\mathbb{R}^m} & \mathbf{c}^\top \mathbf{z} \\ \text{avec} & A\mathbf{z} = \mathbf{b} \\ \text{et} & \mathbf{z} \geq 0 \end{cases} \qquad A \in \mathcal{M}(p, m)$$

$$\begin{cases} \max\limits_{\mathbf{y}\in\mathbb{R}^p} & \mathbf{b}^\top \mathbf{y} \\ \text{avec} & A^\top \mathbf{y} \leq \mathbf{c} \end{cases} \qquad A \in \mathcal{M}(p, m)$$

Conclusion

- L'importance de la forme standart
- Il existe maintenant d'autres algorithmes plus puissants : points intérieurs
- le Lièvre L2 et la tortue L1