# Calcul de dépendances dans les boucles
# Programmation linéaire

F. Thomasset
INRIA - Rocquencourt
Projet A3

http://www-rocq.inria.fr/~thomasse/dea/

http://www-rocq.inria.fr/~thomasse/dea/articles.html

4 janvier 2002

# Programmation linéaire

sert à maximiser (ou minimiser) une fonction de coût linéaire sous des contraintes linéaires

$$\begin{array}{ll} \mathrm{Max} & c \cdot x \\ & Ax \leq b \\ & x \in \mathbf{R}^n \ , \ x \geq 0 \end{array}$$

ou (forme "standard") :

$$\begin{array}{ll} \mathrm{Max} & c \cdot x \\ & Ax = b \\ & x \in \mathbf{R}^n \ , \ x \geq 0 \end{array}$$

passage d'une forme à l'autre :

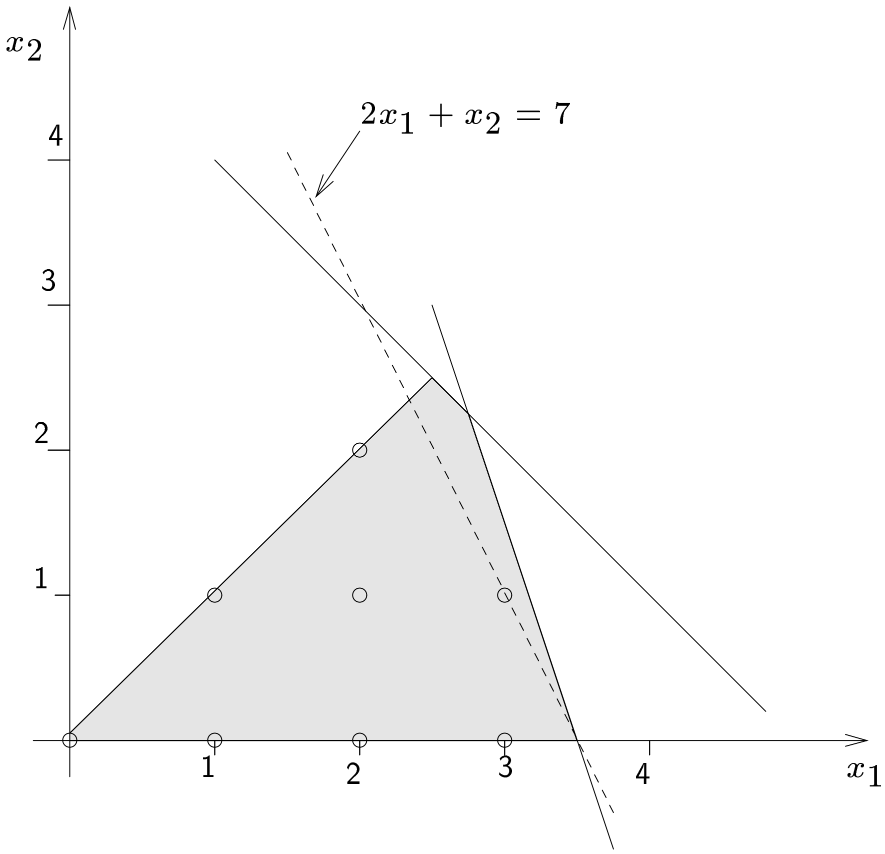
1. $A' = \begin{pmatrix} A \\ -A \end{pmatrix}$, $b' = \begin{pmatrix} b \\ -b \end{pmatrix}$
2. $y = b - Ax, y \geq 0$

Vocabulaire :

“Solution réalisable” :
$x$ satisfaisant les contraintes :
$Ax \leq b, x \geq 0$
ou
$Ax = b, x \geq 0$

“Solution optimale” :
$x$ réalisable maximisant (mimisant) la fonction objectif

$$Ax \leq b, x \geq 0$$




$$\begin{array}{ll} \text{Max} & 2x_1 + x_2 \\ & x_1 + x_2 \leq 5 \\ & -x_1 + x_2 \leq 0 \\ & 6x_1 + 2x_2 \leq 21 \\ & x_1 \geq 0, x_2 \geq 0, x_1 \text{ et } x_2 \text{ entiers} \end{array}$$

Optimum : obtenu en un <u>sommet</u> du polyèdre

Un sommet : intersection de plans de la forme $A_{i.}x = b_i$ (saturation d'une partie des contraintes)

Algorithme du simplexe (primal) : passe d'un sommet à un sommet voisin ("pivotage")

Complexité théorique $O(C^n_{m+n})$
($m$ nombre de contraintes,
$n$ nombre de variables)

En pratique complexité polynômiale (Schrijver 1986)

Le simplexe comme test d'existence de solution d'un système d'inéquations

$$
\begin{array}{ll}
\text{Min} & w_1 + w_2 + \ldots + w_m \\
& Ax + w = b \\
& x \in \mathbf{R}^n \ , \ w \in \mathbf{R}^m \\
& x \geq 0, \ w \geq 0
\end{array}
$$

Si à l'optimum $w \neq 0$ : pas de solution

# Principe de l'algorithme du simplexe primal

Sous forme standard :
Max $\begin{pmatrix} 0 & 0 & 0 & +2 & +1 \end{pmatrix} \cdot \vec{x}$

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & -1 & 1 \\ 0 & 0 & 1 & 6 & 2 \end{pmatrix} \cdot \begin{pmatrix} x_3 \\ x_4 \\ x_5 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 5 \\ 0 \\ 21 \end{pmatrix}$$

de la forme : $c = \begin{pmatrix} c_B & c_N \end{pmatrix}$
$\begin{pmatrix} B & N \end{pmatrix} \cdot \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} b_B \\ b_N \end{pmatrix}$
avec $B$ inversible

Si $b_B \geq 0$ :
Un sommet ("solution de base")[1] :
$x_B = b_B, x_N = 0$

Dans l'exemple : sommet de départ :
$x_3 = 5, x_4 = 0, x_5 = 21, x_1 = x_2 = 0$

1. vocabulaire de la programmation linéaire : $B$ est une "base"

# simplexe primal (suite)

Tester les cas de terminaison :

- Si $c_N - c_B B^{-1} N \leq 0$
  alors $\begin{pmatrix} x_B \\ 0 \end{pmatrix}$ solution optimale,
  avec $x_B = B^{-1} b_B$

- $\hat{c} = c_N - c_B B^{-1} N$, $\hat{N} = B^{-1} N$
  $r$ indice de colonne tel que $\hat{c}_r > 0$
  $\hat{a}_r$ colonne correspondante de $\hat{N}$
  Si $\hat{a}_r \leq 0$ : problème non borné

Dans l'exemple :
$\hat{c} = \begin{pmatrix} 2 & 1 \end{pmatrix}$, $r = 4$, $\hat{c}_r = 2$,
$\hat{N} = \begin{pmatrix} 1 & 1 \\ -1 & 1 \\ 6 & 2 \end{pmatrix}$, $B = \mathrm{Id}$, $\hat{a}_r = \begin{pmatrix} 1 \\ -1 \\ 6 \end{pmatrix}$

$\lambda = \min \left\{ \frac{\hat{b}_i}{\hat{a}_{ir}} : \hat{a}_{ir} > 0 \right\} = \frac{\hat{b}_s}{\hat{a}_{sr}}$

# simplexe primal (suite) : pivotage

Opération fondamentale du simplexe

Échanger $x_r$ et $x_s$ ($x_1$ et $x_5$)

Résoudre en $x_1$ et reporter dans les autres équations :
$x_1 = -\frac{1}{6}x_5 - \frac{1}{6}x_2 + \frac{7}{2}$

Max$\begin{pmatrix} 0 & 0 & 0 & -\frac{1}{3} & \frac{2}{3} \end{pmatrix} \cdot \vec{x}$

$$\begin{pmatrix} 0 & 0 & 1 & \frac{1}{6} & \frac{1}{6} \\ 0 & 1 & 0 & \frac{1}{6} & \frac{7}{6} \\ 1 & 0 & 0 & -\frac{1}{6} & \frac{5}{6} \end{pmatrix} \cdot \begin{pmatrix} x_3 \\ x_4 \\ x_1 \\ x_5 \\ x_2 \end{pmatrix} = \begin{pmatrix} \frac{7}{2} \\ \frac{7}{2} \\ \frac{3}{2} \end{pmatrix}$$

Nouvelle solution de base (sommet) :
$x_3 = \frac{3}{2},\ x_4 = \frac{7}{2},\ x_3 = \frac{7}{2},\ x_5 = \ x_2 = 0$

# Problèmes en nombres entiers

$$\begin{array}{ll} \text{Min} & c \cdot x \\ & Ax \leq b, \quad x \in \mathbf{Z}^n \;, \quad x \geq 0 \end{array}$$

Si on résoud par le simplexe :
solution optimale $x$
si tous les coefficients de $x$ sont entiers :
alors $x$ est solution optimale du problème dans les entiers

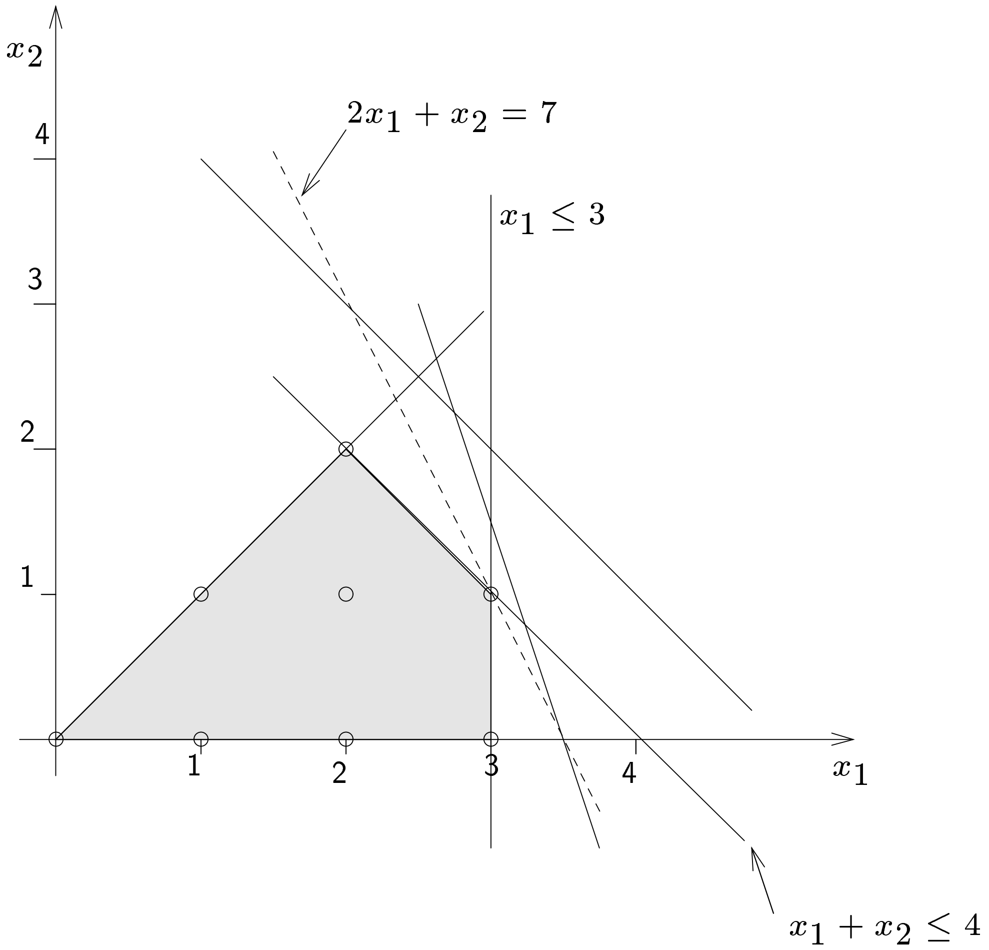
méthodes de coupes :
contraintes ajoutées au problème :
pour éliminer les solutions non entières

Dans l'exemple :
sans contrainte d'intégrité :
solution optimale $(11/4, 9/4)$

on ajoute les contraintes ("coupes") :

$$\begin{cases} x_1 + x_2 \leq 4 \\ x_1 \leq 3 \end{cases}$$

Alors : solution en variables réelles $(3, 1)$ du problème avec coupes :
optimale dans les entiers



# Application du simplexe

Peut-on appliquer le simplexe au calcul de dépendances ?

– *"Memory-based"*
  utiliser le simplexe pour tester l'existence de points entiers dans un polyèdre

– *"value-based"* (flots de données)
  Difficultés :

  – pas d'objectif numérique à maximiser
  – maximisation selon $\ll$

  Solution : simplexe dual

Problème dual d'un problème de maximisation : problème de minimisation

# Dualité

Problème dual d'un problème de maximisation : problème de minimisation :

- matrice transposée
- borne l'objectif du primal
- dual(dual) $\equiv$ le primal

| Pb. primal | Pb. dual |
|---|---|
| $z_{LP} = \text{Max } cx$<br>$Ax \leq b,$<br>$x \geq 0$ | $w_{LP} = \text{Min } ub$<br>$uA \geq c$ |

Primal irréalisable : $\nexists x \geq 0, Ax \leq b$

Dual irréalisable : $\nexists u, uA \geq c$

# Dualité : théorèmes

$$\left.\begin{array}{ll} Ax \le b & uA \ge c \\ x \ge 0 & \end{array}\right\} \Longrightarrow \; cx \le z_{LP} \le w_{LP} \le ub$$

Seulement 4 cas sont possibles :

**i/** $z_{LP} = w_{LP}$ : valeurs finies

**ii/** $z_{LP} = +\infty$, dual irréalisable

**iii/** $w_{LP} = -\infty$, primal irréalisable

**iv/** primal et dual irréalisables

# Dualité : exemple

| | | | |
|---|---|---|---|
| Max | $2x_1 + x_2$ | | |
| | $x_1 + x_2 \leq 5$ | Min | $5u_1 + 21u_3$ |
| | $-x_1 + x_2 \leq 0$ | | $u_1 - u_2 + 6u_3 \geq 2$ |
| | $6x_1 + 2x_2 \leq 21$ | | $u_1 + u_2 + 2u_3 \geq 1$ |
| | $x_1 \geq 0, x_2 \geq 0$ | | |

Simplexe dual : méthode du simplexe sur le pb dual

Quand on atteint une solution primal-réalisable : optimum

avantages du simplexe dual en optimisation :
si on a une base duale, mais pas de base primale :
démarrage du simplexe dual

# Comment trouver le problème dual d'un primal ?

Transformer le problème en la forme équivalente :
Max $\{ \ c \cdot x \ \ : Ax \leq b, x \geq 0 \ \}$
Dont on connaît le dual : Min $\{ \ u \cdot b \ \ : uA \geq c, u \geq 0 \ \}$

Exemple :
Max $\{ \ c \cdot x \ \ : Ax = b, x \geq 0 \ \}$

Équivalent à :
Max $\{ \ c \cdot x \ \ : \begin{pmatrix} A \\ -A \end{pmatrix} x \leq \begin{pmatrix} b \\ -b \end{pmatrix}, x \geq 0 \ \}$

D'où le problème dual :

$$\text{Min } \{ \ u_1 \cdot b - u_2 \cdot b \ \ : \ \begin{pmatrix} u_1 & u_2 \end{pmatrix} \begin{pmatrix} A \\ -A \end{pmatrix} \geq c, \quad u_1 \geq 0, \ u_2 \geq 0 \ \}$$

$$= \text{Min } \{ \ (u_1 - u_2)b \ \ : (u_1 - u_2)A \geq C, u_1, u_2 \geq 0 \ \}$$
$$= \text{Min } \{ \ u \cdot b \ \ : uA \geq c \ \}$$

Exercice : Montrer que le dual de :
Max $\{ \ c \cdot x \ \ : Ax \leq b \ \}$
est : Min $\{ \ u \cdot b \ \ : uA = c, u \geq 0 \ \}$

# Optimisation lexicographique

L'opération de base dans l'algorithme du simplexe :
pivotage :
combinaison linéaire
des lignes de la matrice courante
(passage d'un sommet à un sommet voisin)

Idées de base de PIP
("Parametric Integer Programming", Feautrier 1988) :

- On peut exécuter cette opération "symboliquement" sur le dual
(même si le second membre n'est pas un nombre
cf "paramètres de structure")

- Quel critère de minimisation ?
Maximiser selon l'ordre lexicographique $\ll$

Comment fonctionne la recherche
de l'optimum lexicographique ?

Le principe :

Reformulation du problème :

$$\mathbf{P} = \{ \ \vec{x} \ \mid \ \vec{x} \geq 0, \ M \cdot \vec{x} + \vec{v} \geq 0 \ \}$$

avec $M, \ \vec{v}$ donnés (nombres)

– Est-ce que $\mathbf{P} = \emptyset$ ?

– Si $\mathbf{P} \neq \emptyset$,
quel est le minimum lexicographique de $\mathbf{P}$ :
$\vec{x} = \underset{\ll}{\mathrm{Min}} \ \mathbf{P}$ : $\left\{ \begin{array}{l} \vec{x} \ \in \ \mathbf{P} \\ \forall \vec{y} \ \in \mathbf{P}, \ \vec{y} \ll \vec{x} \end{array} \right.$

Résolution par une succession de changements
de variables de la forme :

$$\vec{x} = M' \cdot \vec{y} + \vec{u}$$

Nouvelles contraintes :

$$\textbf{P'} = \{\ M' \cdot \vec{y} + \vec{u} \mid \quad \begin{array}{l} M' \cdot \vec{y} + \vec{u} \geq 0, \\ MM'\vec{y} + M \cdot \vec{u} + \vec{v} \geq 0 \end{array} \}$$

D'où changement de notation :

$$\textbf{P} = \{\ A \cdot \vec{y} + \vec{b} \mid \quad \begin{array}{l} \vec{x} = A \cdot \vec{y} + \vec{b} \geq 0 \\ \vec{z} = C \cdot \vec{y} + \vec{d} \geq 0 \\ \vec{y} \geq 0 \end{array} \}$$

Initialement :
$C = M,\ A =$ matrice unité , $\vec{b} = \vec{0},\ \vec{d} = \vec{v}$

$$S = \begin{pmatrix} A \\ C \end{pmatrix} \quad \vec{t} = \begin{pmatrix} \vec{b} \\ \vec{d} \end{pmatrix}$$

$$\textbf{P} = \{\ A \cdot \vec{y} + \vec{b}\ \mid\ S \cdot \vec{y} + \vec{t}\ \geq \vec{0},\ \vec{y} \geq \vec{0}\ \}$$

Pour chaque colonne $j$ : $\qquad S_{.j} \gg 0$
(vrai initialement, conservé par les transformations)

Les cas faciles :

- Si $\exists i$ : la ligne $S_{i.} \leq 0$ et $t_i < 0$
  $\Longrightarrow$ polyèdre vide

- Si $\forall i$ : $t_i \geq 0$
  solution optimale $\vec{y}_0 = \vec{0}$ $(\vec{x}_0 = \vec{b}, \vec{z}_0 = \vec{d})$

Le cas non trivial :

- Si $\exists i, j$ : $t_i < 0,\ S_{i,j} > 0$ ?

$\exists i, j$ : $t_i < 0,\ S_{i,j} > 0$ :
la ligne $i$ est de la forme : $\qquad w_i = S_{i,.} \cdot \vec{y} + t_i$

Éliminer $y_j$ : ( *"Pivotage"*)

$$y_j = w_i/S_{i,j} - \sum_{k \neq j}(S_{i,k}/S_{i,j})y_k - t_i/S_{i,j}$$

Nouveau système après *"pivotage"* :

$$S'_{.j} = (1/S_{i,j})S_{.j}$$
$$S'_{.k} = S_{.k} - (S_{i,k}/S_{i,j})S_{.j} \quad (k \neq j)$$
$$t' = t - (t_i/S_{i,j})S_{.j}$$

Choix de la colonne de pivot :

Choisir $j$ pour que chaque colonne $S'_{.k}$
soit encore $\gg 0$ :

$j$ tel que $(1/S_{i,j})S_{.j}$ minimal pour $\ll$

Convergence :
on enlève à $t$ un vecteur $\gg 0$ :
on ne peut pas retrouver 2 fois le même $t$ :
bouclage impossible

# PIP

`http://www.prism.uvsq.fr/public/paf/parmprog.ps`

Si le polyèdre **P** dépend de paramètres :

$$\mathbf{P}(\vec{z}) = \{\ A\cdot\vec{x} + A'\cdot\vec{z} + \vec{b} \mid \begin{array}{l} A\cdot\vec{x} + A'\cdot\vec{z} + \vec{b} \geq 0 \\ C\cdot\vec{x} + C'\cdot\vec{z} + \vec{d} \geq 0 \\ \vec{x} \geq 0\ \} \end{array}$$

Questions :
- Pour quels $\vec{z}$ a-t-on $\mathbf{P}(\vec{z}) = \emptyset$ ?
- Si $\mathbf{P}(\vec{z}) \neq \emptyset$ : minimum lexicographique de $\mathbf{P}(\vec{z})$ ?

$t_i(\vec{z})$ formes linéaires en $z$ :
peut-on appliquer l'algorithme du simplexe à $(S,\ t(\vec{z}))$ ?

1. $S$ matrice numérique :
   le pivotage donne un nouveau système $(S',\ t'(\vec{z}))$
   avec colonnes $\gg 0$

2. comment choisir le pivot $(i,j)$ ?
   le choix de la ligne $i$ : dépend du signe de $t_i(\vec{z})$

(Choix du pivot, suite)

Que sait-on sur $\vec{z}$ ?

Hypothèse :
les paramètres $\vec{z}$ sont soumis à des contraintes linéaires : ("contexte du problème")

$$K \cdot \vec{z} + \vec{h} \geq 0$$

$t_i(\vec{z}) \geq 0$ compatible avec le contexte ?
c'est un PL !

Après pivotage :
$t'(\vec{z})$ : encore des formes linéaires

Algorithme `PIP` :

1/ `Calculer les signes des` $t_i(\vec{z})$
`dans le contexte` $K \cdot \vec{z} + \vec{h} \geq 0$

2/ `Si` $\forall i, \ t_i(\vec{z}) \geq 0$ :
`optimum trouvé`

3/ `Si` $\exists i : \ t_i(\vec{z}) < 0$ :
- `Si` $\forall j, \ S_{i,j} \leq 0$ :
  `pas de solution`
- `Si` $\exists j : S_{i,j} > 0$ :
  `étape de pivotage`

4/ `autres cas : choisir un` $i$
`avec` $t_i(\vec{z})$ `de signe indéterminé`
`séparer le problème en deux branches :`
- $\vec{x_1}$ `solution dans le contexte`
  $K \cdot \vec{z} + \vec{h} \geq \vec{0}, t_i(\vec{z}) \geq 0$
- $\vec{x_2}$ `solution dans le contexte`
  $K \cdot \vec{z} + \vec{h} \geq \vec{0}, t_i(\vec{z}) < 0$

`la solution du problème initial :`
`if` $t_i(\vec{z}) \geq 0$ `then` $x_1$ `else` $x_2$

Application :

```
DO I = 0, M
   DO J = 0, N
      A(2*I+J) = I+J
   END DO
END DO
Y = A(K)
```

Pour quelles valeurs de $k$,
A($k$) est-il défini par la boucle ?

Quelle instance de quelle instruction définit la valeur de A($k$) ?

Maximum lexicographique des $[i, j]$ tels que ($k$ fixé) :

$0 \leq i \leq m$, $0 \leq j \leq n$, $2i + j = k$

Paramètres : $\vec{z} \ = \ [k, m, n]$

Contexte : $m > 0, n > 0$

$$\mathbf{P}([k, m, n]) = \{ \ [i, j] \ \mid \ 0 \leq i \leq m, \ 0 \leq j \leq n \\ 2i + j = k \ \}$$

$i' = m - i,\ j' = n - j$

Minimum lexicographique de **P'** :

$$\textbf{P'}([k,m,n]) = \{\ [i',j']\ \mid\ \begin{array}{l} 0 \le i' \le m,\ 0 \le j' \le n \\ -2i' - j' - k + 2m + n \ge 0 \\ 2i' + j' + k - 2m - n \ge 0\ \} \end{array}$$

Contexte : $m > 0, n > 0$

$\begin{pmatrix} S & \vec{t} \end{pmatrix}$ :

| | $i'$ | $j'$ | $1$ | $k$ | $m$ | $n$ | signe |
|---|---|---|---|---|---|---|---|
| $i'$ | $1$ | $0$ | $0$ | $0$ | $0$ | $0$ | $0$ |
| $j'$ | $0$ | $1$ | $0$ | $0$ | $0$ | $0$ | $0$ |
| $a$ | $-1$ | $0$ | $0$ | $0$ | $1$ | $0$ | $+$ |
| $b$ | $0$ | $-1$ | $0$ | $0$ | $0$ | $1$ | $+$ |
| $c$ | $-2$ | $-1$ | $0$ | $-1$ | $2$ | $1$ | ? |
| $d$ | $2$ | $1$ | $0$ | $1$ | $-2$ | $-1$ | ? |

1er cas : $-k + 2m + n < 0$ : polyèdre vide

Sinon : $-k + 2m + n \ge 0$
pivotage : ligne $d$, colonne $j'$
($d$ entre dans la base, $j'$ quitte la base)

Contexte : $m > 0, n > 0, -k + 2m + n \geq 0$

Après 1er pivotage :
$\begin{pmatrix} S & \vec{t} \end{pmatrix}$ :

| | $i'$ | $d$ | $1$ | $k$ | $m$ | $n$ | signe |
|---|---|---|---|---|---|---|---|
| $i'$ | $1$ | $0$ | $0$ | $0$ | $0$ | $0$ | $0$ |
| $d$ | $0$ | $1$ | $0$ | $0$ | $0$ | $0$ | $0$ |
| $a$ | $-1$ | $0$ | $0$ | $0$ | $1$ | $0$ | $+$ |
| $b$ | $2$ | $-1$ | $0$ | $1$ | $-2$ | $0$ | ? |
| $c$ | $0$ | $-1$ | $0$ | $0$ | $0$ | $0$ | $0$ |
| $j'$ | $-2$ | $1$ | $0$ | $-1$ | $2$ | $1$ | $+$ |

Un seul signe inconnu :

- si $k - 2m \geq 0$ :
  tous les seconds membres sont positifs
  d'où la solution :
  $$i' = 0,\ j' = -k + 2m + n$$
- Sinon encore un pivotage

Contexte :

$m > 0, n > 0,\ -k + 2m + n \geq 0,\ k - 2m < 0$

Après 2ème pivotage :

$\begin{pmatrix} S & \vec{t} \end{pmatrix}$ :

| | $b$ | $d$ | $1$ | $k$ | $m$ | $n$ | signe |
|---|---|---|---|---|---|---|---|
| $b$ | $1$ | $0$ | $0$ | $0$ | $0$ | $0$ | 0 |
| $d$ | $0$ | $1$ | $0$ | $0$ | $0$ | $0$ | $0$ |
| $a$ | $-1/2$ | $-1/2$ | $0$ | $1/2$ | $0$ | $0$ | $+$ |
| $c$ | $0$ | $-1$ | $0$ | $0$ | $0$ | $0$ | $0$ |
| $i'$ | $1/2$ | $1/2$ | $0$ | $-1/2$ | $1$ | $0$ | $+$ |
| $j'$ | $-1$ | $0$ | $0$ | $0$ | $0$ | $1$ | $+$ |

Optimum :

$b = d = 0$

$i' = -k/2 + m$

$j' = n$

Remplacer $i', j'$ par leurs expressions en $i, j$ :
solution (dans **R**) :

$$[i,j] = \text{ if } (-k+2m+n) \geq 0 \\ \quad \text{then} \\ \quad\quad \text{if } (k-2m) \geq 0 \\ \quad\quad\quad \text{then } [m, k-2m] \\ \quad\quad\quad \text{else } [k/2, 0] \\ \quad \text{else } \perp$$

$\perp$ : note l'absence de solution

Prise en compte des contraintes d'intégrité: la méthode des coupes s'applique au problème paramétrique

$x \div y$ : division entière

$$
\begin{array}{l}
[i,j] = \ \text{if } (-k+2m+n) \geq 0 \\
\qquad\quad \text{then} \\
\qquad\qquad \text{if } (k-2m) \geq 0 \\
\qquad\qquad\quad \text{then } [m, k-2m] \\
\qquad\qquad\quad \text{else if } (-k+n+2(k \div 2)) \geq 0 \\
\qquad\qquad\qquad \text{then } [k \div 2, k-2(k \div 2)] \\
\qquad\qquad\qquad \text{else } \perp \\
\qquad\quad \text{else } \perp
\end{array}
$$

# PIP – suite

Le résultat de l'analyse par PIP :
un arbre de sélection fonction les paramètres

*"quast"* : "quasi-affine selection tree"

(quasi-affine : à cause des divisions)

*"quast"* : Quasi-Affine Selection Tree

```
quast := ⊥
   | vecteur
   | ‘if (forme ≥ 0) then quast else quast’

vecteur := [ forme, ... ]

forme := entier
   | paramètre
   | entier * forme
   | forme ÷ entier
   | forme + forme
```

## Application de PIP à l'analyse de dépendances

Dans une boucle :
quelle est l'instruction qui produit
un élément de tableau utilisé dans une instruction ?

À chaque instruction `r` du texte du programme
correspondent plusieurs $instances$

Repérer les instances :

- par l'instruction `r`

- par le vecteur d'itération :
  les indices de boucle à partir de la plus externe

Bornes des boucles pour `r` : linéaires :

$$e_r(\vec{b}) \;\geq \vec{0}$$

```
    DO I = 0, M
       DO J = 0, N
r:        A(2*I+J) = I+J
s:        B(2,I) = I
       END DO
    END DO
```

les vecteurs d'itération : $[i, j]$

instances $\langle \texttt{r}\texttt{,}\ \ [i, j] \rangle$, $\langle \texttt{s}\texttt{,}\ \ [i, j] \rangle$

Comment s'exprime l'ordre standard d'exécution ?

$\lhd$ : précédence textuelle
$\texttt{r} \lhd \texttt{s}$ : vrai ssi `r` est <u>avant</u> `s` dans le texte

$\texttt{r} \lhd \texttt{r} \equiv$ Faux, $\quad \texttt{r} \lhd \texttt{s} \equiv \neg\,(\texttt{s} \lhd \texttt{r})$

<u>Profondeur</u> d'une instruction : nombre de boucles qui la contiennent

dans l'exemple :
`r` et `s` à la profondeur 2
la boucle `DO J` à la profondeur 1

$\langle \texttt{r}\texttt{,}\ \ [i, j] \rangle \prec \langle \texttt{s}\texttt{,}\ \ [i', j'] \rangle$ ?

Si $\texttt{r} = \texttt{s}$ :
$\langle \texttt{r}\texttt{,}\ \ [i, j] \rangle \prec \langle \texttt{r}\texttt{,}\ \ [i', j'] \rangle \Longleftrightarrow [i, j] \ll [i', j']$

Si `r` et `s` sont extérieures aux boucles :
vecteurs d'itération $= [\ ]$
$\langle \texttt{r}\texttt{,}\ \ [\ ] \rangle \prec \langle \texttt{s}\texttt{,}\ \ [\ ] \rangle \Longleftrightarrow r \lhd s$

Cas général :
Soit `t` la boucle la plus interne contenant `r` et `s`
$N_{rs}$ profondeur de `t`

```
     DO I = 1, M
        DO J = 1, N
           DO L = 1, M
r:            ...
           END DO
        END DO
        DO K = 1, N
s:         ...
        END DO
     END DO
```

`t` : la boucle `DO I`
$N_{rs} = 1$

`r'` : instruction du corps de la boucle `t` :
`r'` $=$ `r` ou bien `r` inclus dans `r'`
Dans l'exemple : `r'` $=$ la boucle `DO J`
idem pour `s` : `s'` $=$ la boucle `DO K`

Notation :
$\vec{a}[1..p]$ = les $p$ premières composantes de $\vec{a}$

$\langle \texttt{r}, \ \vec{a}\rangle \prec \langle \texttt{s}, \ \vec{b}\rangle$
$\quad \Longleftrightarrow \langle \texttt{r'}, \ \vec{a}[1..N_{rs}]\rangle \prec \langle \texttt{s'}, \ \vec{b}[1..N_{rs}]\rangle$

– Si $\vec{a}[1..N_{rs}] = \vec{b}[1..N_{rs}]$ :
  ($\texttt{r'}$, $\vec{a}[1..N_{rs}]$) et ($\texttt{s'}$, $\vec{b}[1..N_{rs}]$) sont dans la même itération de la boucle $\texttt{t}$ :
  alors séquencement donné par $r \lhd s$

– Si $\vec{a}[1..N_{rs}] \neq \vec{b}[1..N_{rs}]$ :
  séquencement donné par : $a[1..N_{rs}] \ll b[1..N_{rs}]$

$\langle \texttt{r}, \ [i,j]\rangle \prec \langle \texttt{s}, \ [i',j']\rangle \Longleftrightarrow$
$\quad \vec{a}[1..N_{rs}] \ll \vec{b}[1..N_{rs}]$
$\quad$ ou bien :
$\quad\quad \vec{a}[1..N_{rs}] = \vec{b}[1..N_{rs}]$
$\quad\quad$ et $r \lhd s$

Dans l'exemple :
$r \lhd s =$ Vrai
$\langle \texttt{r}[i,j,l]\rangle \prec \langle \texttt{s}[i',k']\rangle \Longleftrightarrow$
$\quad i < i'$ ou bien : $i = i'$

Décomposition de l'ordre lexicographique :
Disjonction de $p$ formules :

$$
\begin{aligned}
&[a_1, \ldots, a_p] \ll [b_1, \ldots, b_p] \\
&\quad \Longleftrightarrow \\
&\quad a_1 < b_1 \\
&\quad \vee\ (a_1 = b_1)\ \wedge\ (a_2 < b_2) \\
&\quad \vee\ \ldots \\
&\quad \vee\ (a_1 = b_1)\ \wedge\ (a_2 = b_2) \ldots \\
&\qquad \wedge\ (a_{p-1} = b_{p-1})\ \wedge\ (a_p < b_p)
\end{aligned}
$$

$$\langle \mathtt{r}, \vec{a} \rangle \prec \langle \mathtt{s}, \vec{b} \rangle \Longleftrightarrow \bigvee_{p=0}^{p=N_{r,s}} \langle \mathtt{r}, \vec{a} \rangle \prec_p \langle \mathtt{s}, \vec{b} \rangle$$

Décomposition de l'ordre de précédence :

$\langle \mathtt{r}, \vec{a} \rangle \prec_p \langle \mathtt{s}, \vec{b} \rangle \equiv$

Si $p < N_{r,s}$ alors

$$(a_1 = b_1) \wedge \ldots \wedge (a_p = b_p) \wedge (a_{p+1} < b_{p+1})$$

Si $p = N_{r,s}$ alors

$$(a_1 = b_1) \wedge \ldots \wedge (a_p = b_p) \wedge \mathtt{r} \lhd \mathtt{s}$$

Soit `t` une instruction qui utilise un tableau `M` référence résumée par la notation : $\mathtt{M}(\vec{g}(\vec{b}))$

Exemple :

```
DO I = 1, N
   DO J = 1, I-1
      DO K = i+1, n
         X(J,K) = X(J,k) - X(I,K)*X(J,I)/X(I,I)
      END DO
   END DO
END DO
```

Pour `X(I,K)` :

$$\vec{g}(i,j,k) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ k \end{pmatrix}$$

Quelle est la “source” de $\mathtt{M}(\vec{g}(\vec{b}))$ ?

Les candidats (instructions) doivent écrire dans le tableau `M` :

$s_i$ :   $\mathtt{M}(f_i(a_i)) = \ldots$

Il peut y avoir plusieurs candidats : $s_1, s_2, \ldots, s_n$

Si l'élément de tableau est non défini :
la source est notée "$\perp$" :
$\forall$ t, $\vec{b}$: $\perp \prec \langle \mathtt{t}, \vec{b} \rangle$

Notation:
Le vecteur d'itération de la *source*
associée au candidat $s_i$ : $\vec{K}_{s_i,t}(\vec{b})$

Conditions à remplir pour $\vec{K}_{s_i,t}(\vec{b})$ ?

- $\vec{f_i}(\vec{K}_{s_i,t}(\vec{b})) = \vec{g}(\vec{b})$
  (écriture dans le même élément)
- $\langle s_i, \vec{K}_{s_i,t}(\vec{b}) \rangle \prec \langle \mathtt{t}, \vec{b} \rangle$
  (ordre séquentiel)
- $e_{s_i}(\vec{K}_{s_i,t}(\vec{b})) \geq \vec{0}$
  (appartenance à l'espace d'itération)
- $\forall \vec{u}$ : $\vec{f_i}(\vec{u}) = \vec{g}(\vec{b})$
  $\wedge \langle s_i, \vec{u} \rangle \prec \langle \mathtt{t}, \vec{b} \rangle$
  $\wedge \; e_{s_i}(\vec{u}) \geq \vec{0}$
  $\Longrightarrow \vec{u} = \vec{K}_{s_i,t}(\vec{b})$ ou $\vec{u} \ll \vec{K}_{s_i,t}(\vec{b})$
  (pas d'autre écriture $\vec{K}_{s_i,t}(\vec{b})$ entre et $\langle \mathtt{t},\vec{b} \rangle$)

$$\vec{K}_{s_i,t}(\vec{b}) = \underset{\ll}{\mathrm{Max}}\; Q_{s_i,t}(\vec{b})$$

$$\vec{K}_{s_i,t}(\vec{b}) = \underset{\ll}{\text{Max}}\ Q_{s_i,t}(\vec{b})$$

$$\begin{aligned} Q_{s_i,t}(\vec{b}) = \ \{\ \vec{u}\ :\ & \vec{f}_i(\vec{u}) = \vec{g}(\vec{b}), \\ & \langle s_i, \vec{u}\rangle \prec \langle \mathtt{t}, \vec{b}\rangle, \\ & e_{s_i}(\vec{u}) \geq \vec{0} \\ \} & \end{aligned}$$

Décomposons $Q_{s_i,t}(\vec{b})$ selon la profondeur :

$$\begin{aligned} Q^p_{s_i,t}(\vec{b}) = \ \{\ \vec{u}\ :\ & \vec{f}_i(\vec{u}) = \vec{g}(\vec{b}) \\ & \langle s_i, \vec{u}\rangle \prec_p \langle \mathtt{t}, \vec{b}\rangle \\ & e_{s_i}(\vec{u}) \geq \vec{0} \\ \} & \end{aligned}$$

Pour chaque $p$ :
$\vec{K}^p_{s_i,t}(\vec{b}) = \underset{\ll}{\text{Max}}\ Q^p_{s_i,t}(\vec{b})$ : calculé par PIP

Finalement : la source cherchée est le max des vecteurs trouvés pour chaque $s_i$ et chaque $p$ :

$$S = \underset{\prec}{\text{Max}}\ \{\ \langle s_i, \vec{K}^p_{s_i,t}(\vec{b})\rangle\ \mid\ i = 1, \ldots, n,\ p = 0, \ldots, N_{s_i,t}\ \}$$

Exemple :

```
      DO k = 0, 2*n
S1:      C(k) = 0
      END DO
      DO k = 0, 2*n
         DO i = 0, n
S2:         C(i+j) = C(i+j) + A(i)*B(j)
         END DO
      END DO
```

Quelle est la *"source"* de `C(i+j)` dans `S2` ?
Deux candidats : `S1` et `S2`

$\vec{K}_{\mathtt{S1,S2}}[i,j] = [i+j]$

$\vec{K}_{\mathtt{S2,S2}}[i,j]$ ?

$[i',j'] \in Q_{\mathtt{S2,S2}}[i,j]$ ?

$$\left\{ \begin{array}{l} i'+j' = i+j \\ i' < i \vee (i' = i \wedge j' < j) \\ 0 \leq i' \leq n, \quad 0 \leq j' \leq n \end{array} \right.$$
$\Longrightarrow i' < i$

$Q_{S2,S2}[i,j] = \emptyset$ si $i = 0$ ou bien $j = n$
sinon : $\underset{\ll}{\text{Max}}\ Q_{S2,S2}[i,j] = [i-1, j+1]$

$$K_{S2,S2}[i,j] = \begin{array}{l} \text{if } (i \geq 1\ \wedge\ j < n) \\ \quad \text{then } [i-1, j+1] \\ \quad \text{else } \perp \end{array}$$

$N_{s1,s2} = 0, \quad (s1 \lhd s2) = $ True

Source de C(i+j) :

$K_{S2,S2}[i,j]$ si $\neq \perp$
$= \text{if } (i \geq 1 \ \wedge \ j < n)$
then $\langle \texttt{S2}$ , $[i-1, j+1]\rangle$
else $\langle \texttt{S1}$ , $[i+j]\rangle$

$\perp$ dans $\vec{K}_{s_j,t}(\vec{b})$ :
signifie que la source n'est pas une instance de $\mathsf{S}_j$

Évaluation par PIP

Source de la forme :
$\mathcal{T} = \underset{\prec}{\text{Max}} \ \{ \ \langle \mathsf{S}_j, \vec{K}_{s_j,t}(\vec{b})\rangle \ \mid \ j = 1, \ldots, L \ \}$

avec chaque $\vec{K}_{s_j,t}(\vec{b})$ calculé par PIP

Calculer $\mathcal{S}$ par itération :

$\mathcal{T}_0 = \perp$
for $j$ = $1$ to $L$
$\mathcal{T}_j = \underset{\prec}{\texttt{Max}}\,(\mathcal{T}_{j-1}, \ \langle \texttt{S}_j, \vec{K}_{s_j,t}(\vec{b})\rangle \ )$

Le calcul de la source :
à chaque pas de l'algorithme :

$$\mathcal{T}_j = \underset{\prec}{\mathsf{Max}}\ (\mathcal{T}_{j-1}, \langle \mathsf{S}_j, \vec{K}_{s_j,t}(\vec{b})\rangle)$$

Distinguer selon la forme de $\mathcal{T}_{j-1}$ :

- $\mathcal{T}_{j-1} = \perp$ :
  $\mathcal{T}_j = \langle \mathsf{S}_j, \vec{K}_{s_j,t}(\vec{b})\rangle$

- $\mathcal{T}_{j-1} =$ if $\pi(\vec{b})$ then T1 else T2 :
  $\mathcal{T}_j =$ if $\pi(\vec{b})$
  then $\underset{\prec}{\mathsf{Max}}$ (T1, $\langle \mathsf{S}_j, \vec{K}_{s_j,t}(\vec{b})\rangle$ )
  else $\underset{\prec}{\mathsf{Max}}$ (T2, $\langle \mathsf{S}_j, \vec{K}_{s_j,t}(\vec{b})\rangle$ )

- $\mathcal{T}_{j-1} = \langle$ `r`, $\vec{m}(\vec{b})$ $\rangle$ avec $\vec{m}(\vec{b})$ un quast :
  $\mathcal{T}_j =$ if ( $\langle$ `r`, $\vec{m}(\vec{b})$ $\rangle \prec \langle \mathsf{S}_j, \vec{K}_{s_j,t}(\vec{b})\rangle$ )
  then $\langle \mathsf{S}_j, \vec{K}_{s_j,t}(\vec{b})\rangle$
  else $\langle$ `r`, $\vec{m}(\vec{b})$ $\rangle$