今回の講義内容

- 今日の問題
- 3分割法, 黄金分割法
- 最急降下法, 準ニュートン法
- **線形計画問題について**
- シンプレックス法

今回の講義で解く問題 その2

$$\arg\max_{x} f(x)$$

$$\text{subject to } x \in F$$

ただし $f(x)$ は線形

今回の講義で解く問題 その2

より具体的には

$$\underset{x}{\arg\max}\ \boldsymbol{c}^T\boldsymbol{x}$$

$$\text{s.t. } A\boldsymbol{x} \leq \boldsymbol{b}$$

$$x_j \geq 0 \quad (j = 1, \dots, n)$$

ここで $\boldsymbol{c} = (c_1, \dots, c_n)^T, x = (x_1, \dots, x_n)^T,$
$A = \{a_{ij}\}, b = (b_1, \dots, b_n)$

今回の講義で解く問題 その2

線形制約付きの線形最適化問題 : **線形計画問題 (Linear Program : LP)**

例) x, yが不等式$2x + y \leq 8, x + 3y \leq 9, x \geq 0, y \geq 0$を満たすとき, $f(x, y) = x + y$ の値の最大値を求めよ.

実際の問題に置き換えると:
製品A,Bを作るのに以下の表に示した量の部品①,②を必要とする(表は在庫も含む). 製品A,Bはそれぞれ1万円の利益を出す. このとき利益を最大とするそれぞれの生産量を求めよ.

部品	製品Aに必要な数	製品Bに必要な数	在庫
①	2	1	8
②	1	3	9

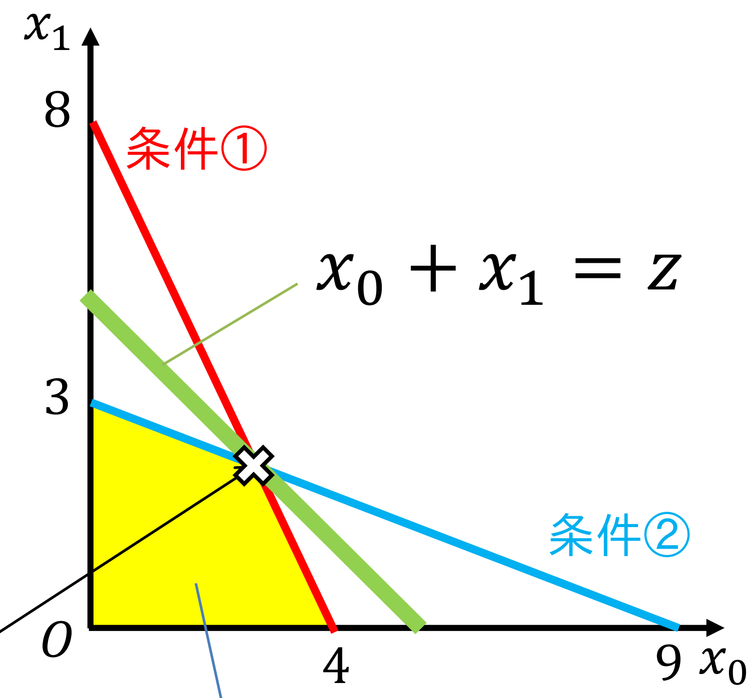
今回の講義で解く問題 その2

例) x, yが不等式$2x + y \leq 8, x + 3y \leq 9, x \geq 0, y \geq 0$を満たすとき, $f(x, y) = x + y$ の値の最大値を求めよ.

目的関数 : $z = x_0 + x_1$

制約条件 :

$$2x_0 + x_1 \leq 8 \quad \cdots ①$$
$$x_0 + 3x_1 \leq 9 \quad \cdots ②$$
$$x_0 \geq 0,\ x_1 \geq 0$$



制約条件を満たしつつ, zが最大となるのは目的関数がこの交点を通るとき

$\Rightarrow (x_0, x_1) = (3, 2)$で$z_{max} = 5$

$x_0 \geq 0, x_1 \geq 0$を含めて**すべての制約条件**を満たす領域

今回の講義で解く問題 その2

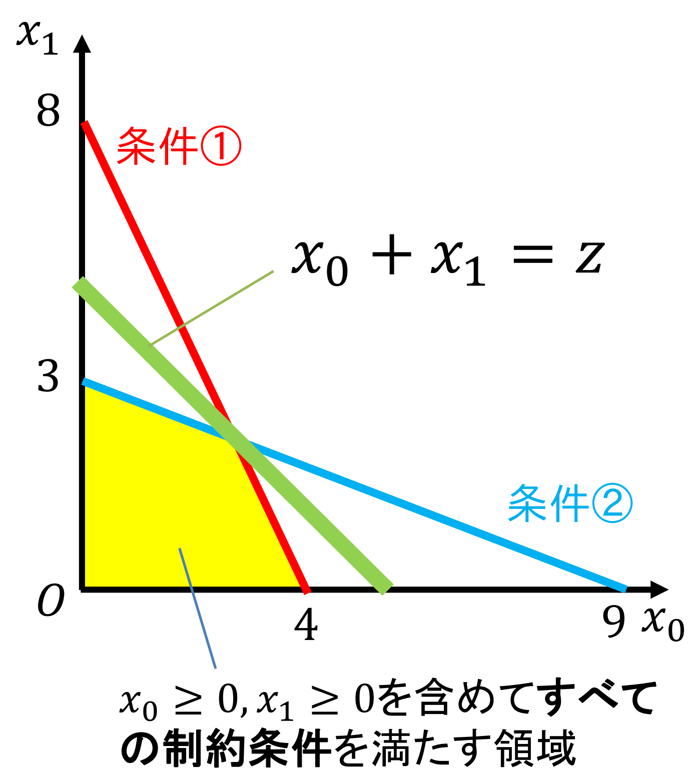
例) x, yが不等式$2x + y \leq 8, x + 3y \leq 9, x \geq 0, y \geq 0$を満たすとき, $f(x, y) = x + y$ の値の最大値を求めよ.

制約条件式が作る**すべての交点を目的関数が通る**ときのzの値を調べれば**解ける**

⇒ 変数や条件の数が増えると**交点数が膨大**になる

効率的に解ける方法

シンプレックス法



今回の講義内容

- 今日の問題
- 3分割法, 黄金分割法
- 最急降下法, 準ニュートン法
- 線形計画問題について
- **シンプレックス法**

シンプレックス法

手順1：スラック変数の導入

不等式が含まれると扱いづらい

⇒ **スラック変数**x_2, x_3を導入して**条件式を等式**にする

目的関数：$z = x_0 + x_1$

制約条件：

$$2x_0 + x_1 \leq 8 \quad \cdots ①$$

$$x_0 + 3x_1 \leq 9 \quad \cdots ②$$

$$x_0, x_1 \geq 0$$

⇨

$$z = x_0 + x_1$$

$$2x_0 + x_1 + x_2 = 8 \quad \cdots ①$$

$$x_0 + 3x_1 + x_3 = 9 \quad \cdots ②$$

$$x_0, x_1, x_2, x_3 \geq 0$$

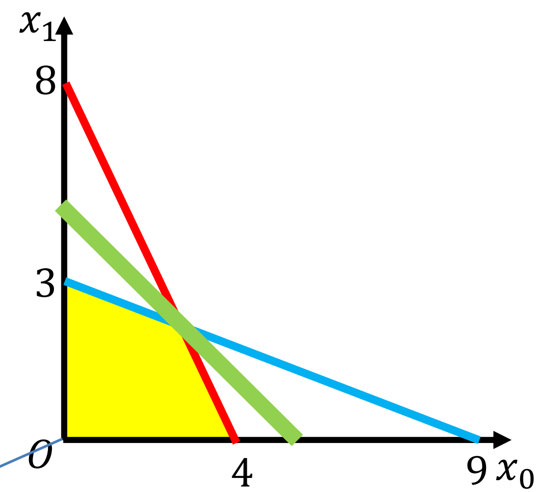
左辺の量が少ないので, 左辺に何らかの正の数x_2, x_3を足して等式にするということ

シンプレックス法

手順2：何でもいいので基底可能解を1つ見つける

基底可能解：すべての制約条件を満たし, 手順1で作成した方程式が成り立つ解(x_0, x_1) (スラック変数は解に含めない)

$$
\begin{aligned}
&z = x_0 + x_1 \\
&\quad 2x_0 + x_1 + x_2 \qquad\quad = 8 \qquad \cdots ① \\
&\quad x_0 + 3x_1 \qquad + x_3 = 9 \qquad \cdots ② \\
&\quad x_0, x_1, x_2, x_3 \geq 0
\end{aligned}
$$



原点, つまり$(x_0, x_1) = (0, 0)$なら条件を満たす領域内で方程式も成り立つ(この場合, $x_2 = 8, x_3 = 9$とすれば方程式が成り立つ)

シンプレックス法

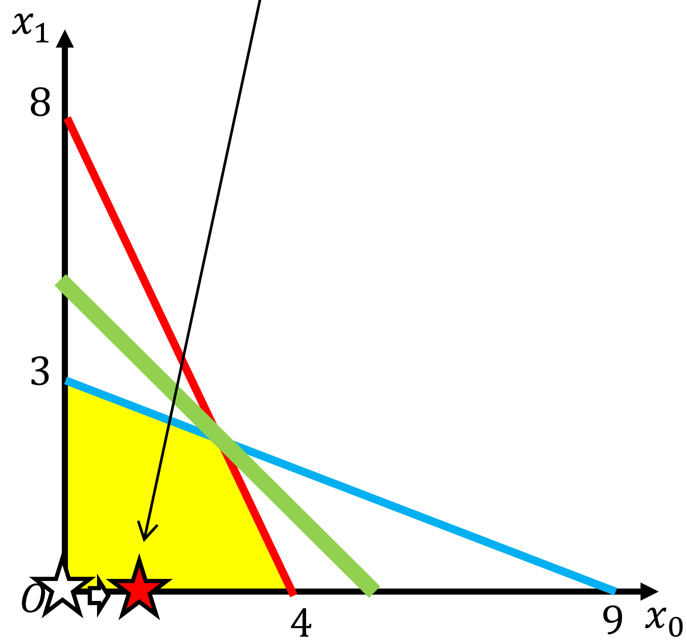
手順3：現在の解が最適解か調べる

現在の解$(x_0, x_1) = (0, 0)$ で例えば, x_0を1増やしても基底可能解のままで, zの値は大きくなる

⇒ **最適解ではない**

最適解ならば, $z = x_0 + x_1$**で**
x_0, x_1**の数を増やしても**z**は増えない.**

例えば, $z = -x_0 - x_1$なら
x_0, x_1を増やしてもzは増えないので$(0, 0)$が最適解となる
$(x_0, x_1 \geq 0$という条件込みで$)$



シンプレックス法

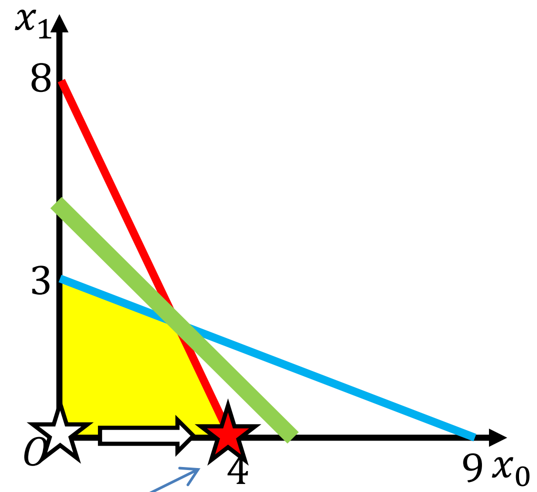
手順4：最適解でなければ解を改良して手順3に戻る

$(x_0, x_1) = (\alpha, 0)$ として, x_0をαだけ増やすことを考えてみる.

式①, ②に$(x_0, x_1) = (\alpha, 0)$を代入すると:

$$2\alpha + x_2 = 8 \Rightarrow x_2 = 8 - 2\alpha \geq 0$$
$$\alpha + x_3 = 9 \Rightarrow x_3 = 9 - \alpha \quad \underline{\geq 0}$$

$x_2, x_3 \geq 0$より



$\alpha \leq 4, \alpha \leq 9$ の両方を満たすαの最大値は $\alpha = 4$

⇩ 上の式に代入して現在の解を更新

$(x_0, x_1, x_2, x_3) = (4, 0, 0, 5)$ で $z = 4$

シンプレックス法

手順3(2回目)：現在の解が最適解か調べる

現在の解$(x_0, x_1, x_2, x_3) = (4, 0, 0, 5)$ で0となっているx_1, x_2について値を増やしてzが大きくなるか調べてみる

$$
\begin{aligned}
&z = x_0 + x_1 \\
&2x_0 + x_1 + x_2 = 8 \quad \cdots ① \\
&x_0 + 3x_1 + x_3 = 9 \quad \cdots ② \\
&x_0, x_1, x_2, x_3 \geq 0
\end{aligned}
$$

式①より：

$$x_0 = 4 - \frac{1}{2}x_1 - \frac{1}{2}x_2$$

式①以外に代入して，x_0の影響を消去

$$
\begin{aligned}
&z = 4 + \frac{1}{2}x_1 - \frac{1}{2}x_2 \\
&2x_0 + x_1 + x_2 = 8 \quad \cdots ① \\
&\frac{5}{2}x_1 - \frac{1}{2}x_2 + x_3 = 5 \quad \cdots ②' \\
&x_0, x_1, x_2, x_3 \geq 0
\end{aligned}
$$

zの式から，x_1を増やすとzが大きくなる

⇒ **まだ最適解ではない**

シンプレックス法

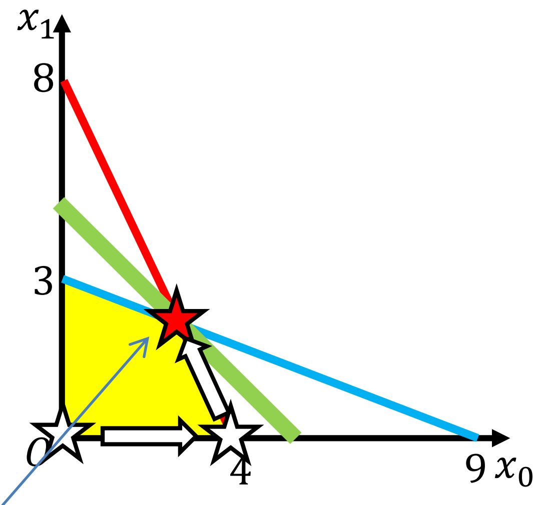
手順4(2回目)：最適解でなければ解を改良して手順3に戻る

$(x_1, x_2) = (\alpha, 0)$ として, x_1をαだけ増やすことを考えてみる.
式①, ②’に$(x_1, x_2) = (\alpha, 0)$を代入すると:

$$2x_0 + \alpha = 8 \Rightarrow x_0 = 4 - \frac{1}{2}\alpha \geq 0$$

$$\frac{5}{2}\alpha + x_3 = 5 \Rightarrow x_3 = 5 - \frac{5}{2}\alpha \geq 0$$

$\alpha \leq 8, \alpha \leq 2$ の両方を満たすαの最大値は $\alpha = 2$



⇩ 上の式に代入して現在の解を更新

$(x_0, x_1, x_2, x_3) = (3, 2, 0, 0)$ で $z = 5$

シンプレックス法

手順3(3回目)：現在の解が最適解か調べる

現在の解$(x_0, x_1, x_2, x_3) = (3, 2, 0, 0)$ で0となっているx_2, x_3について値を増やしてzが大きくなるか調べてみる

式②'より：

$$x_1 = 2 + \frac{1}{5}x_2 - \frac{2}{5}x_3$$

$$\begin{aligned} z &= 4 + \frac{1}{2}x_1 - \frac{1}{2}x_2 \\ 2x_0 + x_1 + x_2 &= 8 \quad \cdots ① \\ \frac{5}{2}x_1 - \frac{1}{2}x_2 + x_3 &= 5 \quad \cdots ②' \\ x_0, x_1, x_2, x_3 &\geq 0 \end{aligned}$$

式②'以外に代入して，x_1の影響を消去

$$\begin{aligned} z &= 5 - \frac{2}{5}x_2 - \frac{1}{5}x_3 \\ 2x_0 + \frac{6}{5}x_2 - \frac{2}{5}x_3 &= 6 \quad \cdots ①' \\ \frac{5}{2}x_1 - \frac{1}{2}x_2 + x_3 &= 5 \quad \cdots ②' \\ x_0, x_1, x_2, x_3 &\geq 0 \end{aligned}$$

zの式から，x_2, x_3を増やしてもzは大きくならない

$\Rightarrow$ **最適解！**

シンプレックス法

無事に最適解$(x_0, x_1) = (3, 2)$が得られたけど
プログラムにするのが難しそう. . .

$\Rightarrow$ **単体表**を使うことで機械的に解けるようになる!

スラック変数x_2, x_3を導入して, 目的関数
含めて問題を置き換え

$$\begin{aligned} & z = x_0 + x_1 \\ & \quad 2x_0 + x_1 + x_2 \qquad = 8 \\ & \quad x_0 + 3x_1 \qquad + x_3 = 9 \\ & \quad x_0, x_1, x_2, x_3 \geq 0 \end{aligned} \quad \Rightarrow \quad \begin{aligned} 2x_0 + \ \ x_1 + x_2 \qquad &= 8 \\ x_0 + 3x_1 \qquad + x_3 &= 9 \\ z - x_0 - \ \ x_1 \qquad\qquad &= 0 \end{aligned}$$

線形システム$(A\boldsymbol{x} = \boldsymbol{b})$になった?

シンプレックス法

手順1: **単体表(シンプレックス表)の作成**

各係数+最後の列に基底可能解
を並べた表を作る.

$$
\begin{aligned}
2x_0 + x_1 + x_2 \quad &= 8 \\
x_0 + 3x_1 \quad + x_3 &= 9 \\
z - x_0 - x_1 \quad &= 0
\end{aligned}
$$

基底可能解$(x_0, x_1) = (0, 0)$とすると
右辺項が(x_2, x_3)の基底可能解になる

基底変数	z	x_0	x_1	x_2	x_3	基底可能解
x_2	0	2	1	1	0	8
x_3	0	1	3	0	1	9
z	1	−1	−1	0	0	0

基底可能解が0
以外の変数+zを
基底変数にする

⇩

初期基底変数は
スラック変数になる

係数を並べて入力

シンプレックス法

手順2: 現在の解が最適解か調べる

一番下の行で, 非基底変数(現在はx_0とx_1)のところに**負の値**が入っていたらまだ**最適解ではない**

基底変数	z	x_0	x_1	x_2	x_3	基底可能解
x_2	0	2	1	1	0	8
x_3	0	1	3	0	1	9
z	1	-1	-1	0	0	0

非基底変数x_0, x_1の列の一番下に負の値があるので, まだ最適解ではない

シンプレックス法

手順3:最適解でなければ解を改良して手順2に戻る

負の値となっている非基底変数で**絶対値最大のもの**を選択し,
基底変数にする (絶対値が同じならどちらでもよい. 今回は左側を選択)

⇒ 前のやり方での x_0をαだけ増やすという手順に相当

基底変数	z	x_0	x_1	x_2	x_3	基底可能解
x_2	0	2	1	1	0	8
x_3	0	1	3	0	1	9
z	1	-1	-1	0	0	0

ピボット要素

ピボット要素の列の値で基底可能解の列の値を割ったときに値が最小となる行の基底変数を新しい基底変数に置き換え

(今回は基底可能解を割ると$\frac{8}{2}=4, \frac{9}{1}=9$で, $x_2 \rightarrow x_0$とする)

シンプレックス法

手順3(続き):最適解でなければ解を改良して手順2に戻る

置き換えた行を**ピボット要素で割る**

⇒ αの最大値を求めるための前段階

基底変数	z	x_0	x_1	x_2	x_3	基底可能解
x_0	0	2	1	1	0	8
x_3	0	1	3	0	1	9
z	1	-1	-1	0	0	0

青枠の値でx_0の行を割っていく

基底変数	z	x_0	x_1	x_2	x_3	基底可能解
x_0	0	1	1/2	1/2	0	4
x_3	0	1	3	0	1	9
z	1	-1	-1	0	0	0

シンプレックス法

手順3(続き):最適解でなければ解を改良して手順2に戻る

ピボット要素の**他の列がすべて0**になるように, 置き換えた行を**何倍かして他の行から引く**

⇒ 前のやり方でαの最大値を求めて新しい解を得ることに相当

基底変数	z	x_0	x_1	x_2	x_3	基底可能解
x_0	0	1	$1/2$	$1/2$	0	4
x_3	0	1	3	0	1	9
z	1	-1	-1	0	0	0

⇩ 青枠の値を何倍かして他の行から引く

基底変数	z	x_0	x_1	x_2	x_3	基底可能解
x_0	0	1	$1/2$	$1/2$	0	4
x_3	0	0	$5/2$	$-1/2$	1	5
z	1	0	$-1/2$	$1/2$	0	4

シンプレックス法

手順2と3(2回目): 現在の解が最適解か調べて改良

負の値となっている非基底関数で**絶対値最大のもの**を選択し，
基底変数にする (絶対値が同じならどちらでもよい. 今回は左側を選択)
⇒ 前のやり方での x_1をαだけ増やすという手順に相当

基底変数	z	x_0	x_1	x_2	x_3	基底可能解
x_0	0	1	1/2	1/2	0	4
x_3	0	0	5/2	−1/2	1	5
z	1	0	−1/2	1/2	0	4

ピボット要素で基底可能解を割った値から
基底変数x_3をx_1にしてその行を割る

基底変数	z	x_0	x_1	x_2	x_3	基底可能解
x_0	0	1	1/2	1/2	0	4
x_1	0	0	1	−1/5	2/5	2
z	1	0	−1/2	1/2	0	4

シンプレックス法

手順2と3(2回目): 現在の解が最適解か調べて改良

ピボット要素の**他の列がすべて0**になるように, 置き換えた行を**何倍かして他の行から引く**

⇒ 前のやり方でαの最大値を求めて新しい解を得ることに相当

基底変数	z	x_0	x_1	x_2	x_3	基底可能解
x_0	0	1	$1/2$	$1/2$	0	4
x_1	0	0	1	$-1/5$	$2/5$	2
z	1	0	$-1/2$	$1/2$	0	4

ピボット要素で基底可能解を割った値から
基底変数x_3をx_1にしてその行を割る

基底変数	z	x_0	x_1	x_2	x_3	基底可能解
x_0	0	1	0	$3/5$	$-1/5$	3
x_1	0	0	1	$-1/5$	$2/5$	2
z	1	0	0	$2/5$	$1/5$	5

シンプレックス法

手順2と3(2回目): 現在の解が最適解か調べて改良

一番下の行で，非基底変数(現在はx_2とx_2)のところに**負の値**が入っていないので**最適解に収束**

基底変数	z	x_0	x_1	x_2	x_3	基底可能解
x_0	0	1	0	$3/5$	$-1/5$	**3**
x_1	0	0	1	$-1/5$	$2/5$	**2**
z	1	0	0	$2/5$	$1/5$	**5**

最適解$(3, 2)$

最大値$z = 5$

負の値がないから
最適解に収束している

この列はzの係数列として
入れたけど，値は変わって
いないので**実際のプログ
ラムでは省いています**

シンプレックス法

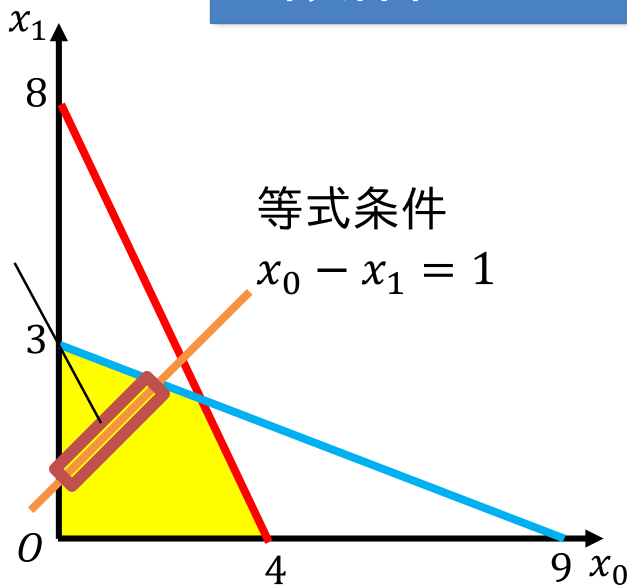
単体表を使った方法の問題

条件式に不等式でなく, 等式(例えば, $x_0 - x_1 = 1$)が入っていると, 最初の基底可能解$(0,0)$が成り立たない

⇒ スラック変数とは別に**人工変数**を更に追加して,
基底可能解を求めた後, 通常の単体表を使って解く

2段階シンプレックス法

解はこの線分上に含まれる



等式が含まれる場合だけでなく, 条件を満たす領域に原点が含まれない場合はこちらを使う
(条件不等式に≤と≥の両方が含まれる場合など)

シンプレックス法

シンプレックス法のコード例(単体表の作成部分)

```
int n = n_cond+1;      // 式の数(条件式の数+最適化式の数)
int m_slack = n_cond;  // スラック変数の数
int m_all = m+m_slack; // スラック変数を含む全変数の数
// 単体表の作成
vector< vector<double> > s(n); // 単体表の行数は条件式の数+1=n, 最後の行が目的関数
vector<int> xi(n); // それぞれの行の基底変数(x1=0,x2=1,...とインデックス値を格納)
for(int i = 0; i < n; ++i){
    s[i].resize(m_all+1); // 単体表の列数はスラック変数を含む全変数の数+1,最後列は基底可能解
    if(i != n-1){ // 条件式の行(0〜n-2行)
        xi[i] = i+m; // 初期基底変数はスラック変数
        int sgn = (a[i][m] < 0 ? -1 : 1); // 条件式の右辺項の符号
        for(int j = 0; j < m; ++j) s[i][j] = sgn*a[i][j];// 変数の係数を入れていく
        // スラック変数部分の初期係数は単位行列のような形に(ただし条件式の符号が>=の場合は-1をセット)
        for(int j = m; j < m_all; ++j) s[i][j] = (xi[i] == j ? sgn*eqn[i] : 0);
        s[i][m_all] = sgn*a[i][m];// 右辺項を初期基底可能解としてセット
    }
    else{// 最終行は最適化式(ここだけ別で設定)
        xi[i] = -1; // 最適化式の変数インデックスには-1を格納しておく
        for(int j = 0; j < m; ++j) s[i][j] = -a[i][j];// 最終行の最適化式は係数の符号を反転
        // 最後の行の最適化式では初期基底可能解に0をセット
        for(int j = m; j < m_all+1; ++j) s[i][j] = 0;
    }
}
```

シンプレックス法

シンプレックス法のコード例(反復処理部分, ピボット要素の探索まで)

```
int k;
for(k = 0; k < max_iter; ++k){
    // 非基底変数(初期状態ではスラック変数以外)から負で絶対値最大のものを選択
    int b = -1;
    double xmax = 0.0;
    for(int j = 0; j < m_all; ++j){
        if(s[n-1][j] < 0 && -s[n-1][j] > xmax){ b = j; xmax = -s[n-1][j]; }
    }
    if(b == -1) break; // 負の変数が見つからなければ収束したとしてループを抜ける

    // ピボット要素の探索
    int p = 0;
    double ymin = s[0][m_all]/s[0][b];
    for(int i = 1; i < n-1; ++i){
        // 基底可能解をb列の値で割った値が最小のものを選択
        double y = s[i][m_all]/s[i][b];
        if(y < ymin){ p = i; ymin = y; }
    }

    // ピボット要素行を選択された非基底変数(b)で置き換えて, ピボット要素でその行を割る
    ・・・(次ページ参照)・・・
}
```

シンプレックス法

シンプレックス法のコード例(反復処理部分, ピボット要素による処理)

```
int k;
for(k = 0; k < max_iter; ++k){
    // ピボット要素の探索
    ・・・(前ページ参照)・・・

    // ピボット要素行を選択された非基底変数(b)で置き換えて, ピボット要素でその行を割る
    xi[p] = b;
    double xp = s[p][b];// ループ内で値が変わるので一時変数に値を確保しておく
    for(int j = 0; j < m_all+1; ++j){
        s[p][j] /= xp;
    }

    // ピボット行(p)以外の行を非基底変数(b)の値が0になるようにピボット行の係数をx倍して引く
    //  -> s[p][b]は1になっているのでs[p][j]にs[i][b]を掛けて引けば良い
    for(int i = 0; i < n; ++i){
        if(i == p) continue; // ピボット行は飛ばす
        xp = s[i][b];// ループ内で値が変わるので一時変数に値を確保しておく
        for(int j = 0; j < m_all+1; ++j){
            s[i][j] -= s[p][j]*xp;
        }
    }
}
```

シンプレックス法

シンプレックス法の実行結果例

目的関数：$z = x_0 + x_1$
制約条件：

$$2x_0 + x_1 \leq 8 \quad \cdots ①$$
$$x_0 + 3x_1 \leq 9 \quad \cdots ②$$
$$x_0, x_1 \geq 0$$

最適解 $(x_0, x_1) = (3, 2)$で
目的関数の値$z = 5$

```
初期単体表
x3:        2        1        1        0        8
x4:        1        3        0        1        9
 z:       -1       -1        0        0        0
処理後の単体表
x1:        1        0      0.6     -0.2        3
x2:        0        1     -0.2      0.4        2
 z:        0        0      0.4      0.2        5
f(3,2) = 5
```