

連続系の数理計画法アルゴリズム とその応用

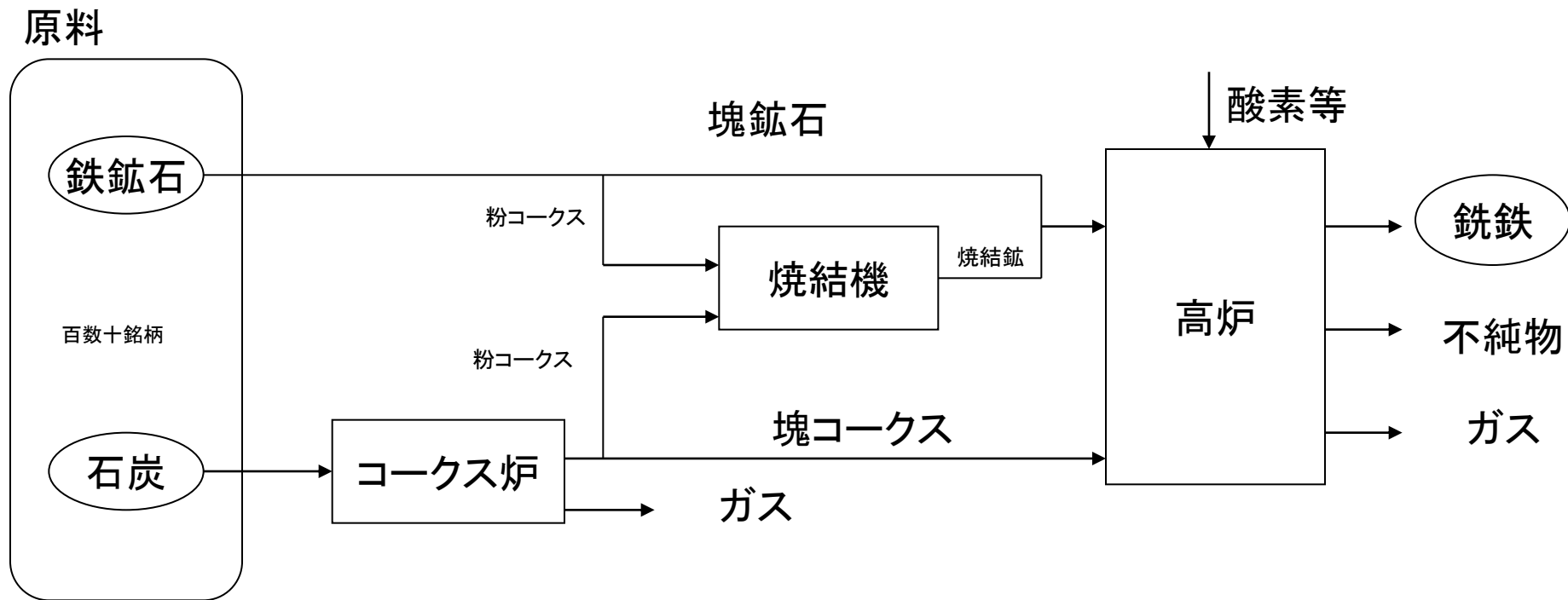
田辺隆人

`tanabe@msi.co.jp`

`nuopt-info@msi.co.jp`

(株)数理システム

銑鉄製造プロセスの一例



ケースA(製鉄工場):

- 製鉄工場全体の生産最適化システムの一部.
- 最適化問題は「鉄鉱石の配合問題 + 工場間の輸送問題」として作られたLPに, 性状(製品に含まれる元素量などの条件)を定義する非線形な制約式が付加した問題.
- 典型的問題(#変数 4166, #制約 2172)をNUOPT(内点法)で反復回数 40, 計算時間 103 秒で解ける.

問題定義のためのドキュメント

目次		式番号	頁
1. 銘柄別購入量制約式	1.1 配合炭購入量制約式	Ta(001) ~ (003)	
	1.2 無煙炭購入量制約式	Ta(004) ~ (006)	
	1.3 微粉炭購入量制約式	Ta(007) ~ (009-2)	
	1.4 重油購入量制約式	Ta(010) ~ (012)	
	1.5 コークス購入量制約式	Ta(013) ~ (018)	
2. 購入量集計定義式	2.1 配合炭装入量定義式	Ta(019) ~ (020)	
	2.2 微粉炭購入量定義式	Ta(021) ~ (022)	
	2.3 コークス購入量定義式	Ta(023) ~ (026)	
3. 品位集計定義式	3.1 配合炭品位定義式	Ta(027) ~ (031)	
	3.2 微粉炭品位定義式	Ta(032) ~ (037)	
	3.3 重油品位定義式	Ta(038) ~ (039)	
	3.4 購入コークス品位定義式	Ta(040) ~ (045)	
4. 平均コークス強度定義式		Ta(046) ~ (052-8)	
5. コークス生産量定義式		Ta(053) ~ (057)	
6. 生産コークス品位定義式		Ta(058) ~ (064)	
7. コークス余剰量定義式		Ta(065) ~ (067)	
8. 余剰コークス品位定義式		Ta(068) ~ (073)	
9. コークス品位制約式	9.1 高炉装入小塊コークス品位制約式	Ta(074) ~ (076)	
	9.2 輸送コークス品位制約式	Ta(077) ~ (082)	
	9.3 外販塊、小塊コークス品位制約式	Ta(083) ~ (088-3)	
	9.4 外販粉コークス品位制約式	Ta(089) ~ (094)	
	9.5 余剰コークス品位制約式	Ta(095) ~ (100)	
10. 燃料比定義式		Ta(101) ~ (101-1)	
11. 燃料比制約式		Ta(102) ~ (106)	

4. 平均コークス強度定義式		Ta(046) ~ (052-8)	
5. コークス生産量定義式		Ta(053) ~ (057)	
6. 生産コークス品位定義式		Ta(058) ~ (064)	
7. コークス余剰量定義式		Ta(065) ~ (067)	
8. 余剰コークス品位定義式		Ta(068) ~ (073)	
9. コークス品位制約式	9.1 高炉装入小塊コークス品位制約式	Ta(074) ~ (076)	
	9.2 輸送コークス品位制約式	Ta(077) ~ (082)	
	9.3 外販塊、小塊コークス品位制約式	Ta(083) ~ (088-3)	
	9.4 外販粉コークス品位制約式	Ta(089) ~ (094)	
	9.5 余剰コークス品位制約式	Ta(095) ~ (100)	
10. 燃料比定義式		Ta(101) ~ (101-1)	
11. 燃料比制約式		Ta(102) ~ (106)	
12. コークス炉操業条件制約式		Ta(107) ~ (113)	
13. 微粉炭吹込操業条件制約式		Ta(114) ~ (115)	
14. コークス量制約式	14.1 コークス余剰量制約式	Ta(116) ~ (118)	
	14.2 コークス輸送量制約式	Ta(119) ~ (120)	
	14.3 コークス外販量制約式	Ta(121)	
15. 原料炭配合比率定義式 16. 原料炭配合比率制約式 (欠番)		Ta(122) ~ (129)	
17. コスト定義式	17.1 コークス炉操業コスト定義式	Ta(130) ~ (137)	
	17.2 Bガス控除定義式	Ta(138)	
	17.3 微粉炭吹込コスト定義式	Ta(139) ~ (142)	
	17.4 重油吹込コスト定義式	Ta(143)	
	17.5 購入・外販コークスコスト定義式	Ta(144) ~ (152)	
	17.6 工場間輸送コークスコスト定義式	Ta(153) ~ (155)	
18. コークス総コスト定義式・その他		Ta(156), Ta(160)	

ドキュメント内容の一部

3. 品位集計定義式

3.1 配合炭品位定義式

(027) 配合炭VM定義式 $_DTCOKVM <L>$

$$\sum_i [\text{配合炭VM割合}]_i * \{1 - [\text{配合炭中水分}]_i\} * (\text{配合炭購入量})_i = (\text{配合炭VM})$$
$$i \quad ZH024i \quad \quad \quad ZH028i \quad \quad \quad _Xi \quad \quad \quad _YTCOKVM$$

(注) i=A00,...,A99,B00,...,B99(配合炭全銘柄)についての和。

(027-1) 配合炭CI定義式 $_DTCOCL <L>$

$$\sum_i [\text{配合炭CI割合}]_i * \{1 - [\text{配合炭中水分}]_i\} * (\text{配合炭購入量})_i = (\text{配合炭CI})$$
$$i \quad ZH057i \quad \quad \quad ZH028i \quad \quad \quad _Xi \quad \quad \quad _YTCOCL$$

(注) i=A00,...,A99,B00,...,B99(配合炭全銘柄)についての和。

(027-2) 配合炭S定義式 $_DTCOSS <L>$

$$\sum_i [\text{配合炭S割合}]_i * \{1 - [\text{配合炭中水分}]_i\} * (\text{配合炭購入量})_i = (\text{配合炭S})$$
$$i \quad ZH011i \quad \quad \quad ZH028i \quad \quad \quad _Xi \quad \quad \quad _YTCOSS$$

(注) i=A00,...,A99,B00,...,B99(配合炭全銘柄)についての和。

(028) 生産コークスASH定義式 $_DTCKASH <L>$

$$\sum_i [\text{配合炭ASH割合}]_i * \{1 - [\text{配合炭中水分}]_i\} * (\text{配合炭購入量})_i = (\text{生産コークスASH})$$
$$i \quad ZH023i \quad \quad \quad ZH028i \quad \quad \quad _Xi \quad \quad \quad _YTCKASH$$

(注) i=A00,...,A99,B00,...,B99(配合炭全銘柄)についての和。

(029) 生産コークスASH中化学成分定義式 $_DTCKA??, ??=FE,SI,AL,CA,MG,MN,PP,TI,ZN,KO,NA,VV <L>$

$$\sum_i [\text{配合炭中化学成分割合}]_i * \{1 - [\text{配合炭中水分}]_i\} * (\text{配合炭購入量})_i = (\text{生産コークスASH中化学成分})$$
$$i \quad ZHD**i \quad \quad \quad ZH028i \quad \quad \quad _Xi \quad \quad \quad _YTCKA??$$

(注) 化学成分は、T・Fe, SiO₂, Al₂O₃, CaO, MgO, Mn, P, TiO₂, Zn, K₂O, Na₂O, V の12種類。

(注) **=02,04,05,06,07,09,10,08,12,21,20,19

??=FE,SI,AL,CA,MG,MN,PP,TI,ZN,KO,NA,VV respectively

(注) i=A00,...,A99,B00,...,B99(配合炭全銘柄)についての和。

(030) 生産コークスS定義式 $_DTCKSUL <L>$

$$\sum_i [\text{配合炭中コークス残留S割合}]_i * \{1 - [\text{配合炭中水分}]_i\} * (\text{配合炭購入量})_i = (\text{生産コークスS})$$
$$i \quad ZHDSRi \quad \quad \quad ZH028i \quad \quad \quad _Xi \quad \quad \quad _YTCKSUL$$

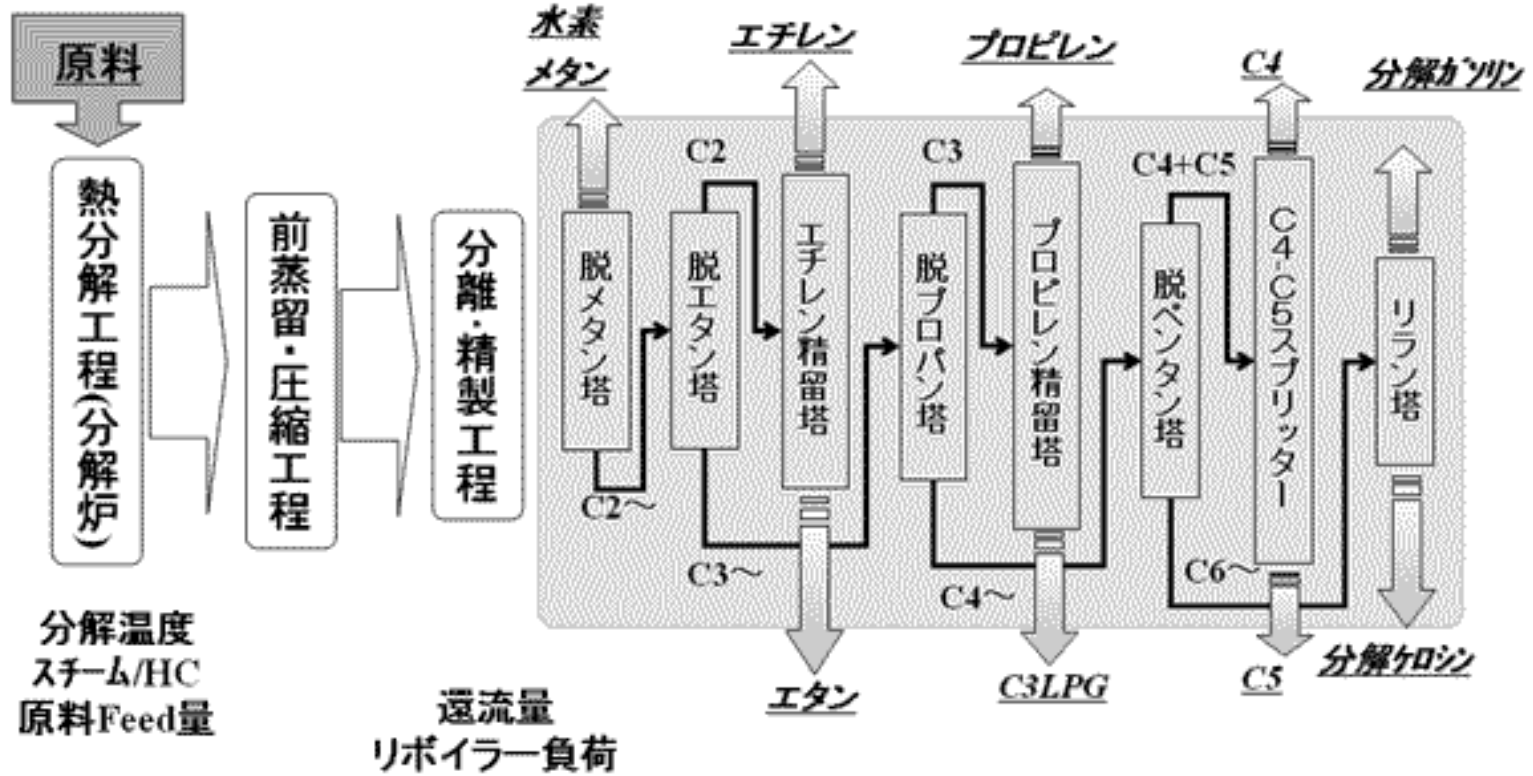
(注) i=A00,...,A99,B00,...,B99(配合炭全銘柄)についての和。

(031) Cガス中S定義式 $_DTCGASS <L>$

$$\sum_i [\text{配合炭中Cガス残留率割合}]_i * \{1 - [\text{配合炭中水分}]_i\} * (\text{配合炭購入量})_i = (\text{Cガス中S})$$
$$i \quad ZHDSGi \quad \quad \quad ZH028i \quad \quad \quad _Xi \quad \quad \quad _YTCGASS$$

(注) i=A00,...,A99,B00,...,B99(配合炭全銘柄)についての和。

エチレンプラントの一例



ケースB (エチレンプラント):

- エチレンプラント工場の最適化で多くの非線形(物理化学)モデルからなる.
- 典型的な問題(# 変数 439, #制約 294)が NUOPT(内点法)で反復回数 69, 計算時間 212 秒で解ける.

ケースC(エチレンプラント):

- これもエチレンプラント工場の最適化であるが、殆どの部分が線形モデルからなり、温度パラメータの部分のモデルにのみ非線形性が入っている。他のモデルと異なるのは時系列的な要素が入っている。
- 典型的な問題(# 変数 76868, #制約 10880)をNUOPT(内点法)で 反復回数 82, 計算時間 62 秒で解く。

数理計画法アルゴリズム

- 線形計画法(単体法、内点法)
- 二次計画法(有効制約法、内点法)
- 非線形計画法(内点法, 逐次二次計画法)
- 非線形半正定値計画(内点法)
- 線形混合整数計画法(分枝限定法)
- 混合整数二次計画法(分枝限定法)
- 非線形整数計画法(分枝限定法)
- 制約充足問題(タブ・サーチ)
- 資源制約付プロジェクトスケジューリング(タブ・サーチ)

線形計画問題(標準形)

最小化 $c^t x, \quad x \in \mathbf{R}^n,$

条件 $Ax = b, \quad A \in \mathbf{R}^{m \times n},$

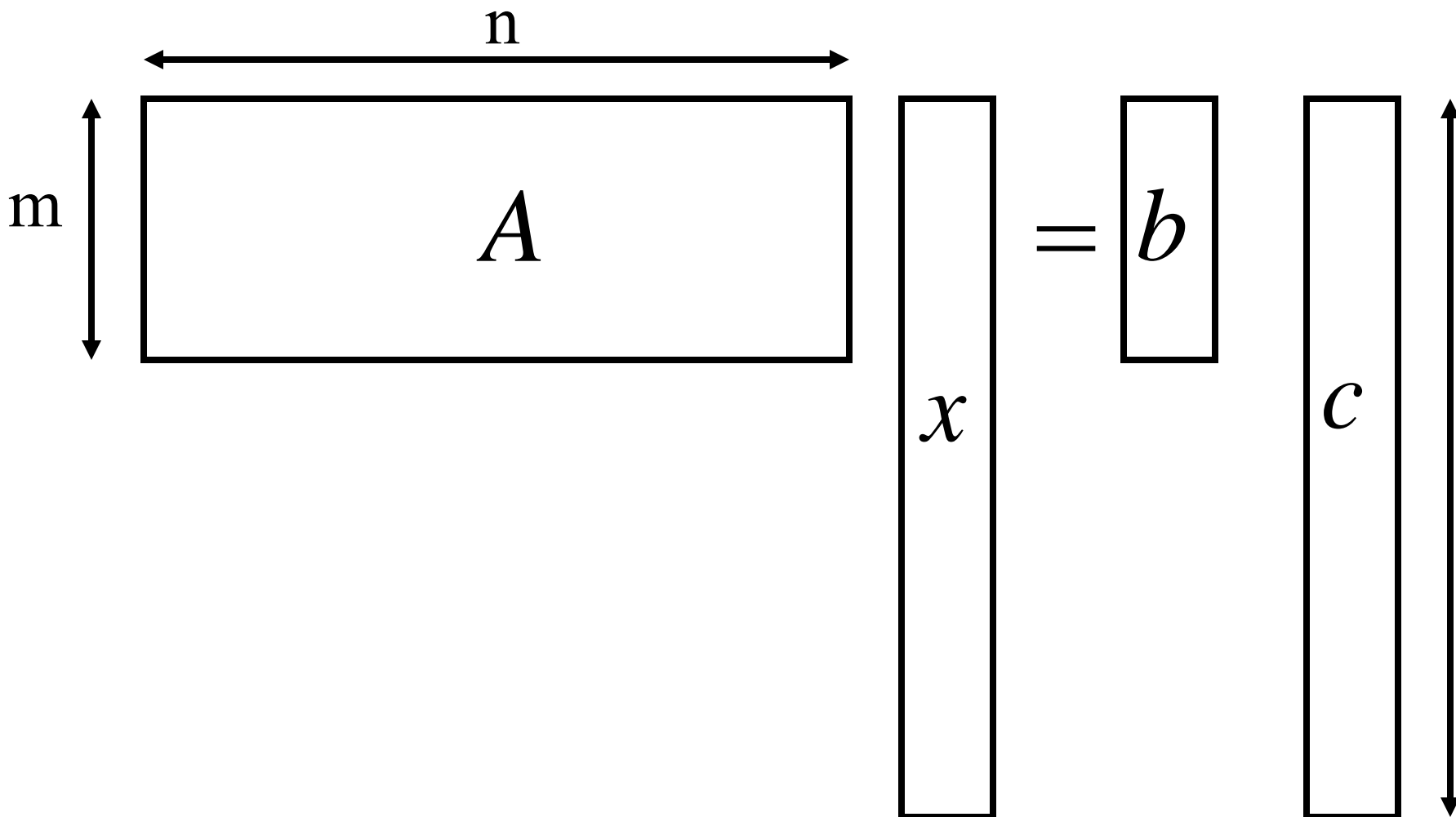
$$x \geq 0$$

標準形への変形

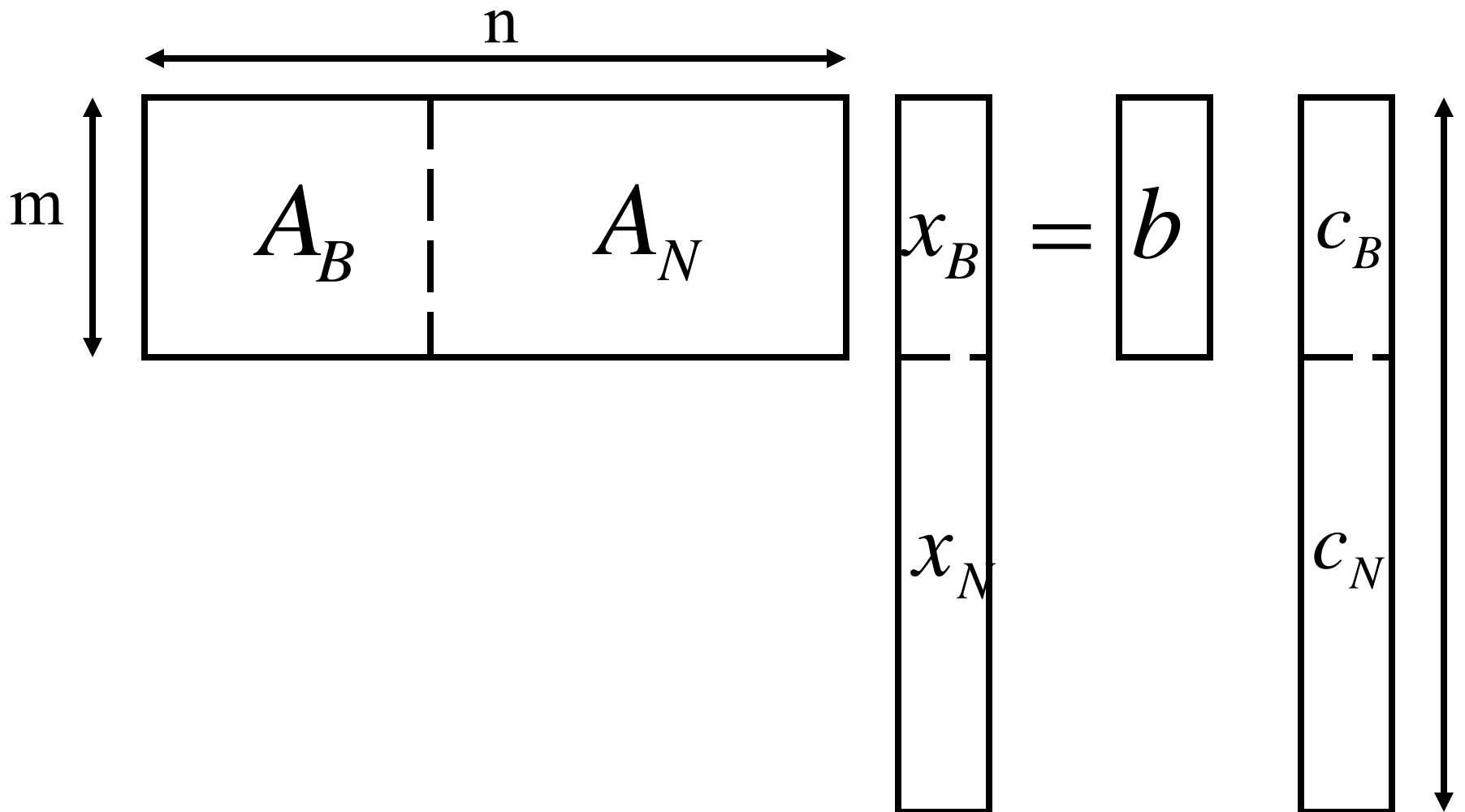
$$Ax \geq b \quad \Rightarrow \quad \begin{aligned} Ax - s &= b, \\ s &\geq 0 \end{aligned}$$

$$Ax \leq b \quad \Rightarrow \quad \begin{aligned} Ax + s &= b, \\ s &\geq 0 \end{aligned}$$

変数の分割..



変数の分割



線形計画問題(変数分割後)

最小化 $c_B^t x_B + c_N^t x_N,$

条件 $A_B x_B + A_N x_N = b,$

$$x_B \geq 0, x_N \geq 0$$

知られている性質

適当な分割

$$x = (x_B \mid x_N)$$

を行い、

$$x_N = 0$$

非基底

基底

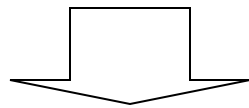
としたものが最小化問題の解になる。

求解アルゴリズム(単体法)

- 分割を探す \Leftrightarrow 解を探す

$$A_B x_B + A_N x_N = b$$

$$x = (x_B \mid x_N) = (x_B \mid 0)$$



$$x_B = A_B^{-1} b$$

分割があれば
解の取得は**容易**

求解アルゴリズム(単体法)

- 初期設定

$$x = (x_B \mid x_N) = (x_B \mid 0)$$

$$x_B = A_B^{-1}b \geq 0$$

なる分割を得る

求解アルゴリズム(単体法)

- 目的関数の評価

x_N のいずれかの要素は
正にした方が良い？

求解アルゴリズム(単体法)

- 目的関数を x_N で表す

$$c^t x = c_B^t x_B + c_N^t x_N$$

$$= c_B^t A_B^{-1} (b - A_N \cdot x_N) + c_N^t x_N$$

$$= \underline{(c_N - A_N^t y)^t} x_N + b^t y$$

$$y \equiv (A_B^t)^{-1} c_B$$

求解アルゴリズム(単体法)

- 停止条件判定

ベクトル: $c_N - A_N^t y$

すべて正なら
探索停止
(最適解)

で、負の要素があれば、

対応する x_N は正にした方が良い。

求解アルゴリズム(単体法)

- プライシング

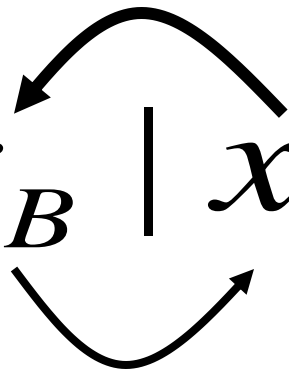
ベクトル: $c_N - A_N^t y$

の中で負となる要素のうちから
一つを選択する

求解アルゴリズム(単体法)

- ピボットティング

選択した一つを基底に入れる

$$x = (x_B \mid x_N)$$


非零の変数のうち一つを非基底に

求解アルゴリズム(単体法)

- 基底から出すものを決定

入れるもの

$$x_B = A_B^{-1}b + x_i \cdot (A_B^{-1}a_i)$$

0から増大

≥ 0

いずれかの要素が当たる

求解アルゴリズム(単体法)

- 初期設定
 - 初期の分割
- 最適性の判定
 - $c_N - A_N^t y$ の評価
- プライシング
 - 基底に入れるものの選択
- ピボットティング
 - 交換

研究は

現在も続く

求解アルゴリズム(単体法)

- 特徴

- 分割 \Leftrightarrow 解

- 小～中規模問題に適する

- 似通った問題群を高速に求解可能

- \Rightarrow 分枝限定法への応用

最適性条件

- 解の分割が得られたのなら...

$$y \equiv (A_B^t)^{-1} c_B \Leftrightarrow c_B - A_B^t y \equiv z_B = 0$$

$$c_N - A_N^t y \equiv z_N \geq 0$$

最適性条件

- 主変数との対応

$$c_B - A_B^t y \equiv z_B = 0 \quad \Longrightarrow \quad x_B \geq 0$$

$$c_N - A_N^t y \equiv z_N \geq 0 \quad \Longrightarrow \quad x_N = 0$$

x と z のいずれか一方は0

最適性条件 (線形計画)

- KKT条件 (最適性の必要条件)

$$Ax = b,$$

$$c - A^t y - z = 0,$$

$$Xz = 0,$$

$$x \geq 0, z \geq 0$$

線形な場合

線形計画問題(標準形)

最小化 $c^t x, \quad x \in \mathbf{R}^n,$

条件 $Ax = b, \quad A \in \mathbf{R}^{m \times n},$

$$x \geq 0$$

非線形計画問題(標準形)

最小化 $f(x), \quad x \in \mathbf{R}^n,$
条件 $g(x) = 0, \quad g(x) \in \mathbf{R}^m,$
 $x \geq 0$

最適性条件 (非線形計画)

- KKT条件 (最適性の必要条件)

$$g(x) = 0,$$

$$\nabla f(x) - A^t y - z = 0,$$

$$Xz = 0,$$

$$x \geq 0, z \geq 0$$

非線形な場合

$$A \equiv \nabla g(x)$$

求解アルゴリズム(内点法)

- KKT条件を非線形方程式として解こう！

$$g(x) = 0,$$

$$\nabla f(x) - A^t y - z = 0,$$

$$Xz = 0$$

$$x \geq 0, z \geq 0$$

線形・非線形の大きな差はない

非線形計画問題のための 内点法

- KKT条件を反復的に解く

$$\nabla f(x) - A(x)^t y - z = 0$$

$$g(x) = 0$$

$$XZe = 0$$

$$x \geq 0, \quad z \geq 0$$

$$X = \text{diag}(x_1, \dots, x_n), \quad Z = \text{diag}(z_1, \dots, z_n), \quad e = (1, \dots, 1)^t$$

非線形計画問題のための 内点法

- KKT条件を反復的に解く

$$\nabla f(x) - A(x)^t y - z = 0$$

$$g(x) = 0$$

$$XZe = \mu$$

$$x > 0,$$

$$z > 0$$

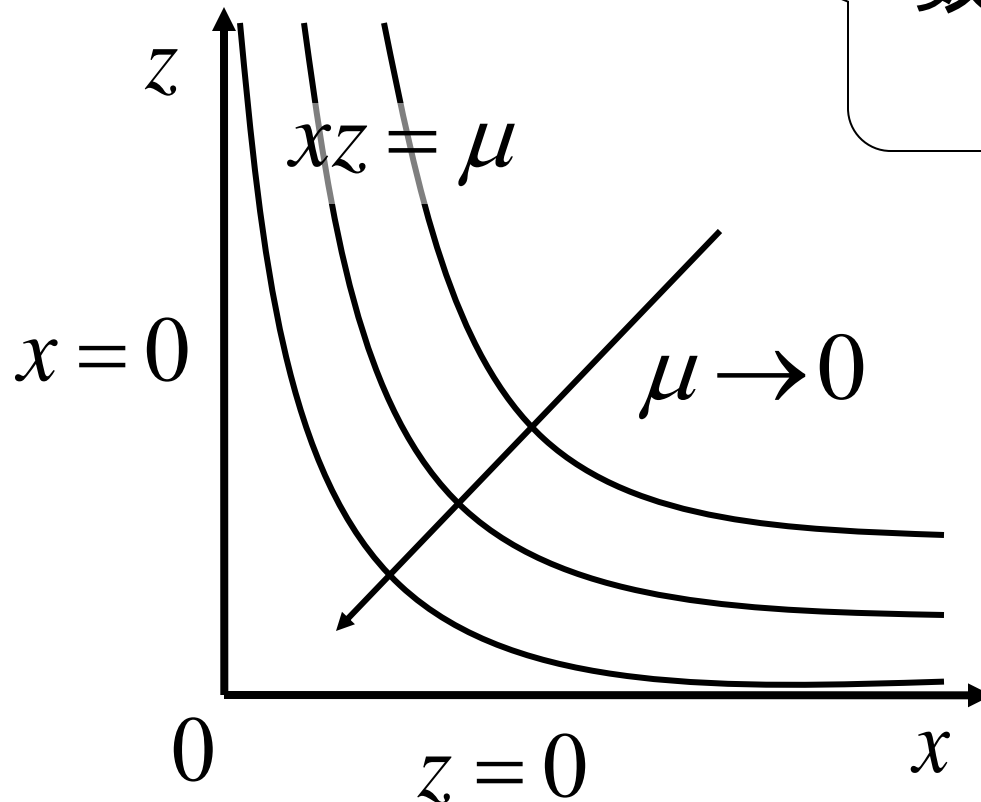
解法テクニック

$$X = \text{diag}(x_1, \dots, x_n), Z = \text{diag}(z_1, \dots, z_n), e = (1, \dots, 1)^t$$

求解アルゴリズム(内点法)

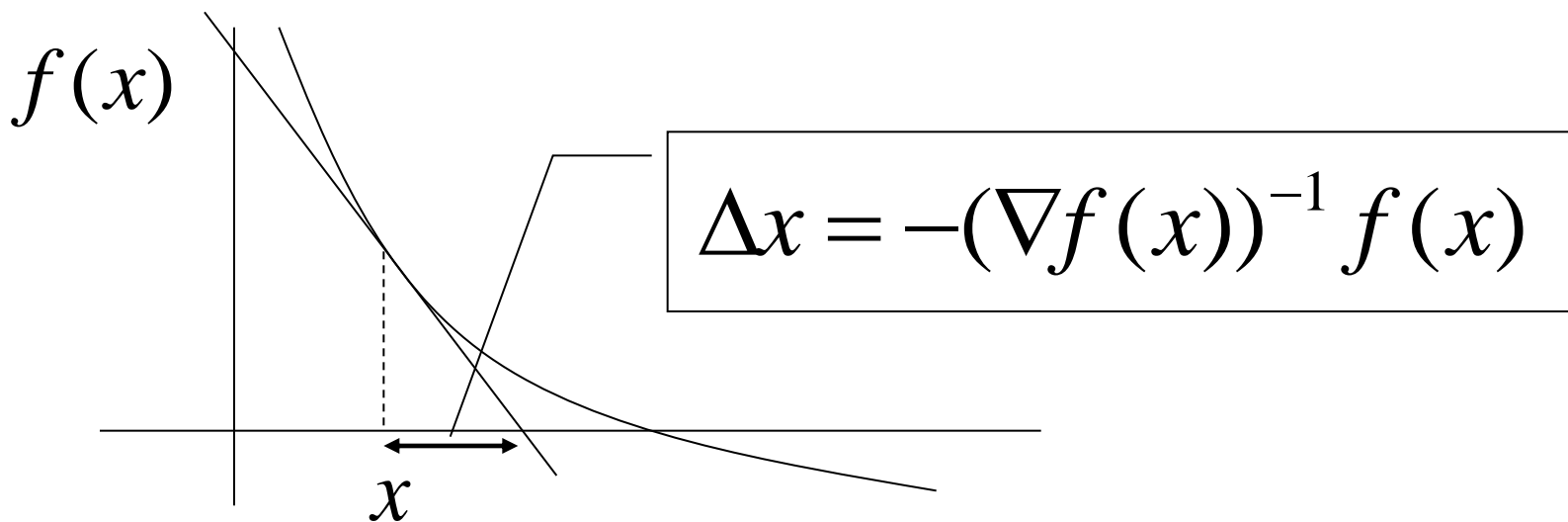
- 相補性条件の緩和

数値的に分割を
決定

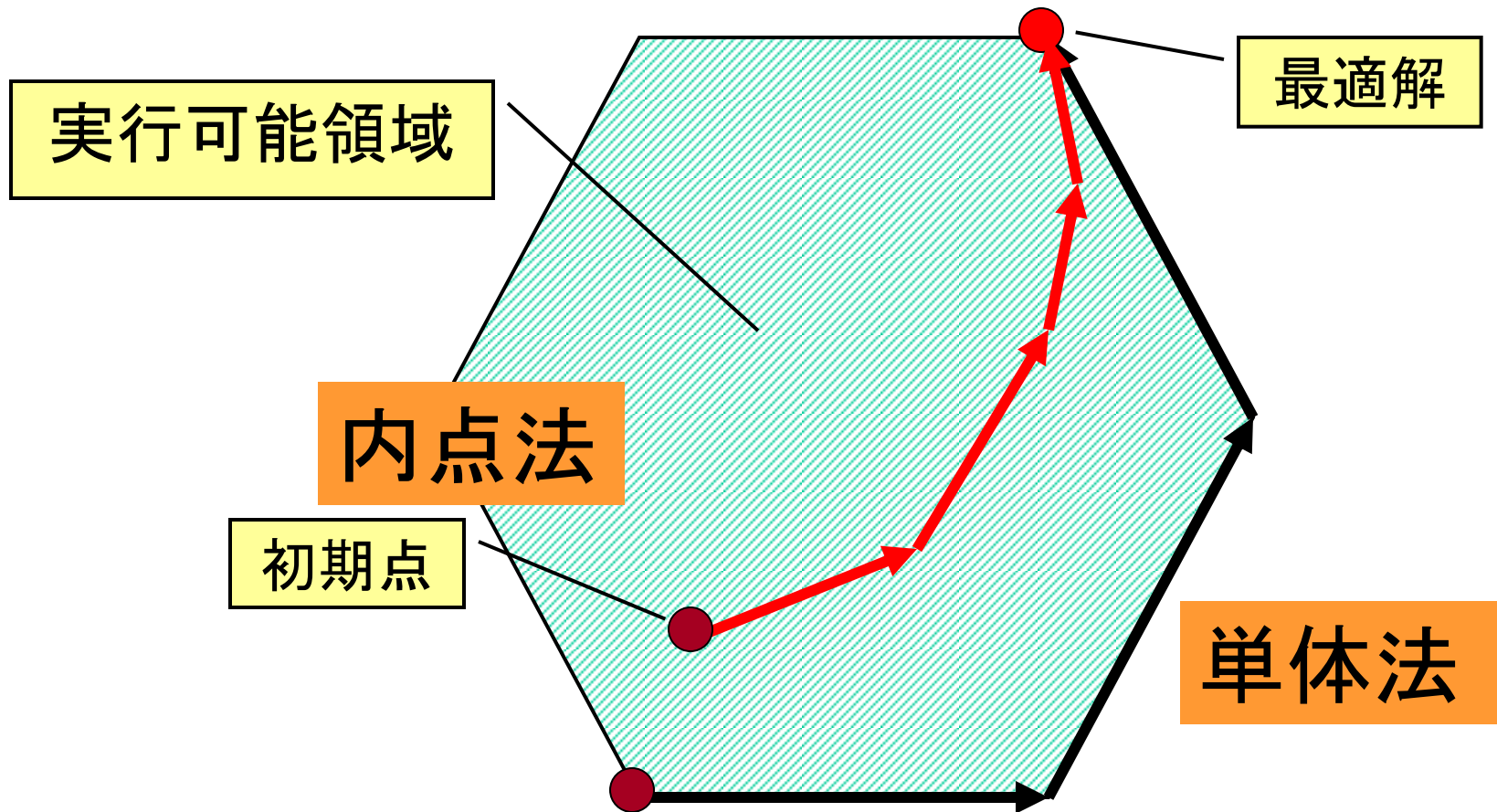


Newton法

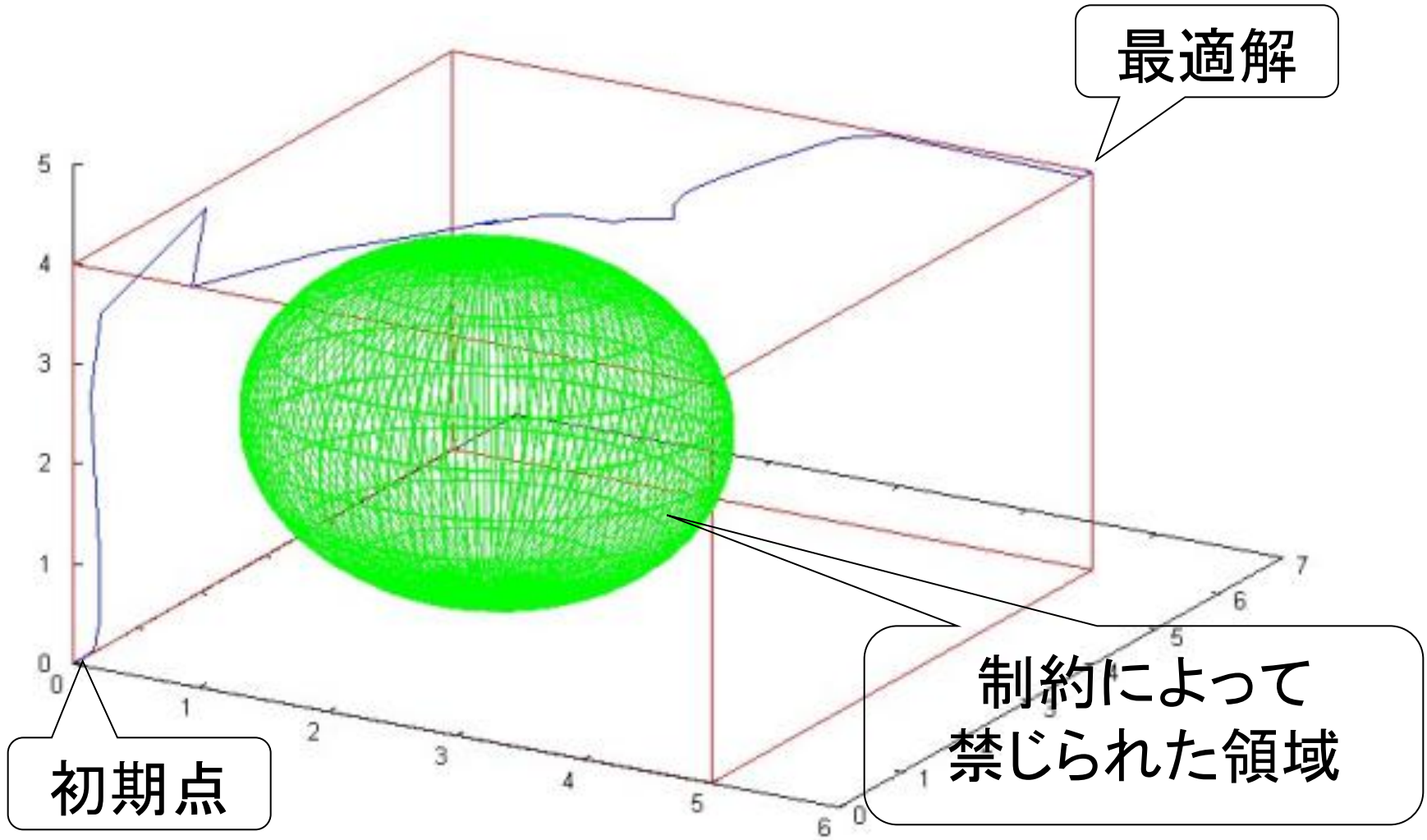
$$f(x + \Delta x) \simeq f(x) + \nabla f(x)\Delta x = 0$$



概念图(单体法.VS.内点法)



内点法による反復の例



最適解

初期点

制約によって
禁じられた領域

NUOPTの主双対内点法

理論的結果

- メリット関数(バリアペナルティ関数)

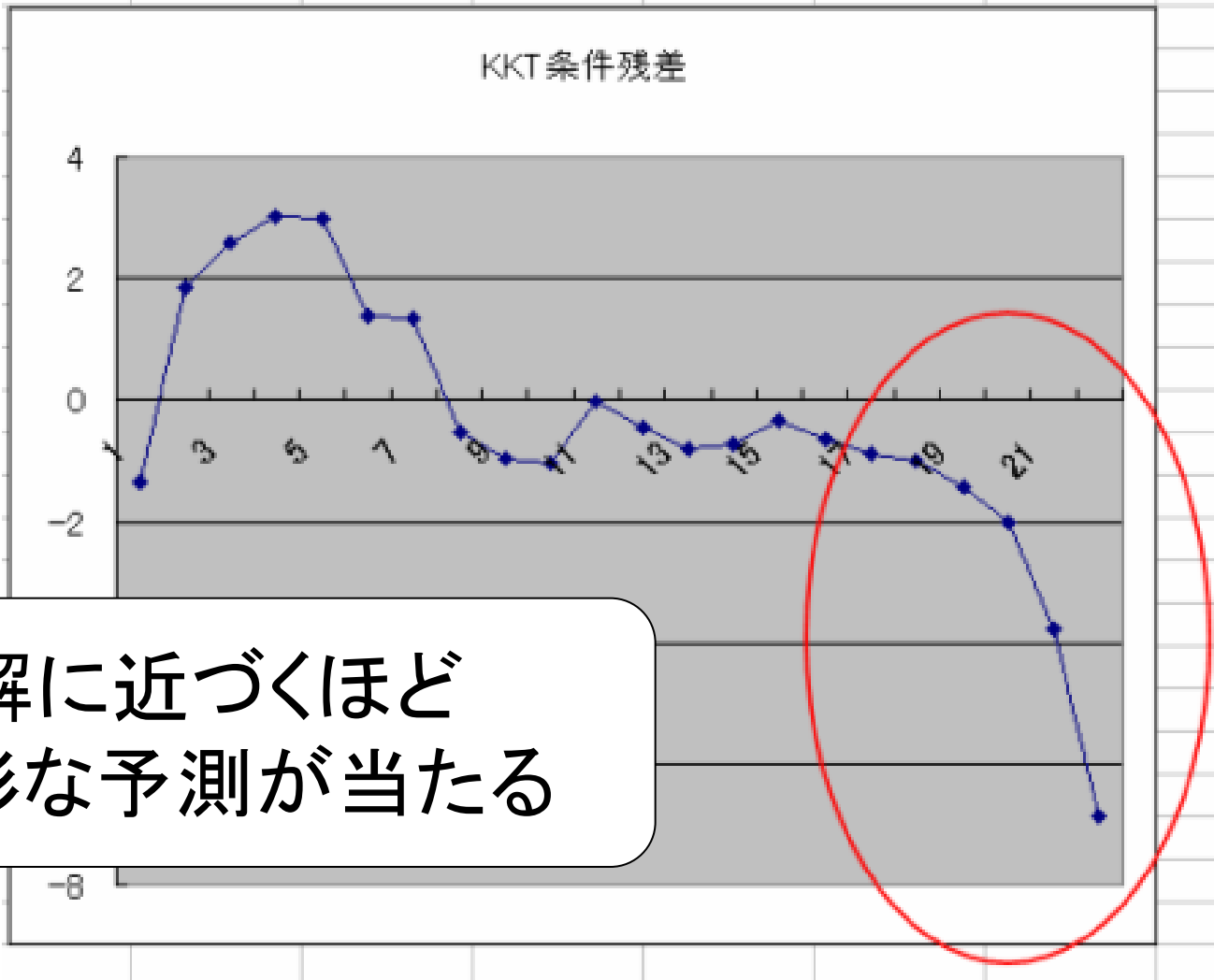
$$F(x, \mu) = f(x) - \mu \sum \log(x_i) + \rho \sum |g_i(x)|$$

のもとでの**大域的収束**

- 適当な μ の更新方法のもとでの
超一次収束

H. Yamashita, H. Yabe, and T. Tanabe, A globally and superlinearly convergent primal-dual interior point trust region method for large scale constrained optimization, Math. Program., 102 (2005), pp. 111.151.

超一次収束 (n=20000)



解に近づくほど
線形な予測が当たる

内点法の速度

問題名	種別	#変数	#制約式	反復	計算時間
資源計画	LP	76308	28129	44	37秒
大規模 PF	QP	6273	7874	48	3秒
微分方程式	NLP	20006	20000	10	421.5秒
原料購入計画	NLP	4166	2172	40	103秒

(計算機:Pentium1.5GHz+メモリ1G,ソフト:NUOPT)

LP/QP/NLPによらず高速

内点法の速度(線形)

問題名	種別	#変数	#制約式	#非零要素数	計算時間
LP1	LP	383927	201156	1053564	208.3秒
LP2	LP	161692	132021	1868865	170.5秒
LP3	LP	283250	139758	2408585	364.7秒
LP4	LP	308634	246078	902275	74.4秒

(計算機:Pentium1.5GHz+メモリ1G,ソフト:NUOPT)

LPならさらに高速

最適性条件(線形計画)

- KKT条件(最適性の必要条件)

$$Ax = b,$$

線形

$$c - A^t y - z = 0,$$

線形

$$Xz = 0,$$

唯一の非線形

$$x \geq 0, z \geq 0$$

高次オーダーの導入

$$(x + \Delta x)(z + \Delta z)$$

$$= xz + z\Delta x + x\Delta z + \Delta x\Delta z$$

$$\simeq xz + z\Delta x + x\Delta z = 0$$

線形計画問題では現在の主流

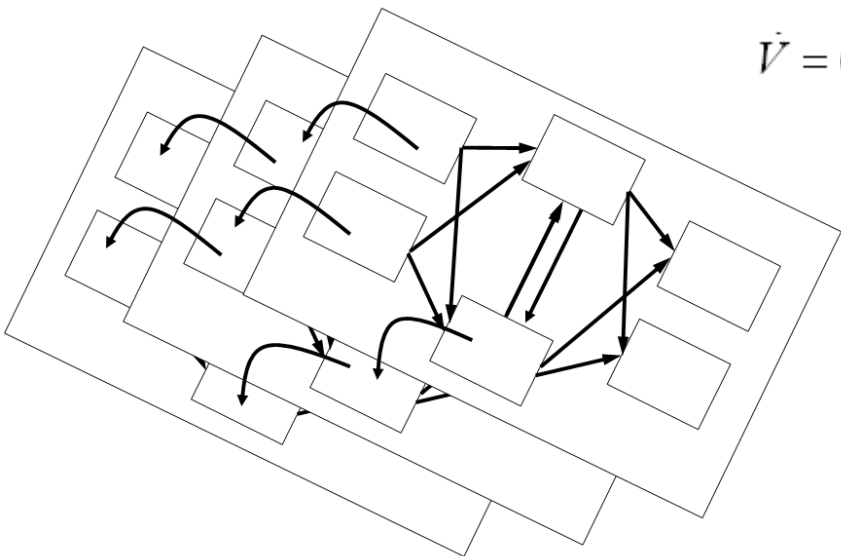
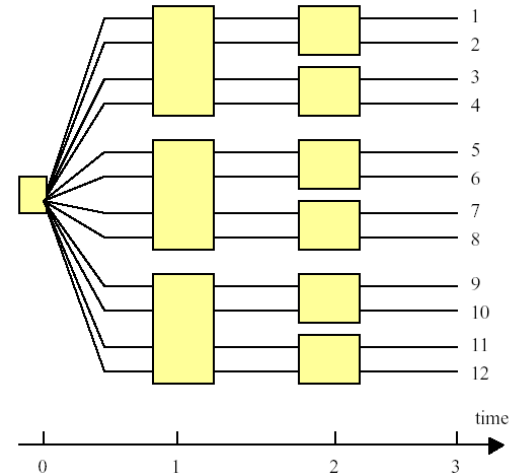
大規模性の源泉

- 時系列的要素
- 連続系の近似
- 確率計画

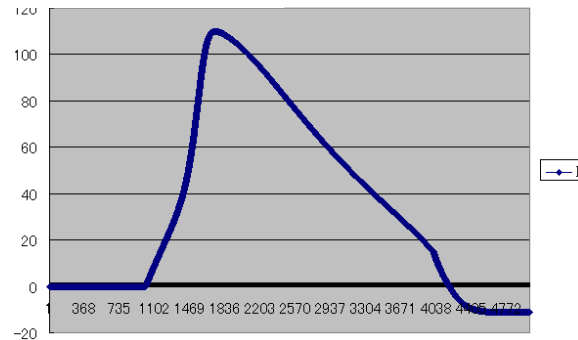
$$\dot{n} = \alpha_n \cdot (1 - n) + \beta_n \cdot n,$$

$$\dot{m} = \alpha_m \cdot (1 - m) + \beta_m \cdot m$$

$$\dot{h} = \alpha_h \cdot (1 - h) + \beta_h \cdot h$$



$$\dot{V} = (I - I_{Na} - I_K - I_L) / C_m$$



連続変数モデルの現状

- 数万変数でもほぼ**数分**で厳密解
(PentiumIV 1GHzクラス)
- 2Gバイトというメモリーの壁はある
(50~100万変数)

10年前に比べて約100倍の性能

内点法実装

- 対称不定値行列:

$$\nabla^2 f(x) - \sum_i y_i \nabla^2 g_i(x)$$

対角行列

$$\nabla g(x) \begin{bmatrix} G + D & -A^t \\ -A & 0 \end{bmatrix}$$

を係数行列とする一次方程式解法に大きく依存

行列解法 所要時間

- $n=10000, m=5000$ の convex QP
の計算時間:
 - case1 ($\text{nonz}(Q) = n$) 2秒(/5秒)
 - case2 ($\text{nonz}(Q) = 2n$) 43秒(/55秒)
 - case3 ($\text{nonz}(Q) = 2.5n$) 172秒(/228秒)

大規模問題では約8割以上

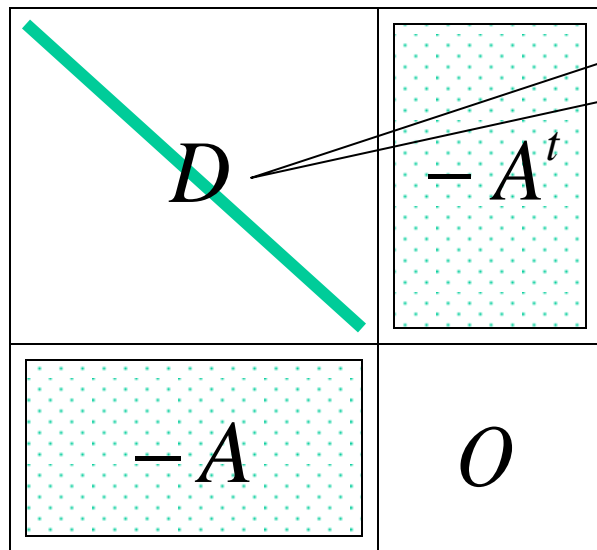
行列解法 技術的課題

- 特殊な構造・疎行列性の利用
非零要素は**1%以下**
- 行列要素の不変な部分の利用
- CPUパワーの効率利用
パイプライン・キャッシュ...
- 並列化
超大規模問題... 4G(32bit)の壁

特殊な構造の利用

LPの場合

- Normal Equation Form (Adler, Karmarkar'89~)



不定値行列

要素は正
反復中はここのみ変化



$$AD^{-1}A^t \equiv B$$

正定値行列

行列解法が容易

特殊な構造の利用

LPの場合

- 更新アルゴリズム

```
for(k) {  
  for(i,j) {  
    B[i,j] += A[k,i] · A[k,j] · D[k,k]  
  }  
}
```

反復中不変

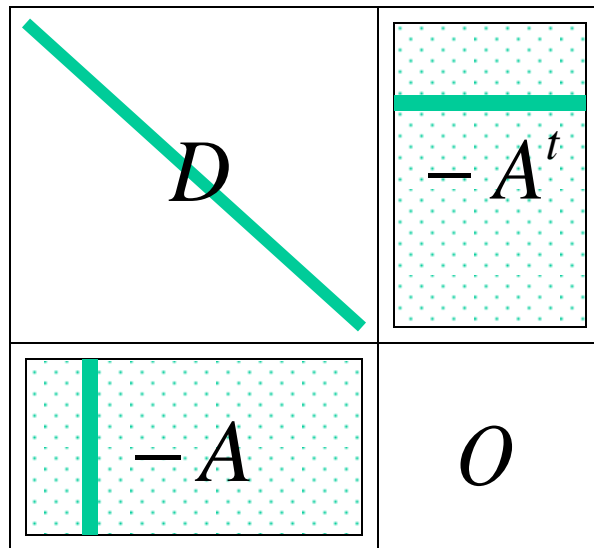


特殊データ構造

特殊な構造の利用

LPの場合

- Dense column の問題



疎行列



計算効率悪化

A solid orange square representing a dense matrix, labeled $ADA^t \equiv B$.

密行列

特殊な構造の利用 LPの場合

- Dense Column 対策の効果
($n=1000, m=500, \text{convex QP}$)
 - Dense column 対策なし 19秒
 - Dense column 対策あり 1秒

特殊な構造の利用

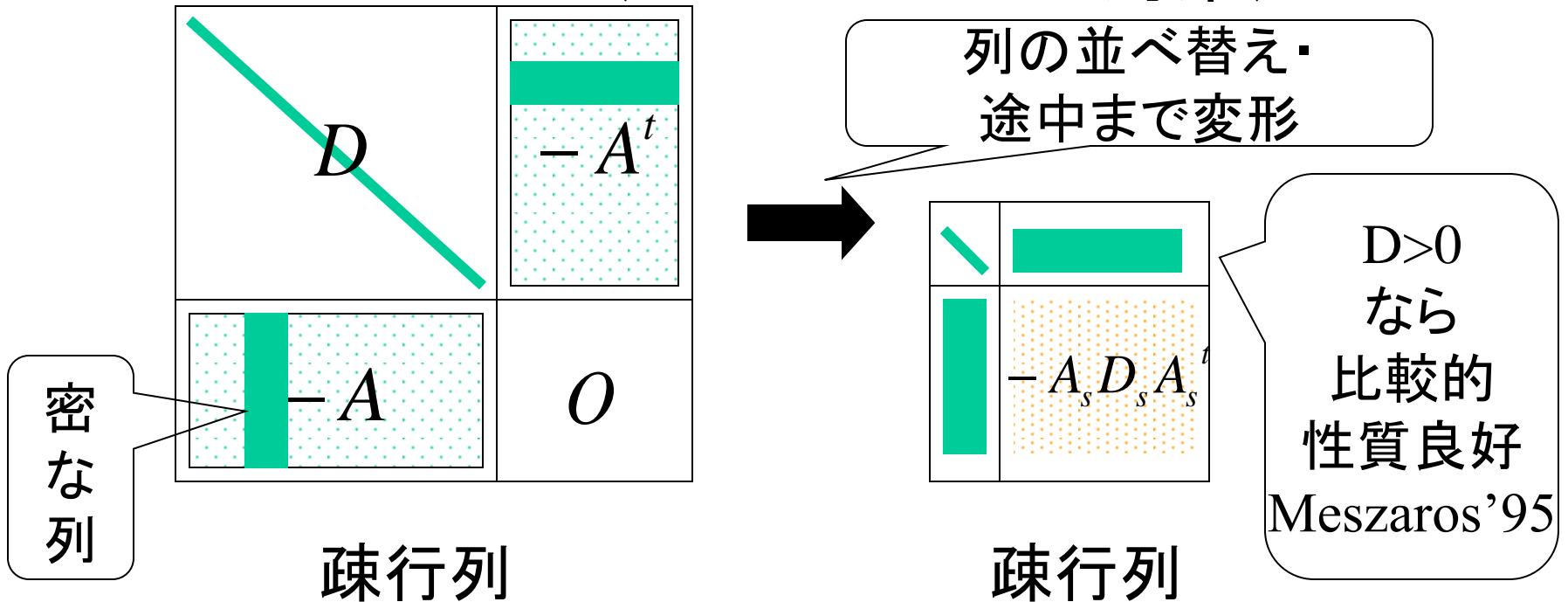
LPの場合

- Dense column 対策
 - Column 分割(Gonzio'94)
 - Schur Complementの利用 (Andersen'94)
 - Augumented System Approach(Maros,Meszaros'95)

特殊な構造の利用

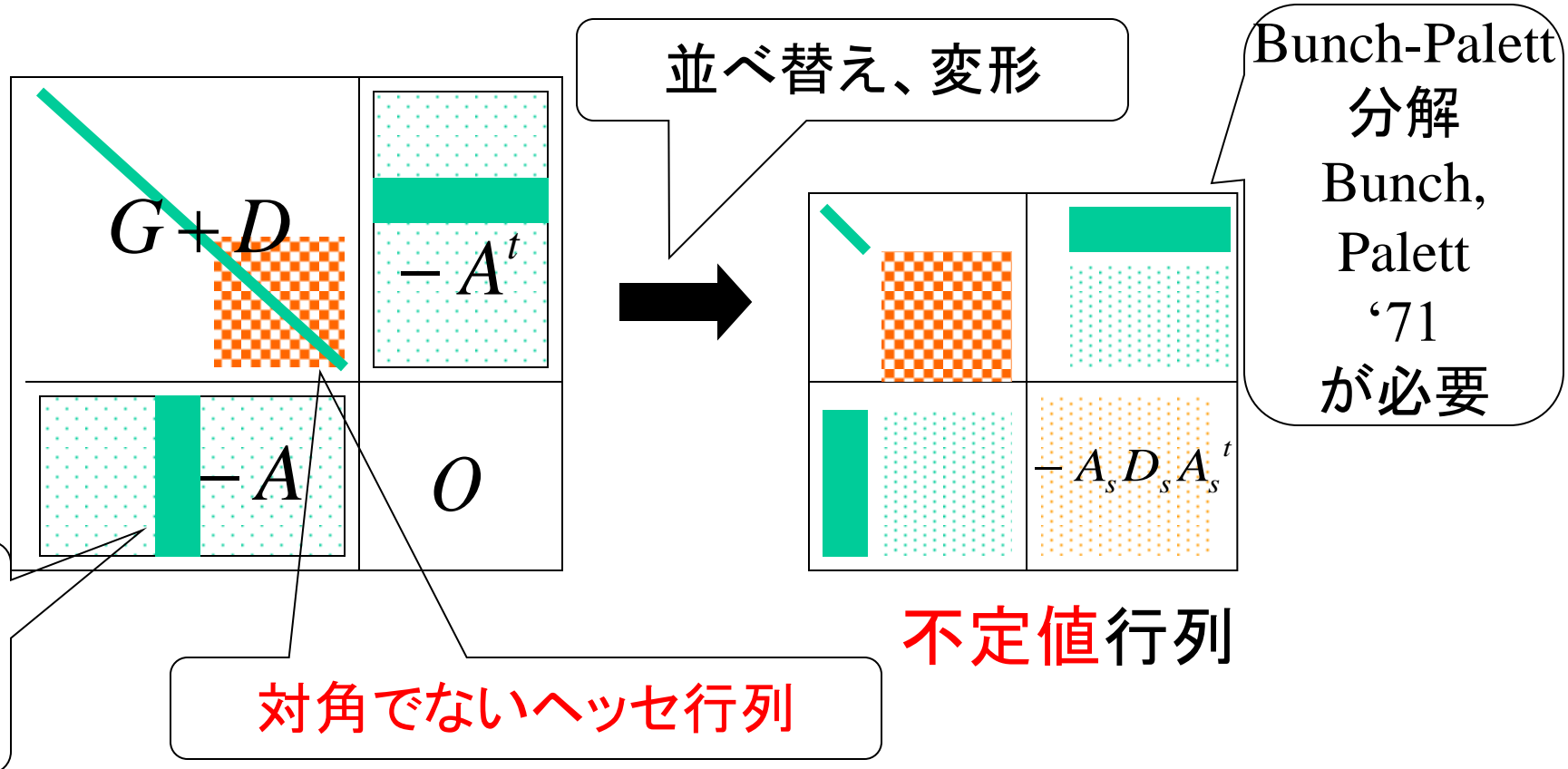
LPの場合

- Augmented System Approach
(Dense Column 対策)



特殊な構造の利用 NLPへの発展

- NLP用のAugmented System Approach

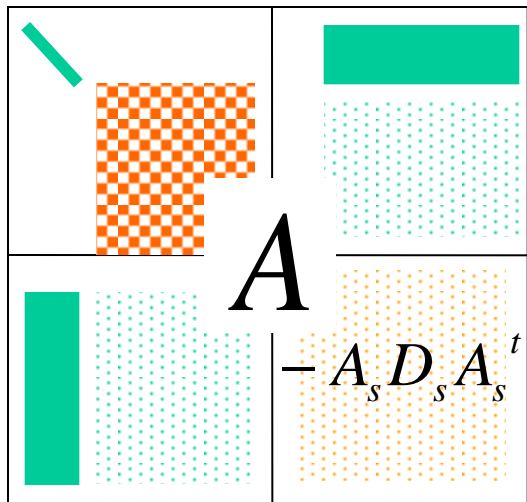


特殊な構造の利用

NLPへの発展

- 直接法による一次方程式求解

不定値行列



下三角行列

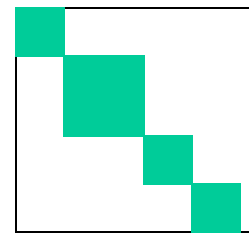
$$\Rightarrow LDL^t$$

前進消去・後退代入
 $O(n^2)$

$$x = L^{-t} D^{-1} L^{-1} b$$

分解 $O(n^3)$

ブロック対角



\neq 対角

CPUパワーの効率利用

行列解法手順

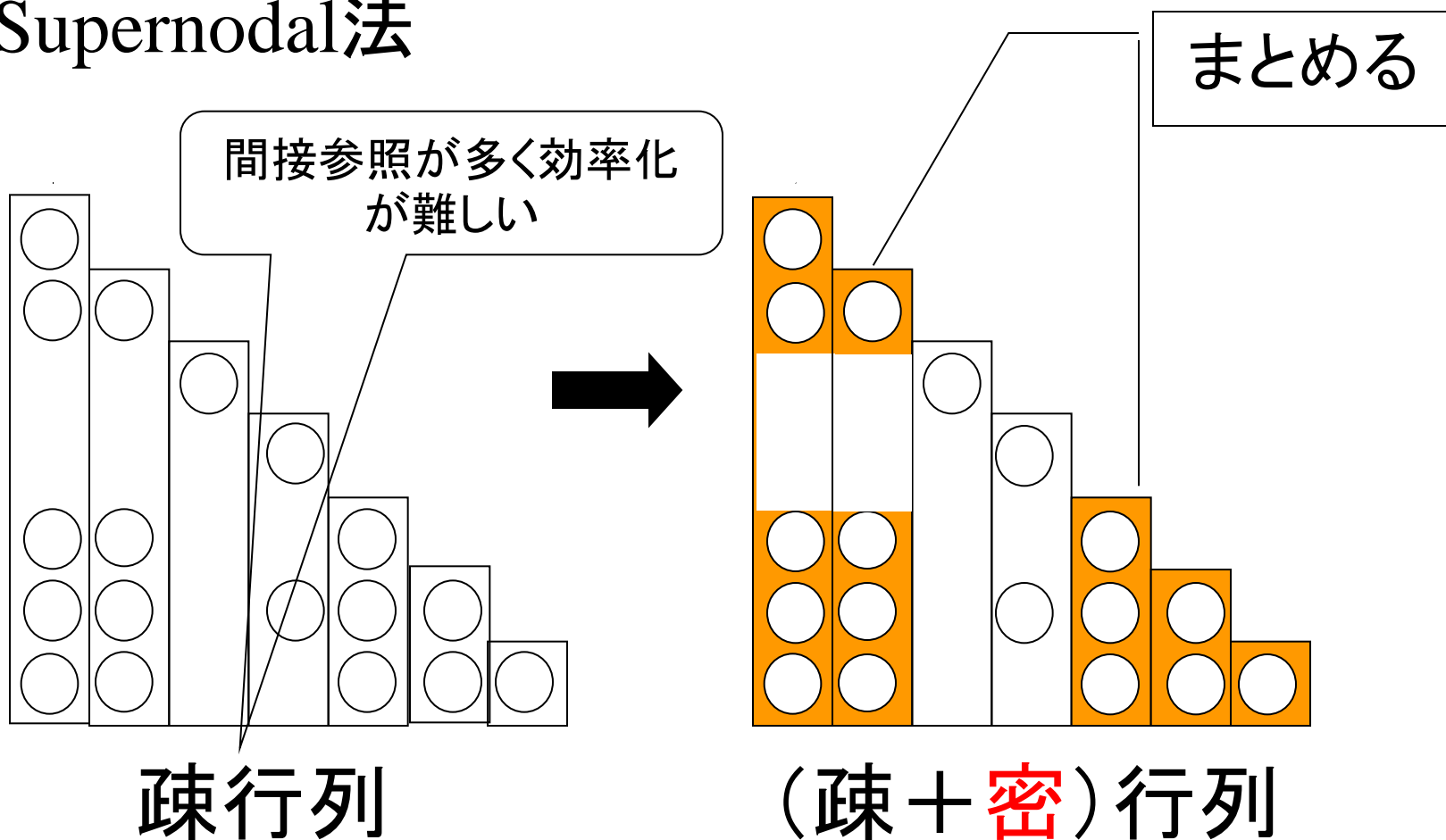
1. 不定値行列への変形
2. Fill-in減少オーダリング
3. bunch-palett+multifrontal 分解 (Duff'83)
 - ピボットを保存
 - 数値的安定性考慮 (2×2 ピボットの援用)
4. 一般化LDL^t分解
 - ピボット固定

回路・PDE解法より

速度にインパクト

CPUパワーの効率利用 密行列への帰着

- Supernodal法

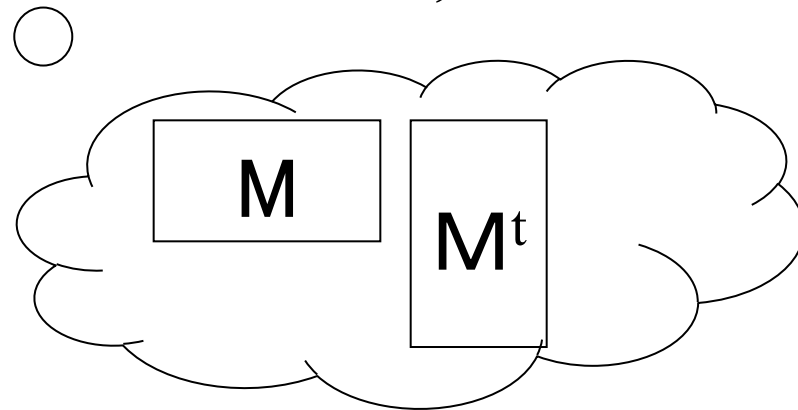


CPUパワーの効率利用

密行列操作ライブラリ

- ・ **密行列演算** (MM^t) の計算の性能
(M: 1200×1700 PentiumIV, 1.9GHz)

簡単なコード
特化した実装(**ATLAS**)



密行列演算には
特化した実装が利用可能

DSYMRK
(**BLAS3**)利用

CPUパワーの効率利用

密行列操作ライブラリ

- ・ **密行列**演算 (MM^t) の計算の性能

(M: 1200×1700 PentiumIV, 1.9GHz)

簡単なコード

23.7秒

特化した実装(**ATLAS**)

1.7秒

1Gflops超

Supernodal法
高速化の源泉

DSYMRK
(BLAS3)利用

CPUパワーの効率利用

LDL^t分解公式

$$A = \begin{bmatrix} D & B^t \\ B & C \end{bmatrix} = \begin{bmatrix} L_D & 0 \\ B(D_D L_D)^{-1} & I \end{bmatrix} \cdot \begin{bmatrix} D_D & 0 \\ 0 & C - BL_D^{-1}D_D^{-1}L_D^{-t}B^t \end{bmatrix}$$

A^+

$$\begin{bmatrix} L_D^t & (D_D L_D)^{-t} B^t \\ 0 & I \end{bmatrix}$$

(1ステップ)

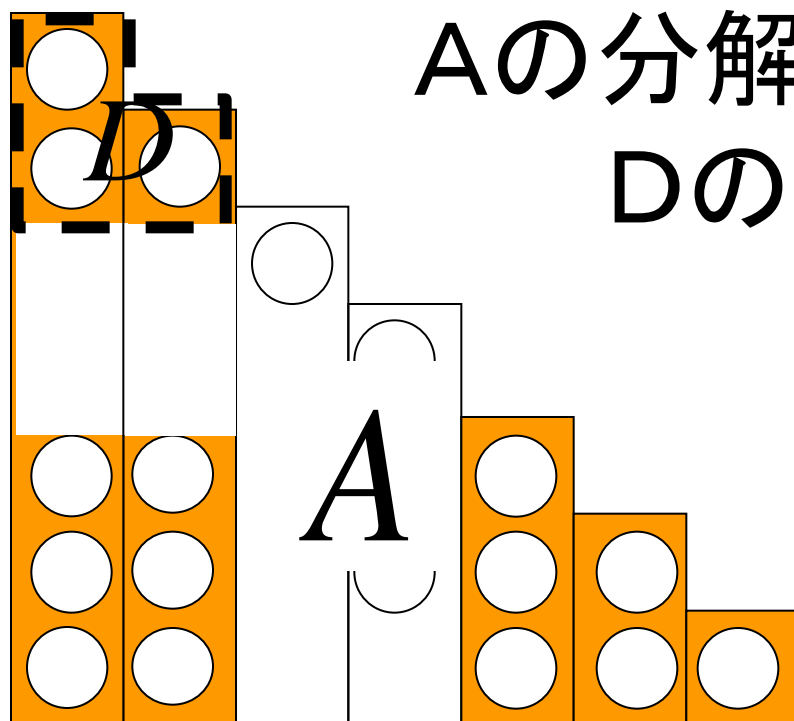
- ブロックピボット対応
- Super-nodal法対応

密MM^tの計算に帰着

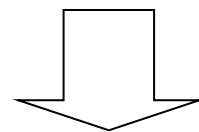
CPUパワーの効率利用

LDL^t分解公式

- 再帰的構造と積演算



Aの分解は
Dの分解に帰着



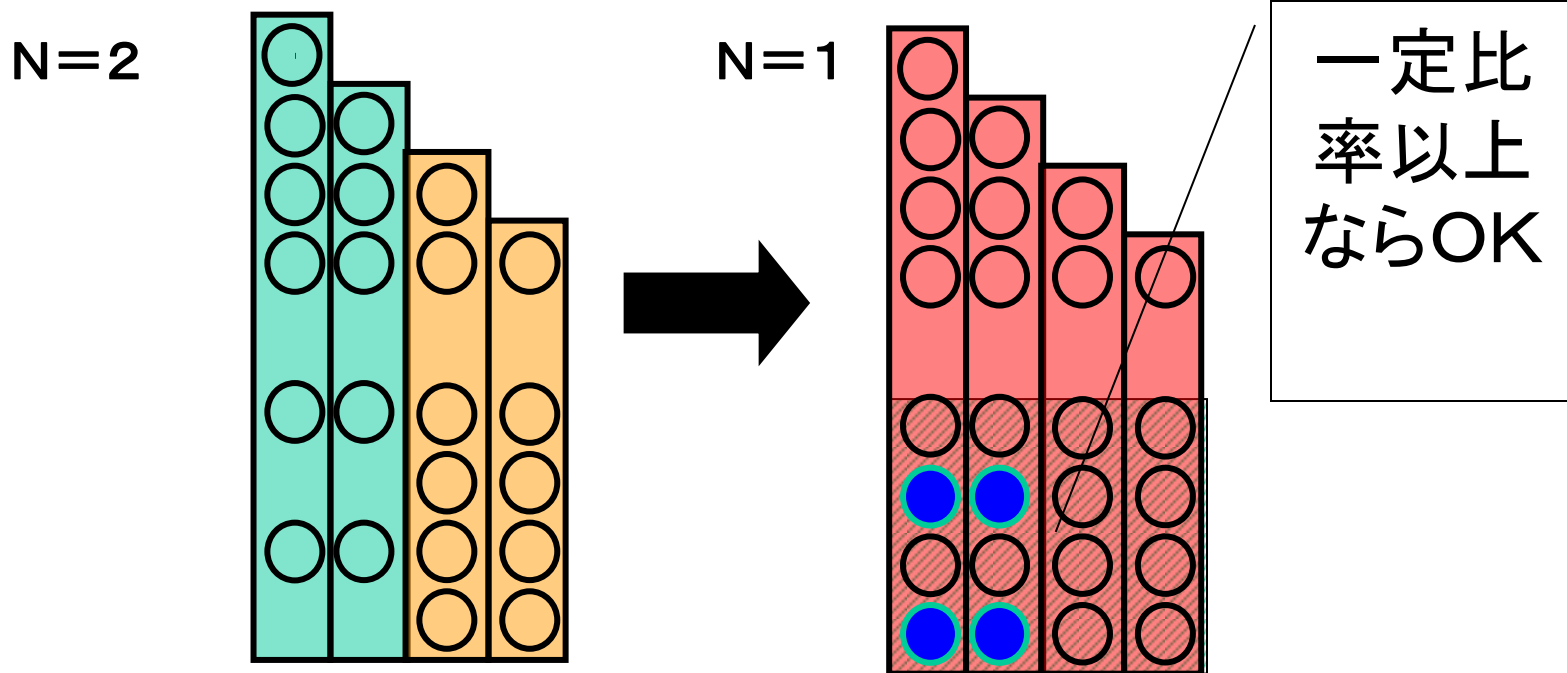
密MM^tの計算

CPUパワーの効率利用

密行列部分を増やす

- Supernode拡張

⇒少数の零要素は無視



CPUパワーの効率利用

密行列部分を増やす

- 実験結果(1)

	改良前	Super-node拡張	Aの分解にATLAS	Dの分解にATLAS
LP1	12.4秒	8.9秒	9.0秒	8.0秒
LP2	12.6秒	10.8秒	10.7秒	8.5秒
LP3	78.0秒	67.3秒	42.0秒	30.0秒

(使用計算機 : Pentium1.5GHz, メモリ 1 G)

CPUパワーの効率利用 密行列部分を増やす

- 実験結果(2)

問題種別	変数	制約	導入前	導入後
LP (ポートフォリオ)	12230	6072	150秒	75秒
NLP (非線形ネットワーク)	112769	44757	8828秒	2102秒

(使用計算機 : Pentium1.5GHz,メモリ 1G)

内点法の拡張

- 外点法
- 非線形半正定値計画問題

内点法 問題

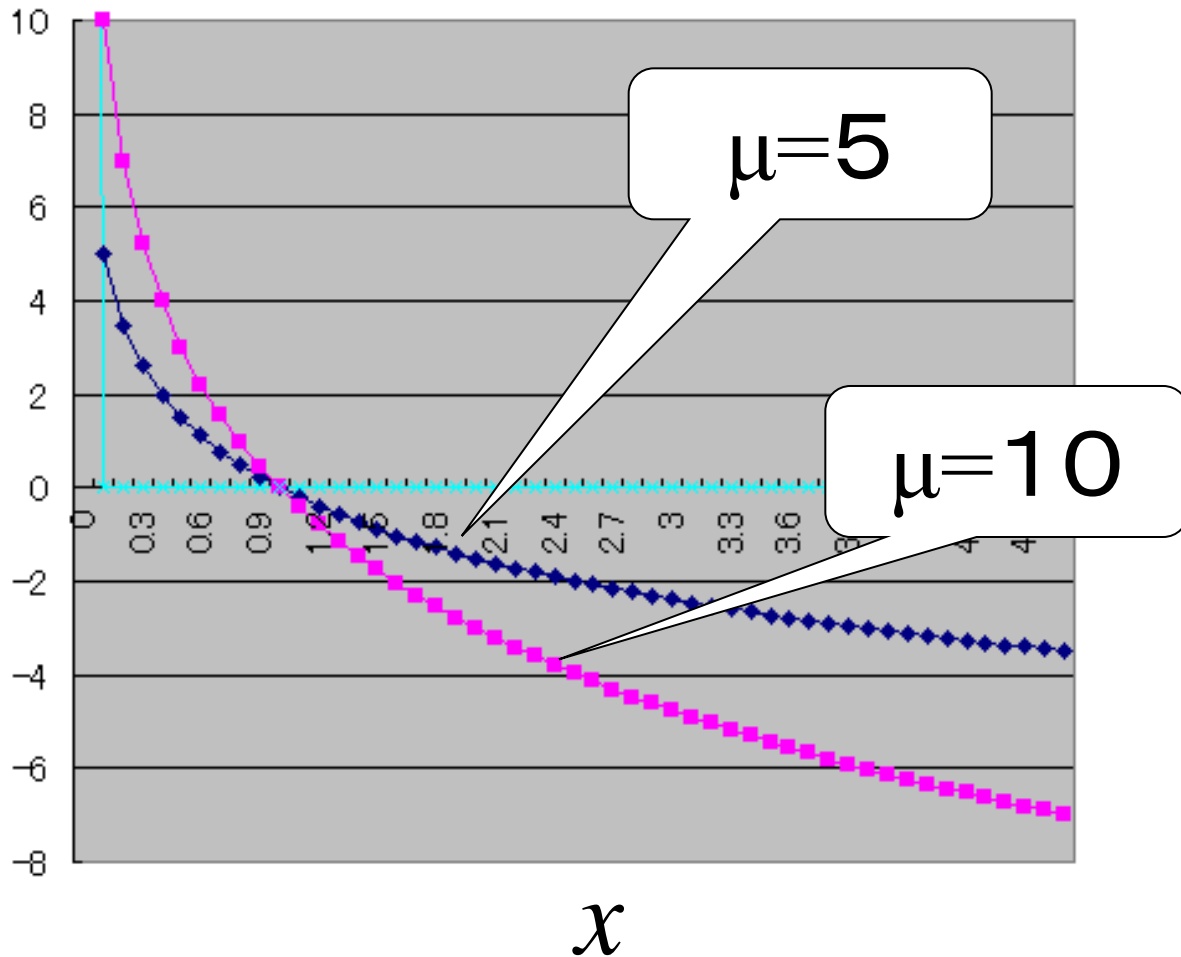
最小化 $f(x) - \mu \sum_i \log(x_i), \quad x \in \mathbf{R}^n,$

条件 $g(x) = 0,$

$x > 0$

$\mu \zeta$ $\mu \log x$

$\mu \log x$



内点法

KKT条件

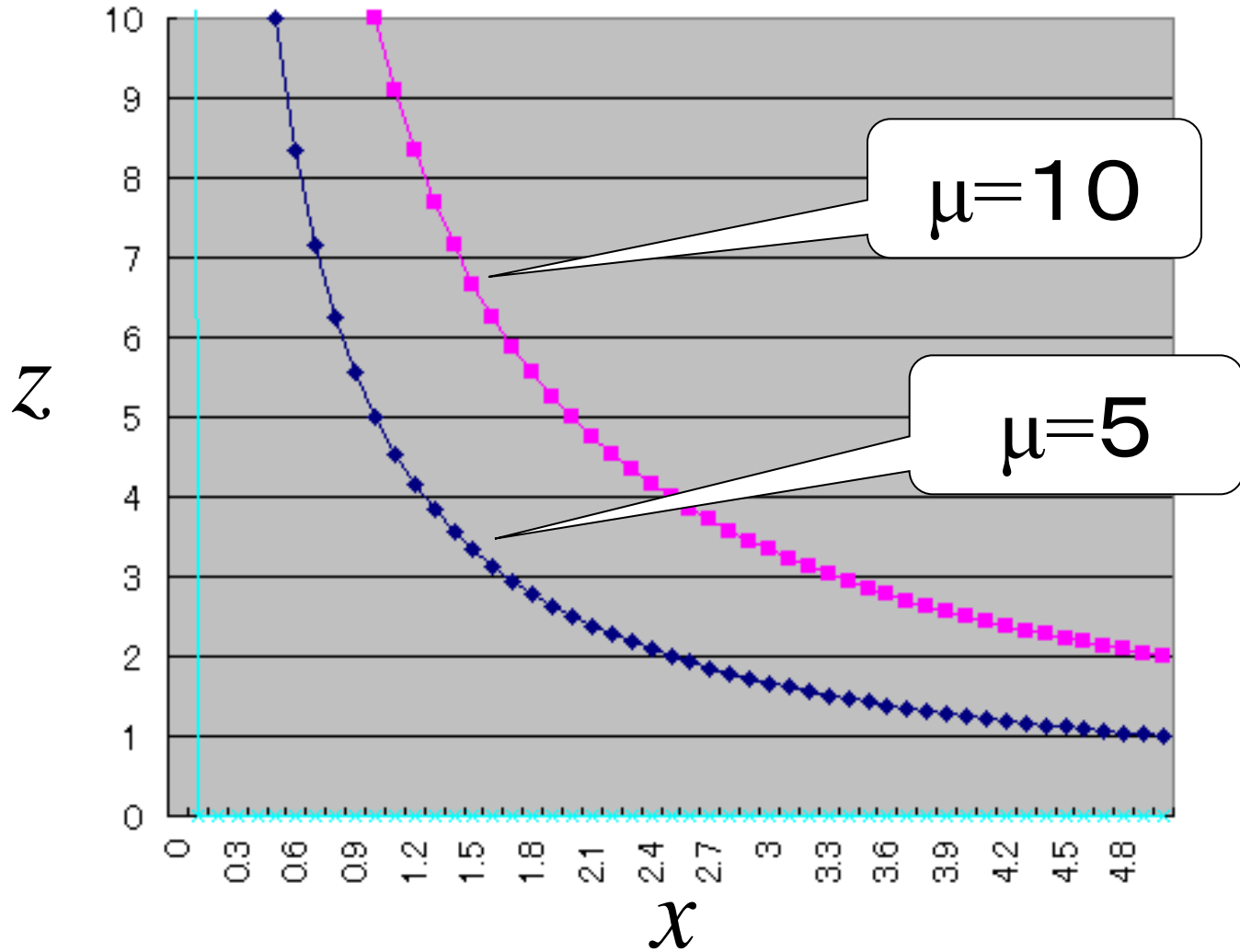
$$g(x) = 0,$$

$$\nabla f - A^t y - z = 0,$$

$$XZe = \underline{\mu e},$$

$$x > 0, \quad z > 0$$

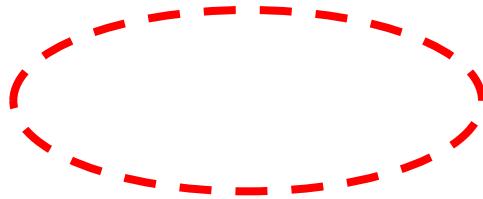
$$\mu \ll XZe = \mu e$$



外点法 問題

最小化 $f(x) + \rho \sum_i |x_i|_-$, $x \in \mathbf{R}^n$,

条件 $g(x) = 0$



$$|x|_- \equiv \max(-x, 0)$$

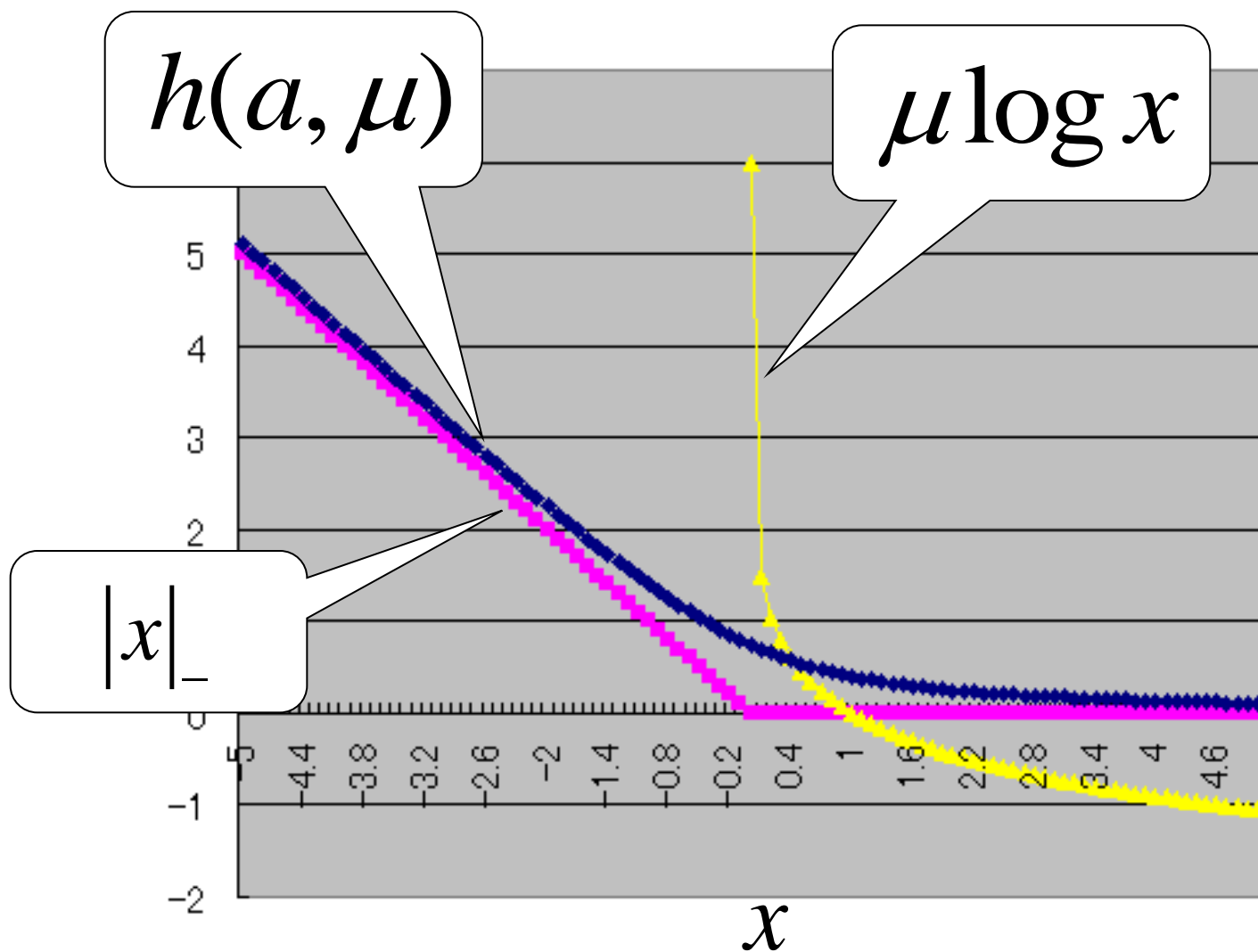
外点法 問題(近似)

最小化 $f(x) + \rho \sum_i h(x_i, \mu), \quad x \in \mathbf{R}^n,$

条件 $g(x) = 0$

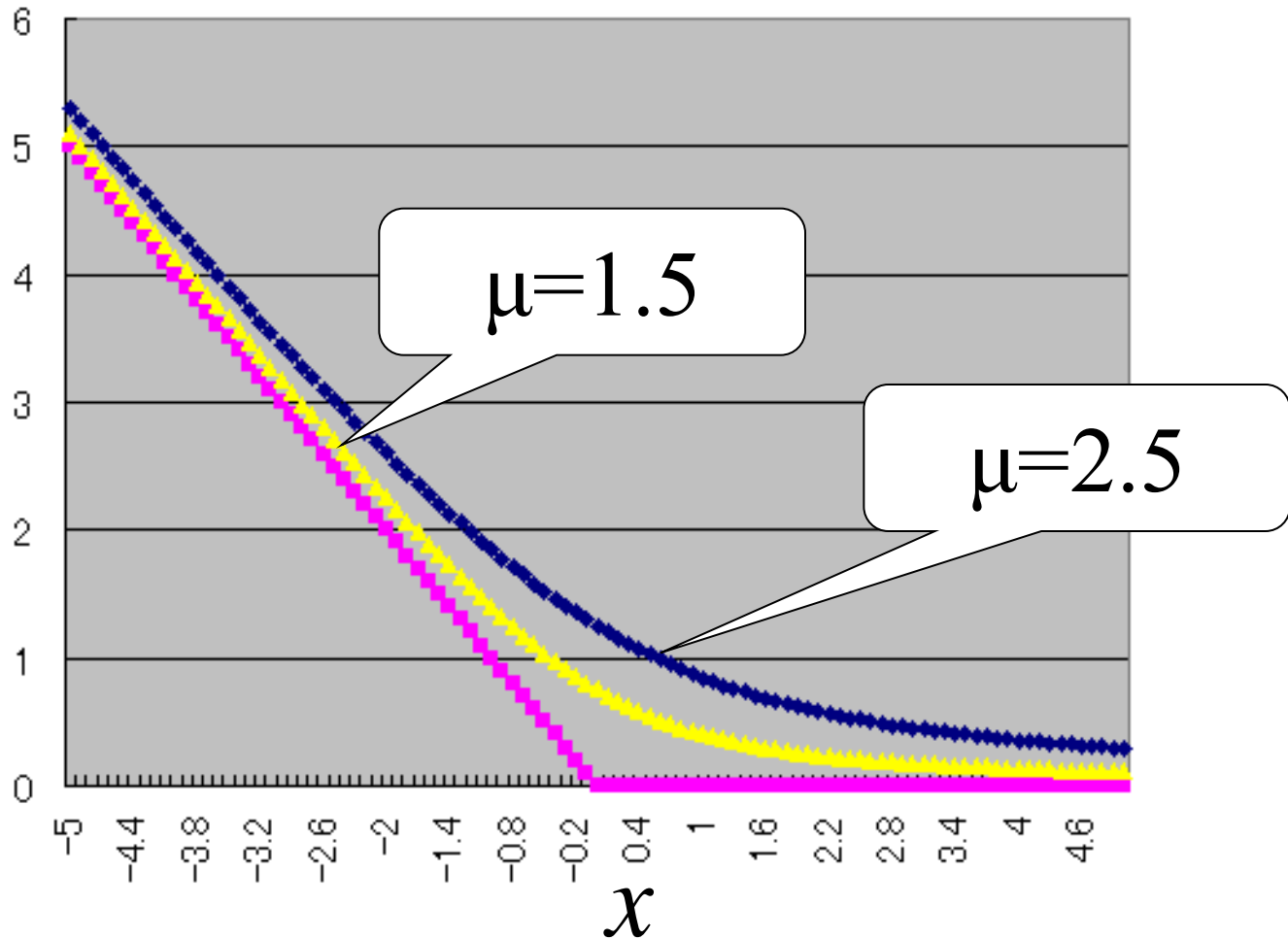
$$h(a, \mu) \equiv \frac{1}{2} (\sqrt{a^2 + \mu^2} - a)$$

バリア関数



$\mu \propto h(a, \mu)$

$h(a, \mu)$



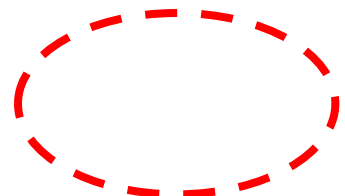
外点法

KKT条件

$$g(x) = 0,$$

$$\nabla f - A^t y - z = 0,$$

$$XZe + \underline{\rho |X|_-} e = 0,$$


$$\underline{\rho \geq z \geq 0}$$

$$|X|_- \equiv \text{diag}(|x_1|_-, \dots, |x_n|_-)$$

外点法

KKT条件(近似)

$$g(x) = 0,$$

$$\nabla f - A^t y - z = 0,$$

$$\underline{U(x, \mu)Ze} + \underline{\rho H(x, \mu)} = 0,$$

$$\rho \geq z \geq 0$$

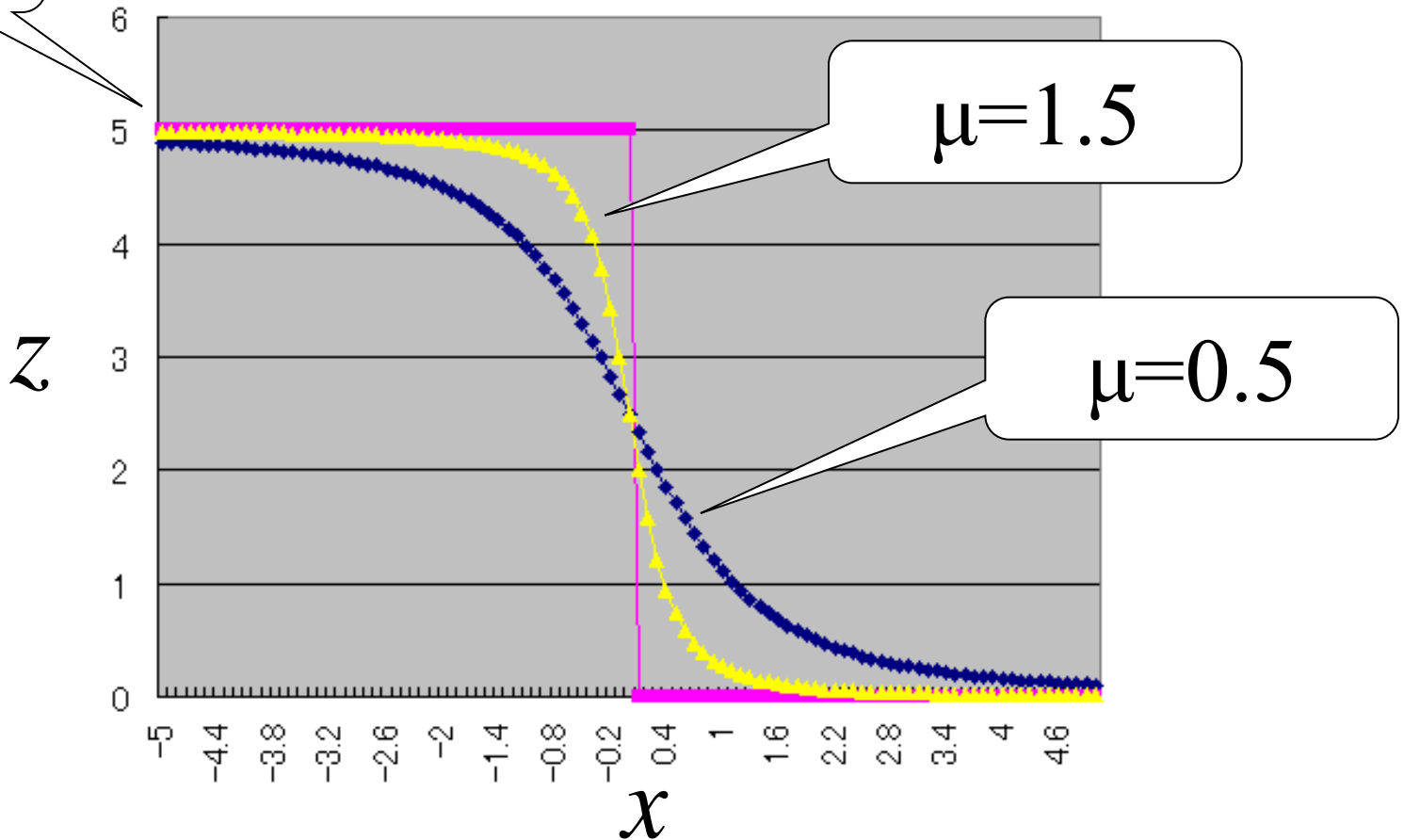
$$U(x, \mu) \equiv \text{diag}(\sqrt{x_1^2 + \mu^2}, \dots, \sqrt{x_n^2 + \mu^2})$$

$$H(x, \mu) \equiv \text{diag}(h(x_1, \mu), \dots, h(x_n, \mu))$$

$$XZe + \rho |X|_- = 0 \quad \rightsquigarrow$$

$$U(x, \mu)Ze + \rho H(x, \mu) = 0$$

ρ



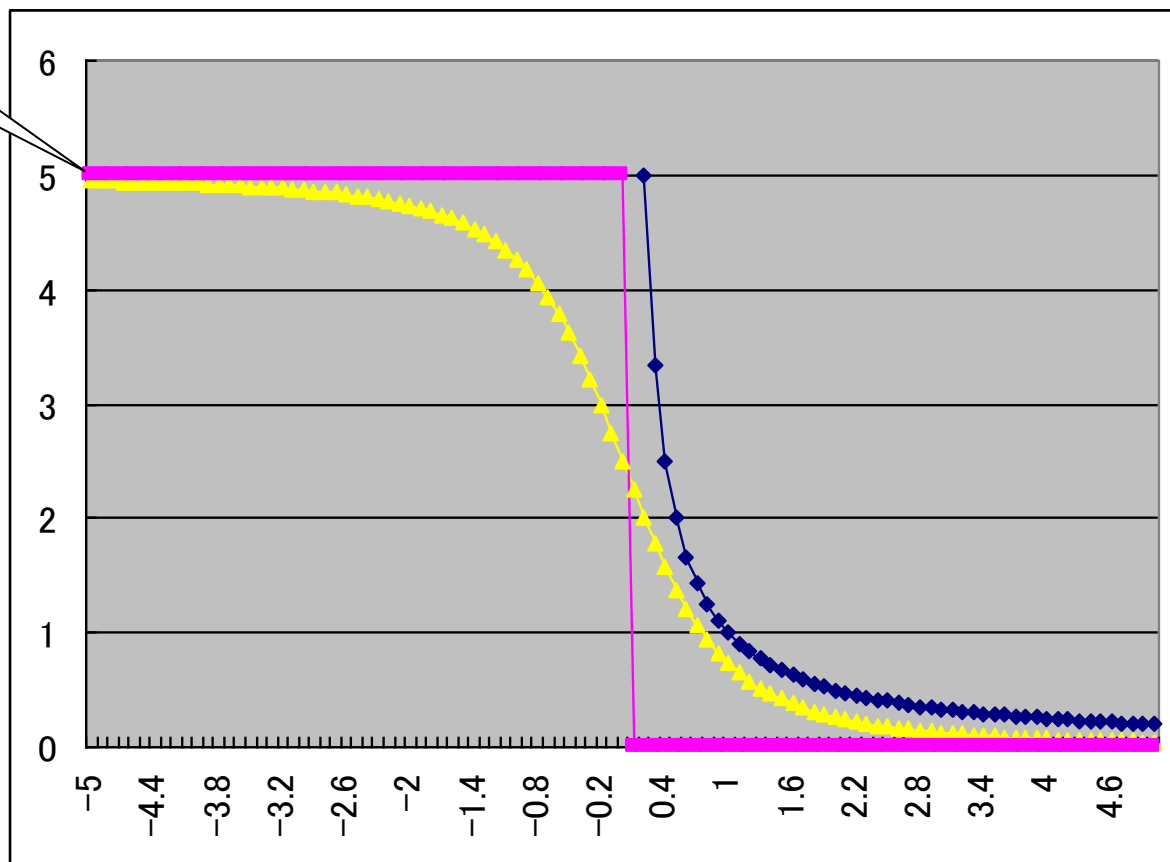
$$XZe = \mu$$

\sim

$$U(x, \mu)Ze + \rho H(x, \mu) = 0$$

ρ

Z



x

外点法のメリット

- 柔軟な初期点
- パラメトリック最適化への対応
- リスタートへの対応
 - 分枝限定法, ラグランジュ緩和法. . .

H. Yamashita and T. Tanabe, A primal-dual exterior point method for nonlinear optimization., SIAM J. Optim. 2010, Vol. 20, No. 6, pp. 3335-3363

パラメトリック最適化実験

マルコビッツモデルによるポートフォリオ最適化問題

(x : 組み入れ比率, R : 収益率行列, r : 平均収益率ベクトル) :

$$\text{最小化 } s^t s / T, s \in \mathbb{R}^n,$$

$$e^t x = 1, x \geq 0, x \in \mathbb{R}^n,$$

$$\text{条件 } Rx = s, R \in \mathbb{R}^{T \times n},$$

$$r^t x \geq r_p$$

(261変数, 37制約)

において, 期待収益率下限 r_p を 2.0% から 4.0% まで 0.2% 刻みで変化させたパラメトリック最適化を試みた. 以下に反復回数結果をあげる.

r_p	2.0	2.2	2.4	2.6	2.8	3.0	3.2	3.4	3.6	3.8	4.0
内点法	28	22	22	22	21	23	21	23	24	24	25
外点法	24	3	3	3	3	8	4	8	9	8	4

内点法の拡張

- 外点法
- 非線形半正定値計画問題

非線形半正定値計画問題(NLSDP)

$$\begin{aligned} & \text{最小化 } f(x), \quad x \in \mathbf{R}^n, \\ & \text{条 件 } g(x) = 0, \quad X \equiv \mathcal{A}x - B \succeq 0 \end{aligned}$$

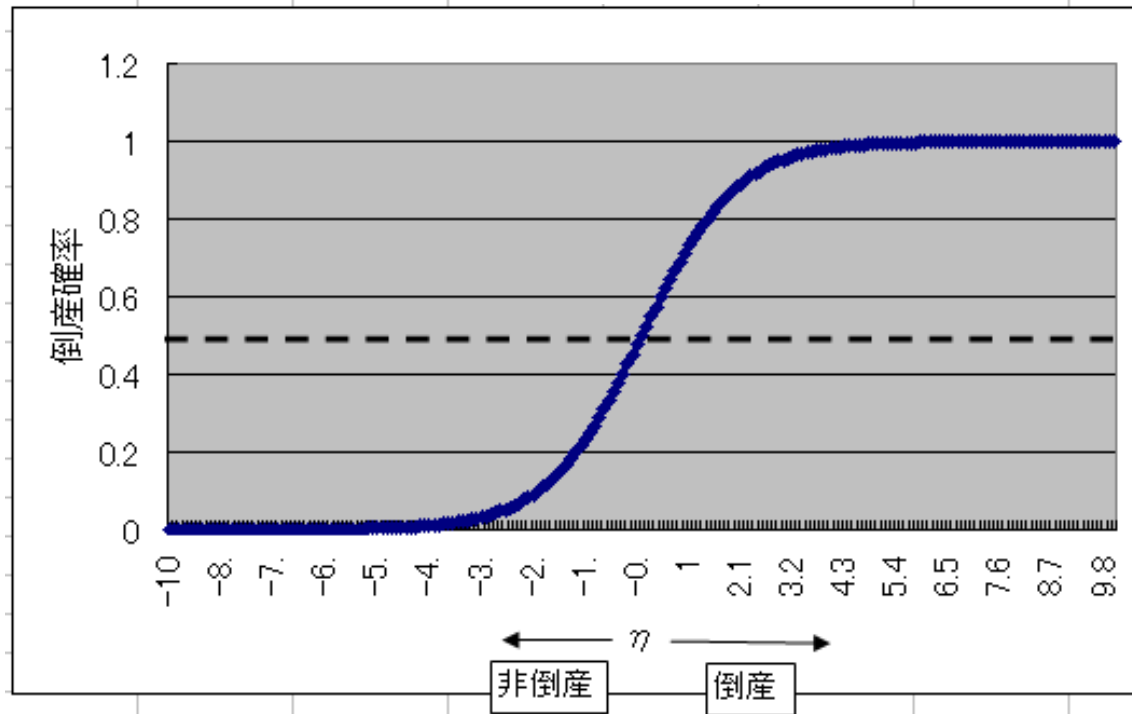
$X \succeq 0$ は半正定値行列を意味する

$\mathcal{A}x = \sum_{i=1}^n x_i A_i$ で, $A_i \in \mathbf{SR}^{p \times p}, i = 1, \dots, p, B \in \mathbf{SR}^{p \times p}$ は
与えられた対称行列

以下では $X, Z \in \mathbf{SR}^{p \times p}$ に対して内積を $\langle X, Z \rangle = \text{tr}(XZ)$ と定義する. このとき $\langle \mathcal{A}x, Z \rangle = x^t(\mathcal{A}^*Z)$ となる. ただし, \mathcal{A}^*Z の第 i 成分は $(\mathcal{A}^*Z)_i = \text{tr}(A_i Z)$ で定義される.

判別分析

- ロジスティクス関数 $\Phi(\eta)$



η の値によって倒産確率が決定

η : 財務諸表の値の関数

- 線形なモデル

$$\begin{aligned}\eta &= a_0 + a_1 X_1 + a_2 X_2 + a_3 X_3 + a_4 X_4 \\ &\equiv a_0 + \mathbf{a}^t \mathbf{X}\end{aligned}$$

X_1, X_2, X_3, X_4 ($\equiv \mathbf{X}$) 財務諸表の値

a_0, a_1, a_2, a_3, a_4 ($\equiv \mathbf{a}$) パラメータ

η : 財務諸表の値の関数

- 二次モデル

$$\eta = a_0 + \mathbf{a}^t \mathbf{X} + \mathbf{X}^t \mathbf{Q} \mathbf{X}$$

$$\begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ q_{41} & q_{42} & q_{43} & q_{44} \end{bmatrix} \equiv \mathbf{Q}$$

表現力
増大

パラメータ
(二次項)

パラメータ推定のための 数理計画モデル

- 変数

$$a_0, \mathbf{a}, \mathbf{Q} \text{ (パラメータ)}$$

- 目的関数

倒産データについては $\frac{e^\eta}{1+e^\eta}$ が 1 に近い

存続データについては $\frac{e^\eta}{1+e^\eta}$ が 0 に近い

$$\eta = a_0 + \mathbf{a}^t \mathbf{X} + \mathbf{X}^t \mathbf{Q} \mathbf{X}$$

パラメータ推定のための 数理計画モデル

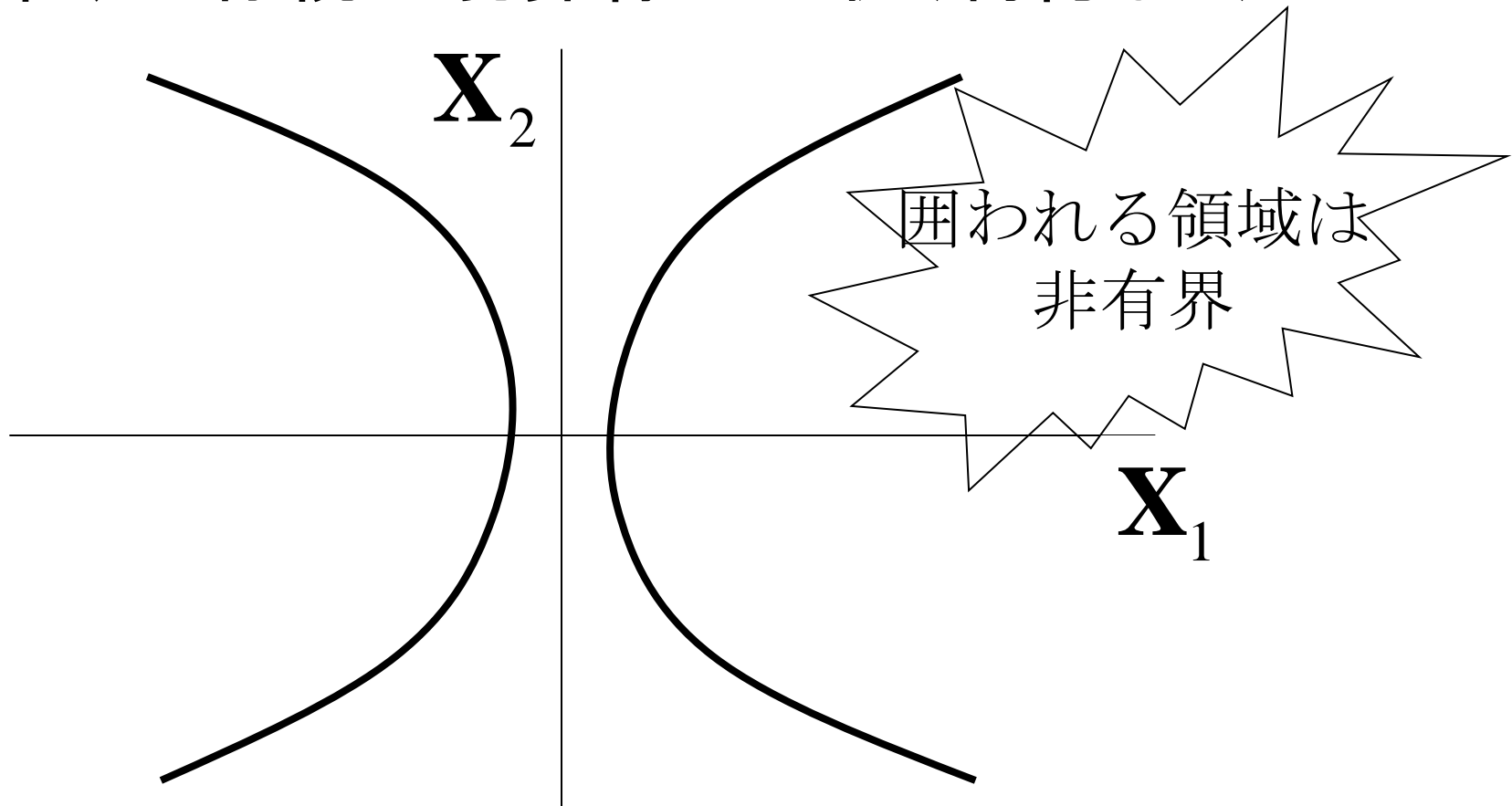
- 制約

Qが半正定値

予測力
向上

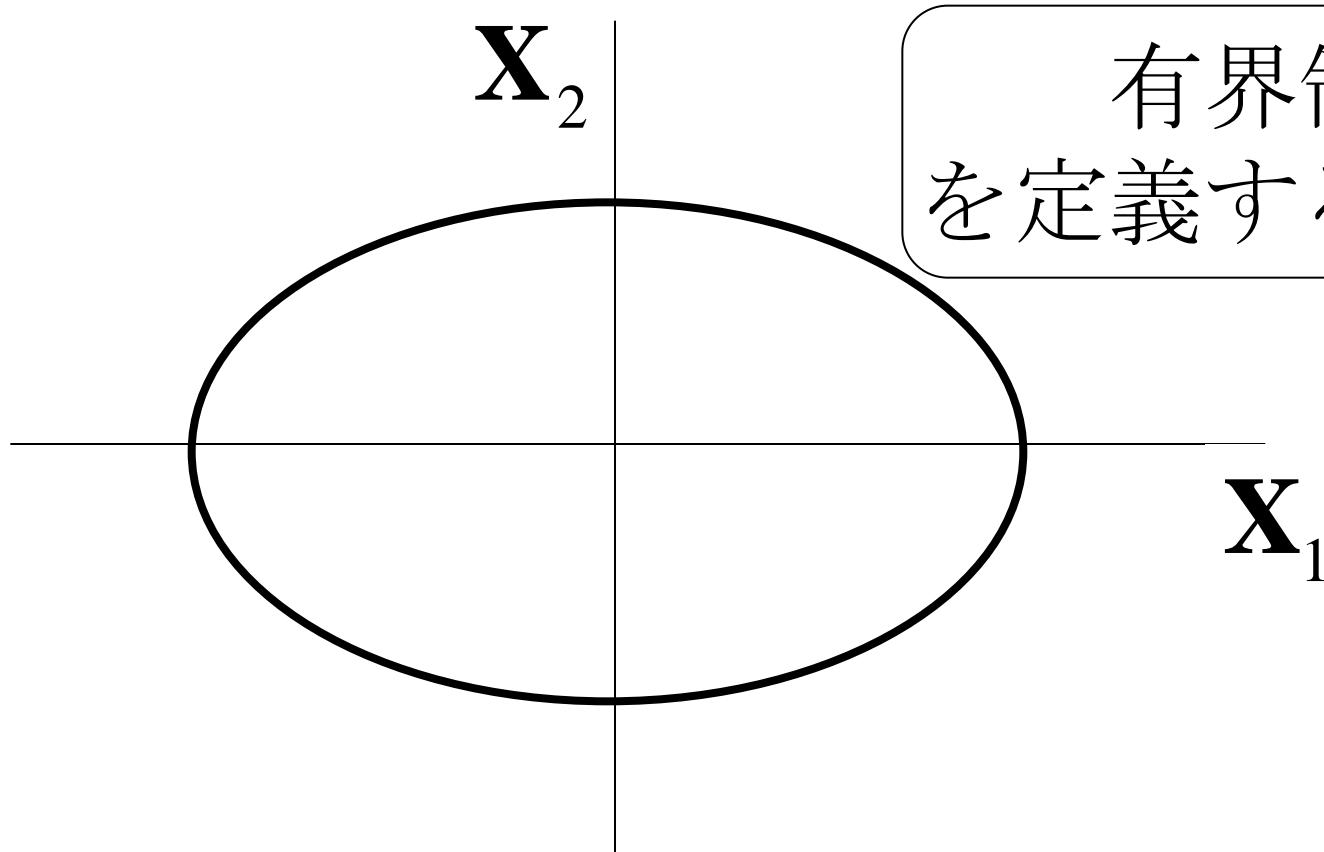
半正定値制約の意義

- 倒産・存続の境界線の形状(制約なし)



半正定値制約の意義

- 倒産・存続の境界線の形状(制約あり)



有界領域
を定義する可能性

半正定値の制約

- 内点法のみを利用するケース
 1. 内点法起動
 2. 半正定値性のチェック(OKなら終了)
 3. カットの追加, 1へ



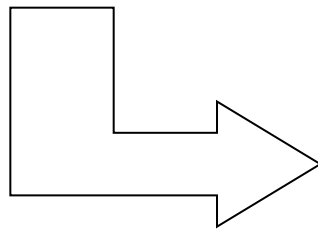
ループ (~ 30回)

プログラム高速化(2)

- SDP制約のアルゴリズムへの取り込み
 - 非線形SDP解法エンジン(SDP用内点法)
(数理システムオリジナル)

1. 内点法起動
2. 半正定値性のチェック
(OKなら終了)
3. カットの追加, 1へ

ループを解消



1. SDP用内点法
起動, 終了

半正定値計画法用 内点法の適用結果

- カット追加＋内点法と比べて

5～10倍の実行速度