九州大学 工学部地球環境工学科
船舶海洋システム工学コース

システム設計工学 （担当：木村）

(4) 連続関数の最適化（勾配法）

場所： 船1講義室

授業の資料等は

http://sysplan.nams.kyushu-u.ac.jp/gen/index.html

【最適化とは？】

「評価」を適切に定義していない例：
工場において、生産コストを最小化する運転方法
→工場を閉鎖して生産をストップ

「最も良い状態になるように、手段や方法を考えて行動すること」

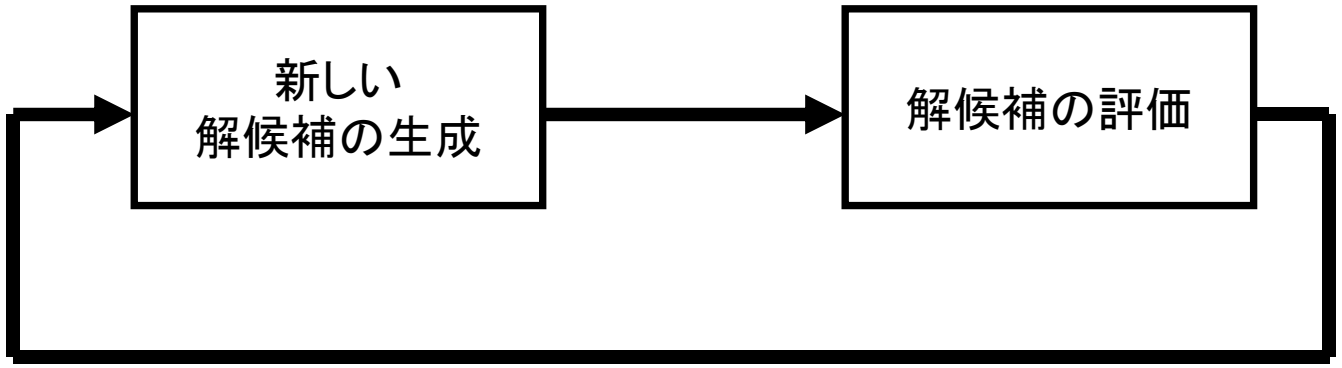
解候補ーーー評価値

例） 製品のデザインーーー売上げ高
船の形状ーーー推進抵抗

最適化以前に、その目的である「評価」を定義することが重要

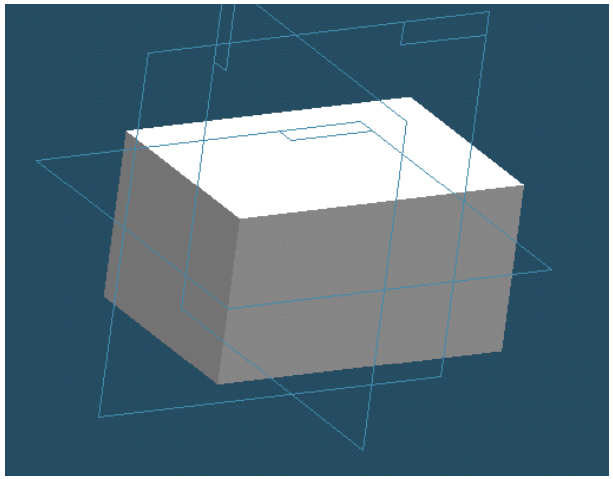
モデル化に関連

一般的な最適化のプロセス：



以前の解候補の評価値を参考にして、より有望な解候補を生成することで解候補を改善していく

【例：箱の寸法デザイン問題】



●商品としてよく売れるような箱の寸法（比率）をデザインする
　→　目的ははっきりしているが、
これだけではモデル化も評価値の定義もできていない

【問題のモデル化と評価の定義】

消費者は、商品に「黄金比」に近い長方形が含まれているものを好む傾向

黄金比　$= 1 : \dfrac{1+\sqrt{5}}{2}$

$$\mathrm{Cost} = \sum_{3\text{面}} \left(\text{黄金比} - (\text{長辺} / \text{短辺})\right)^2 \times \text{面積}$$

Cost が最小になるデザインが最もよく売れる

このモデルが現実と
かけ離れていては意味がない

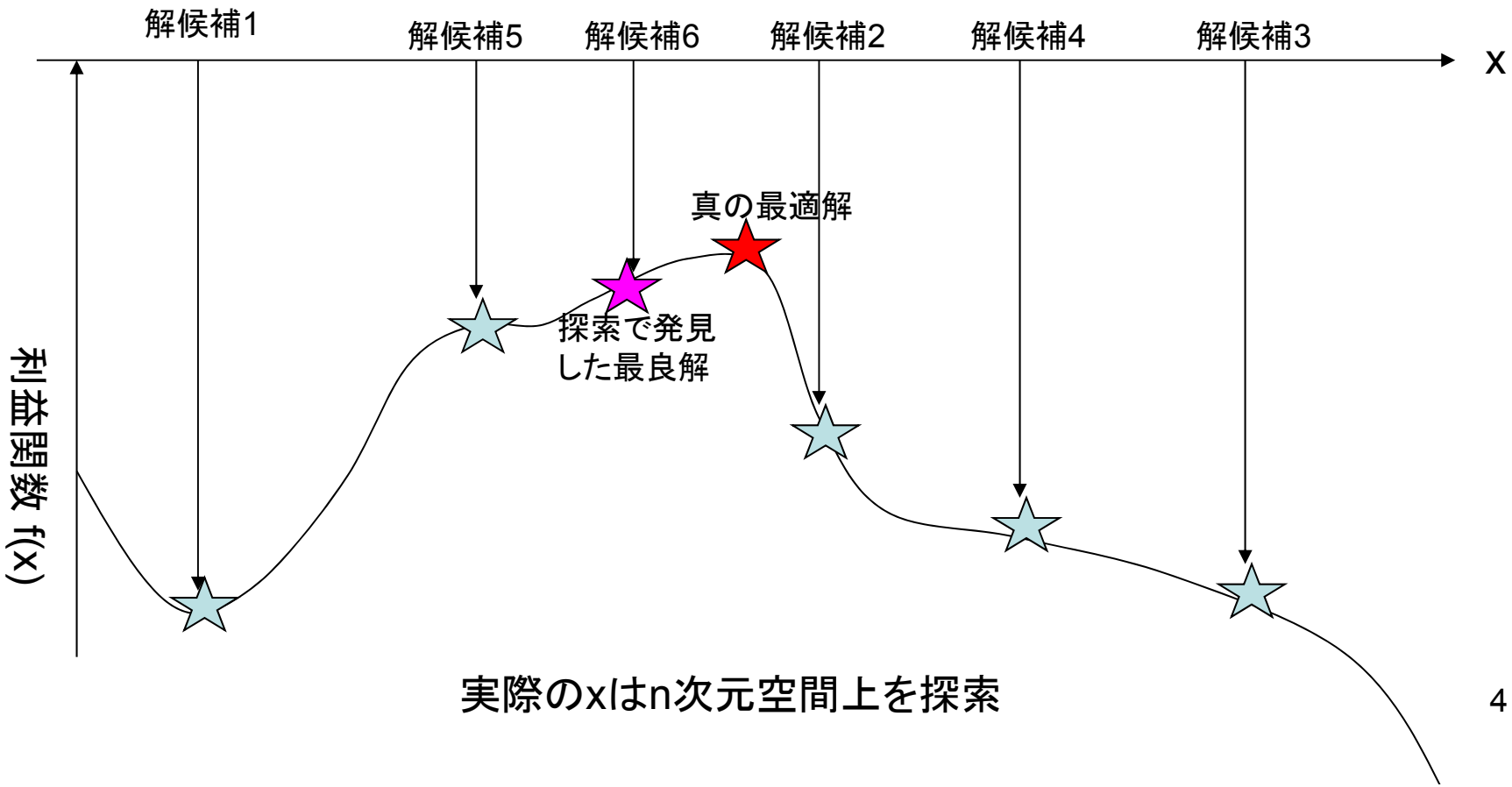
【最適化】

箱の寸法の比率を　X:Y:1　ただし　X ＜ 1　Y ＜ 1　とおく
Cost を最小化する X および Y を求める

この例題の場合はCostの式から解析的に解けるが、例えばデータベースに基づいた計算などが加わると解析的に求められないので、解候補を次々代入して改善するしかない

【最適化とは？】　　1）連続関数最適化問題

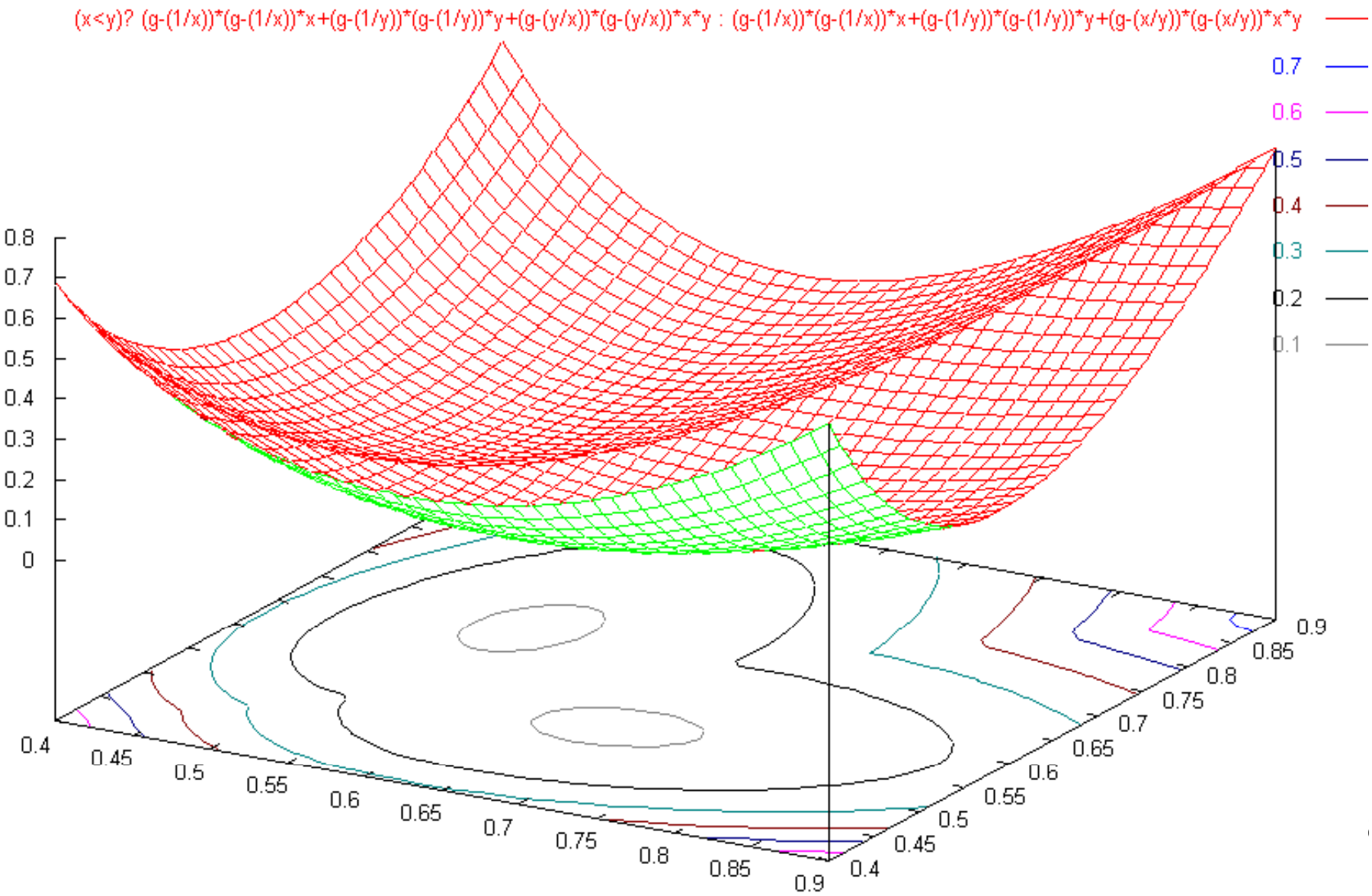
例：　石油を汲み上げるための油井を掘る
　　油を含む地層が最も地上に近い部分を掘ると利益が最大になる
　　予算の都合で、試掘の回数が制限される
→　限られた試掘回数（6回）の範囲内で、最も地上に近い油層を探索する



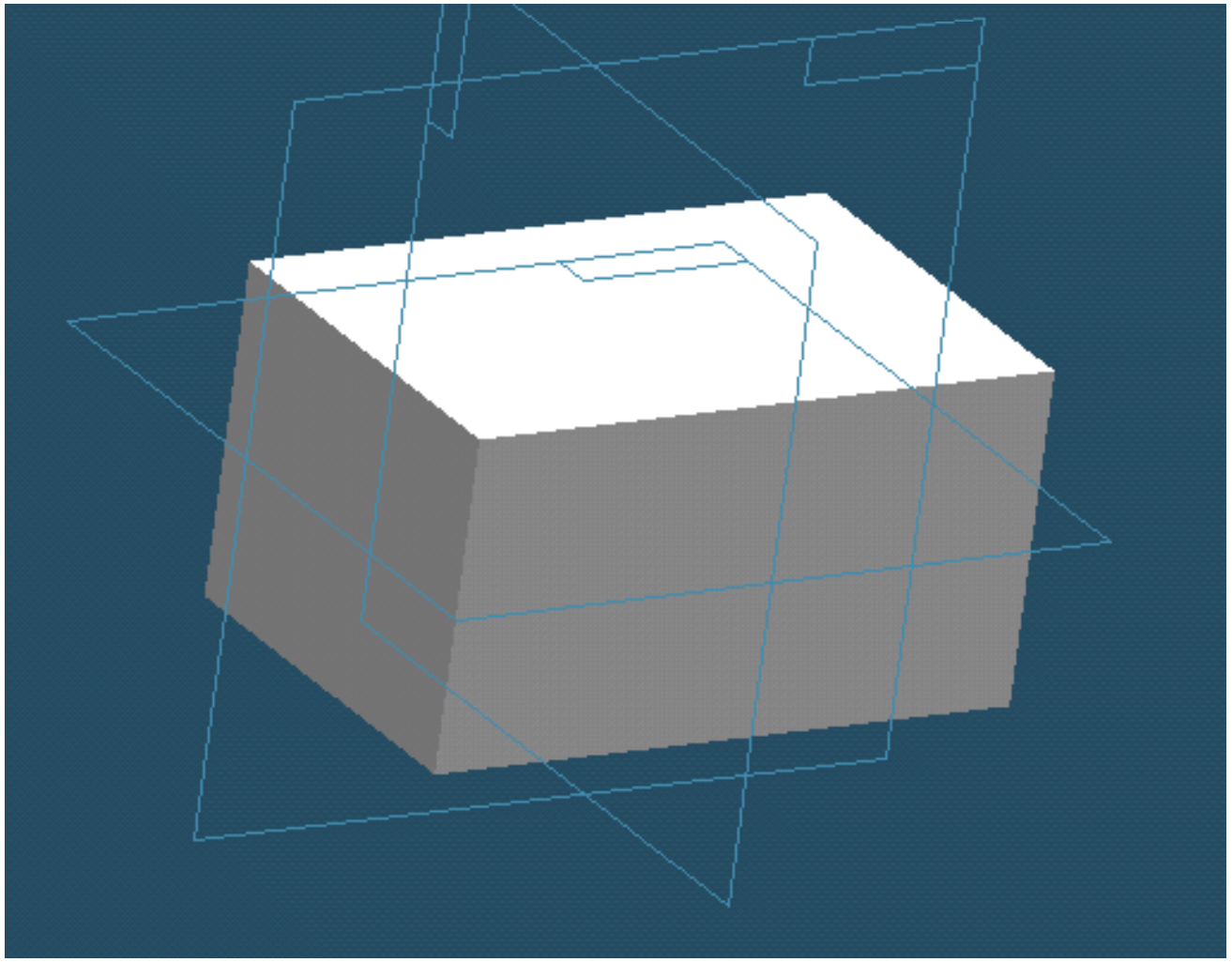
実際のxはn次元空間上を探索

箱の寸法デザイン問題におけるコスト関数の景観

最小値をとるパラメータは2箇所に存在

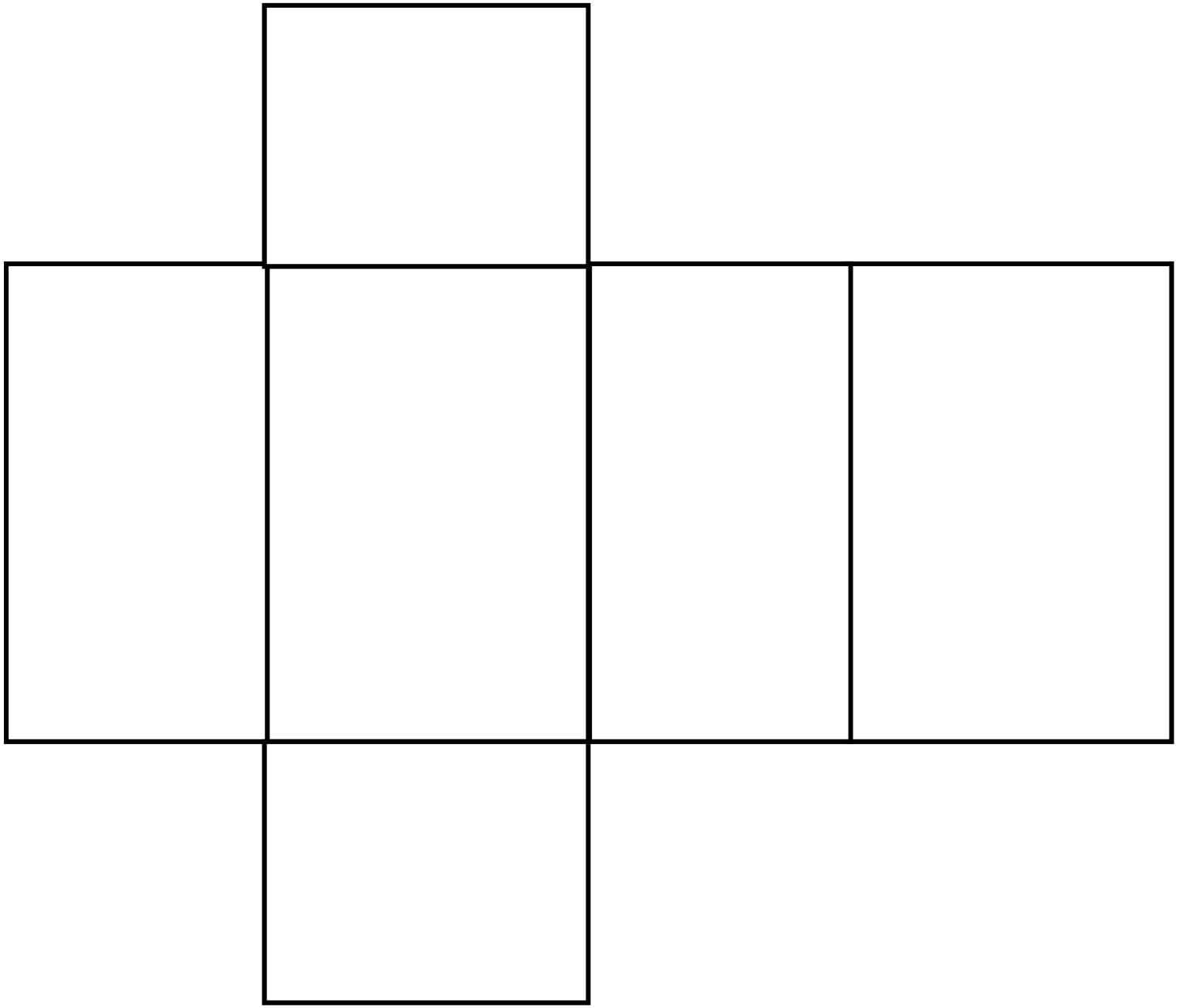


箱の寸法デザイン問題：解答



各辺の比率　　1　：　0.67676　：　0.54626

このときのコスト：　0.09117



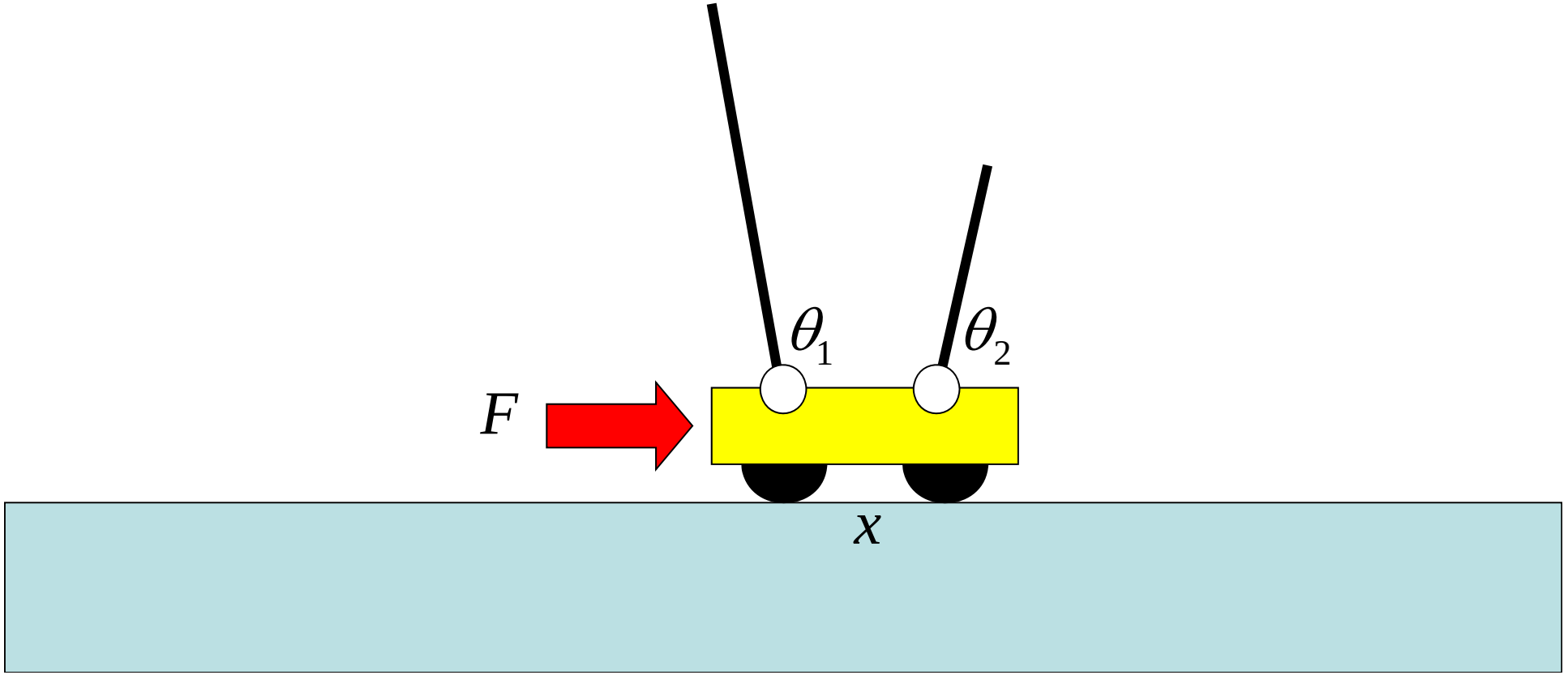
【例：二重倒立振子制御問題】

●台車の位置と速度、および2本の振子（棒）のそれぞれの角度と角速度の計6つの状態量を用いて、2本の棒が倒立した状態で安定して立つように台車を動かすようなコントローラのフィードバックゲインを設計する。ただし台車に作用させる力を制御する。

$$F = w_1 x + w_2 \dot{x} + w_3 \theta_1 + w_4 \dot{\theta}_1 + w_5 \theta_2 + w_6 \dot{\theta}_2$$

コスト関数＝（10000 － steps）　ただし倒れた時点あるいは10000stepで打切り

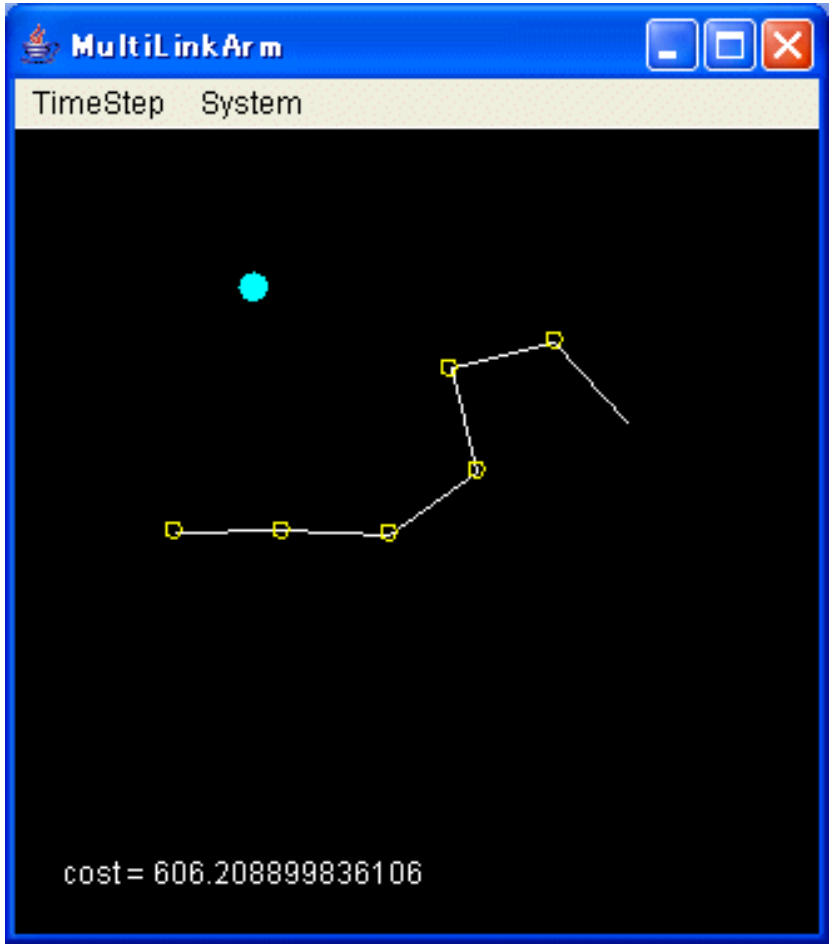
1 step =0.02 sec

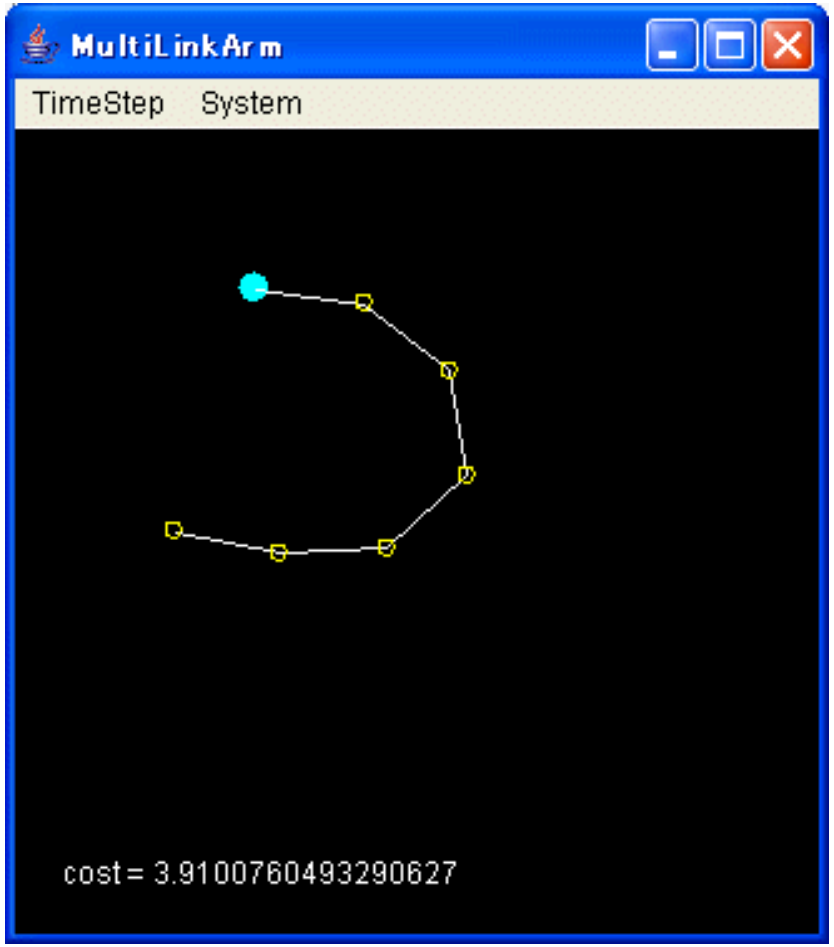


【例：冗長自由度アームのリーチング問題】

●各関節の角度をパラメータとして、アーム先端がターゲット座標に一致するパラメータを求める問題。ただし、各関節はリンク同士を±90度以内で繋ぐ制約があるので、なるべく関節の可動角度中央付近の角度にすることが好ましい。

コスト関数＝（アーム先端座標とターゲット座標の偏差の2乗）
＋（各関節の可動中央角度からの偏差の2乗）

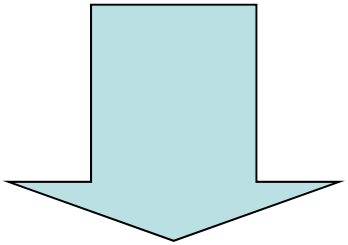




かなり粗い

１０個のパラメータを持つ問題において、各パラメータが0～1の区間の値をとるものとして、**各パラメータを 0.1 刻み**で全てのパラメータの組合せを評価すると、評価回数は　11 の10 乗、これをおおよそ 10 の10乗で近似しても、　10,000,000,000（百億）回必要になる。

１回の評価に1秒かかるものとすると、**計算には 317年 を要する**



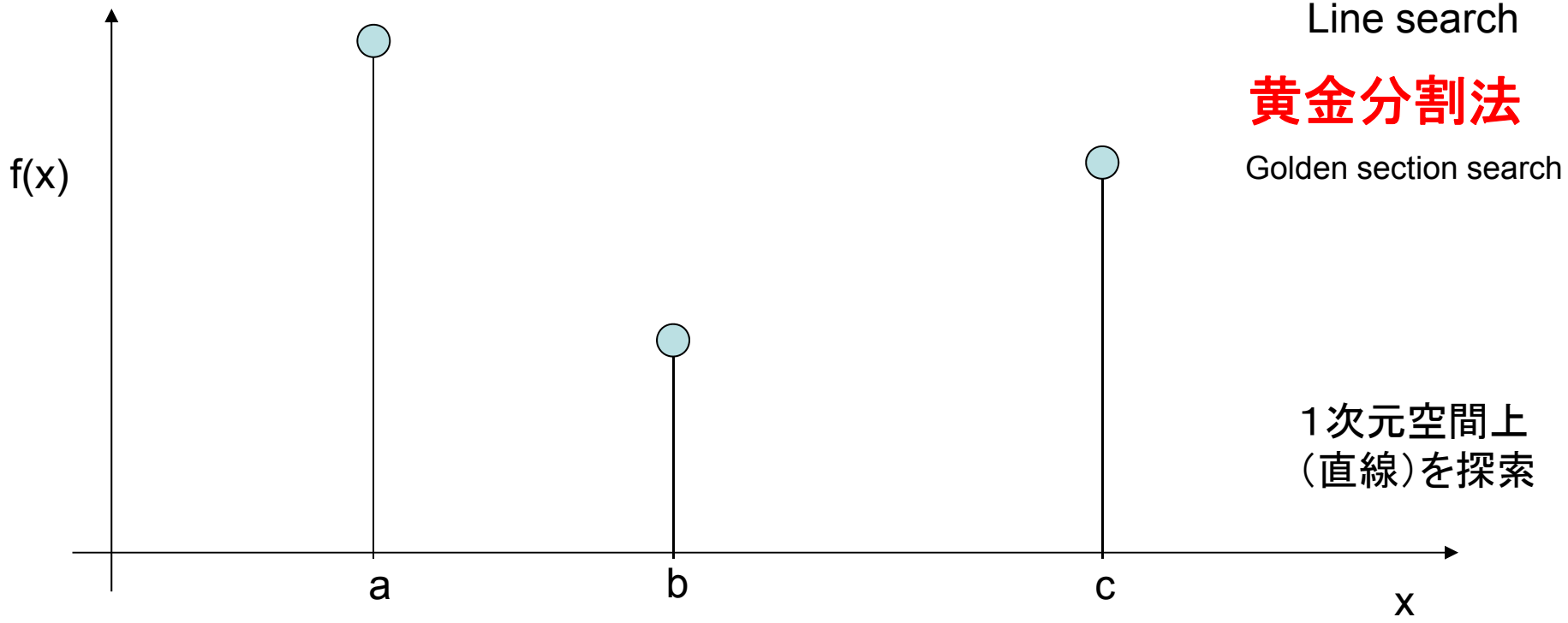
パラメータの探し方（探索方法）を工夫しなければ
答えを見つけることができない

【制約のない関数最適化】1次元の最小化：ラインサーチ

Line search

黄金分割法

Golden section search



1） a < b < c の3点をとる　f(b) が最小なら、区間 [a,c] に必ず極小がある。

2） [b, c] の区間で新しい点 x をとり、f(x) を計算する。

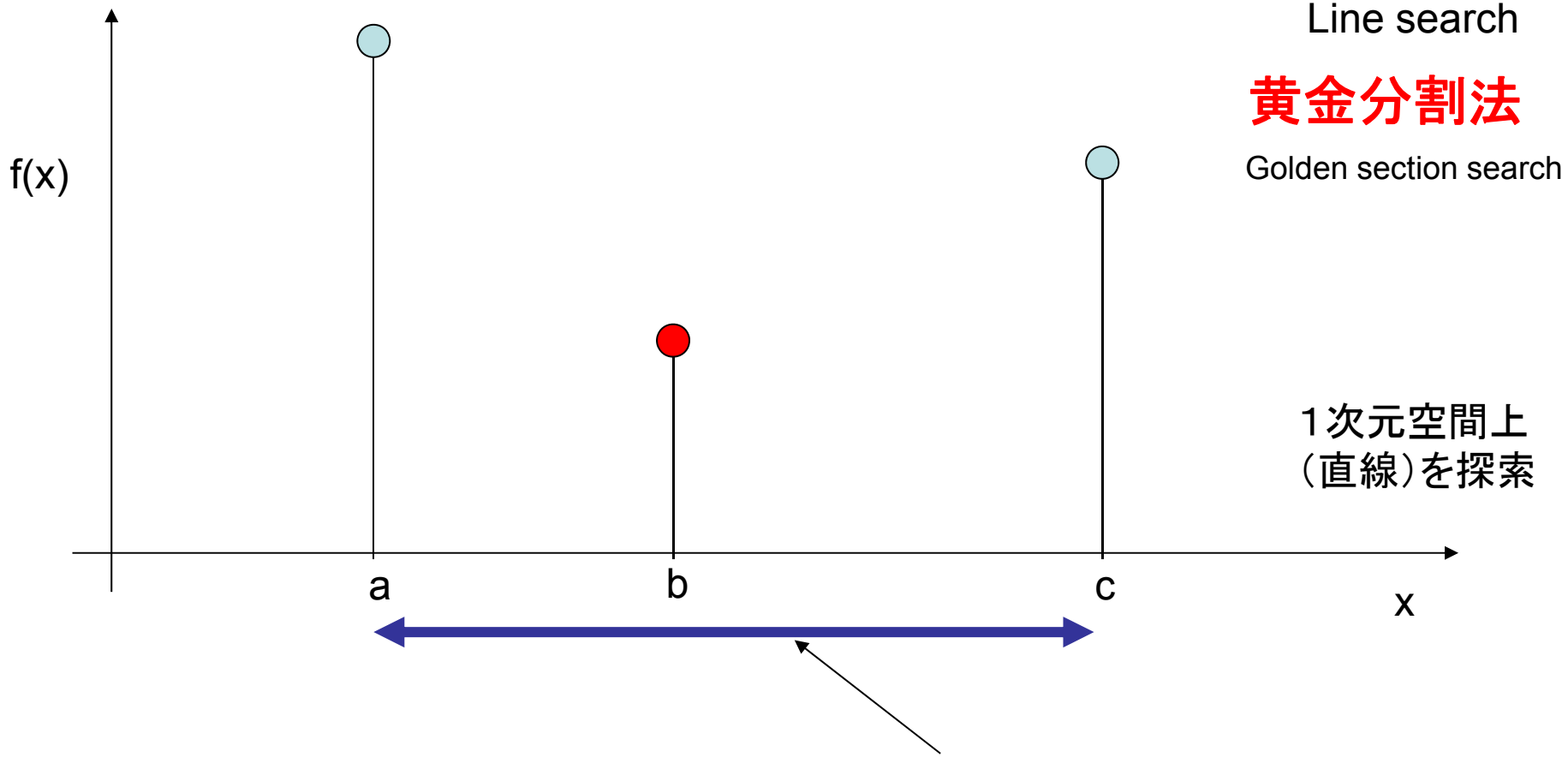
3） f(b) < f(x) なら、a < b < x に狭められた範囲に極小がある
f(x) < f(b) なら、b < x < c に狭められた範囲に極小がある

【制約のない関数最適化】1次元の最小化：ラインサーチ

Line search

黄金分割法

Golden section search



1） a < b < c の3点をとる　f(b) が最小なら、区間 [a,c] に必ず極小がある。

2） [b, c] の区間で新しい点 x をとり、f(x) を計算する。

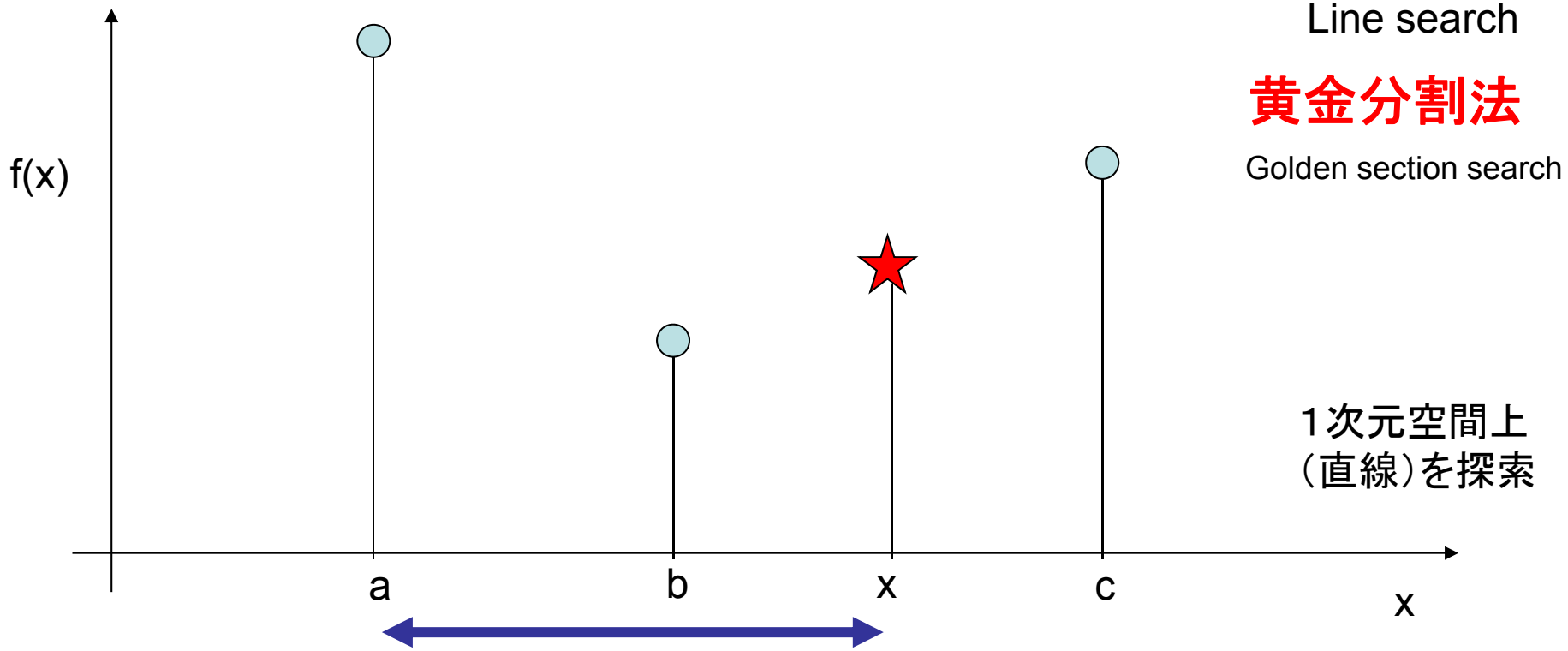
3） f(b) < f(x) なら、a < b < x に狭められた範囲に極小がある
f(x) < f(b) なら、b < x < c に狭められた範囲に極小がある

【制約のない関数最適化】1次元の最小化：ラインサーチ

Line search

黄金分割法

Golden section search



1） a < b < c の3点をとる　f(b) が最小なら、区間 [a,c] に必ず極小がある。

2） [b, c] の区間で新しい点 x をとり、f(x) を計算する。

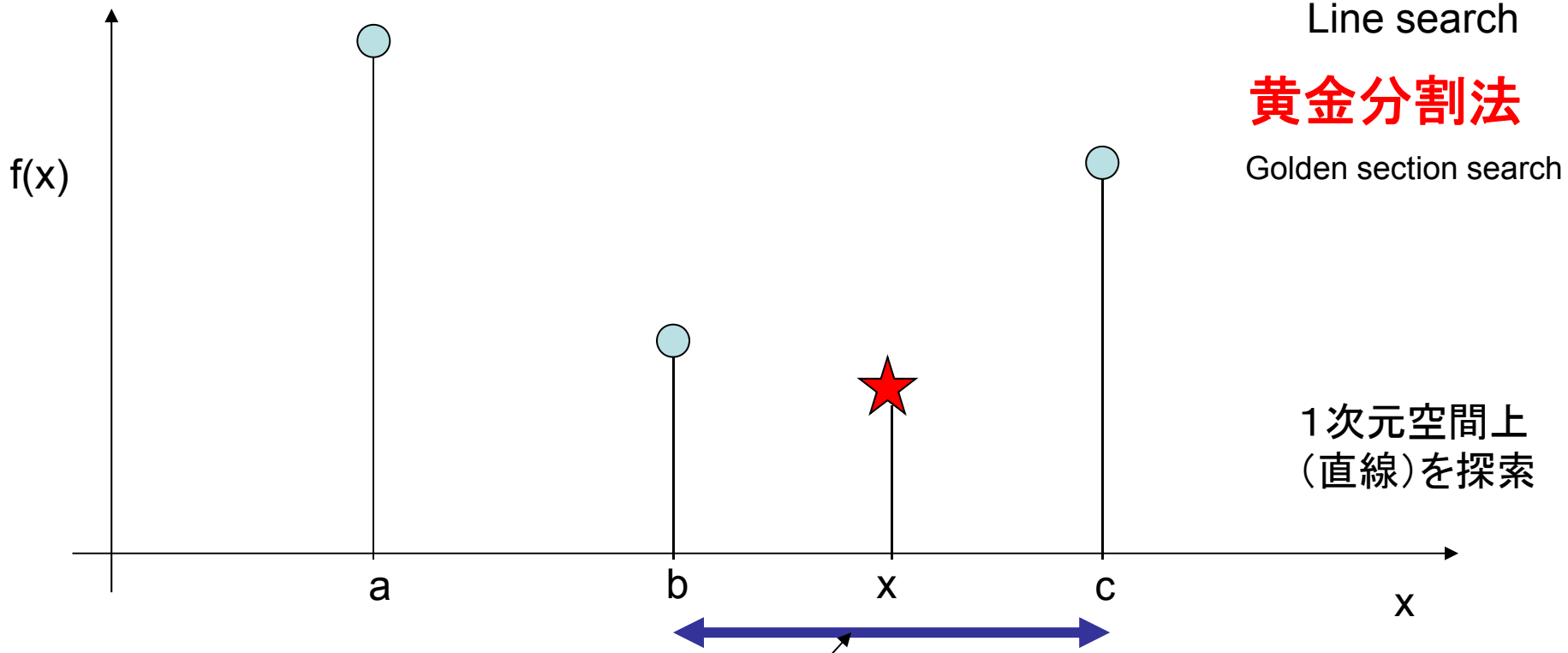
3） f(b) < f(x) なら、a < b < x に狭められた範囲に極小がある→ a,b,x をa,b,c に置換
f(x) < f(b) なら、b < x < c に狭められた範囲に極小がある

【制約のない関数最適化】1次元の最小化：ラインサーチ

Line search

黄金分割法

Golden section search



1） a < b < c の3点をとる　f(b) が最小なら、区間 [a,c] に必ず極小がある。

2） [b, c] の区間で新しい点 x をとり、f(x) を計算する。

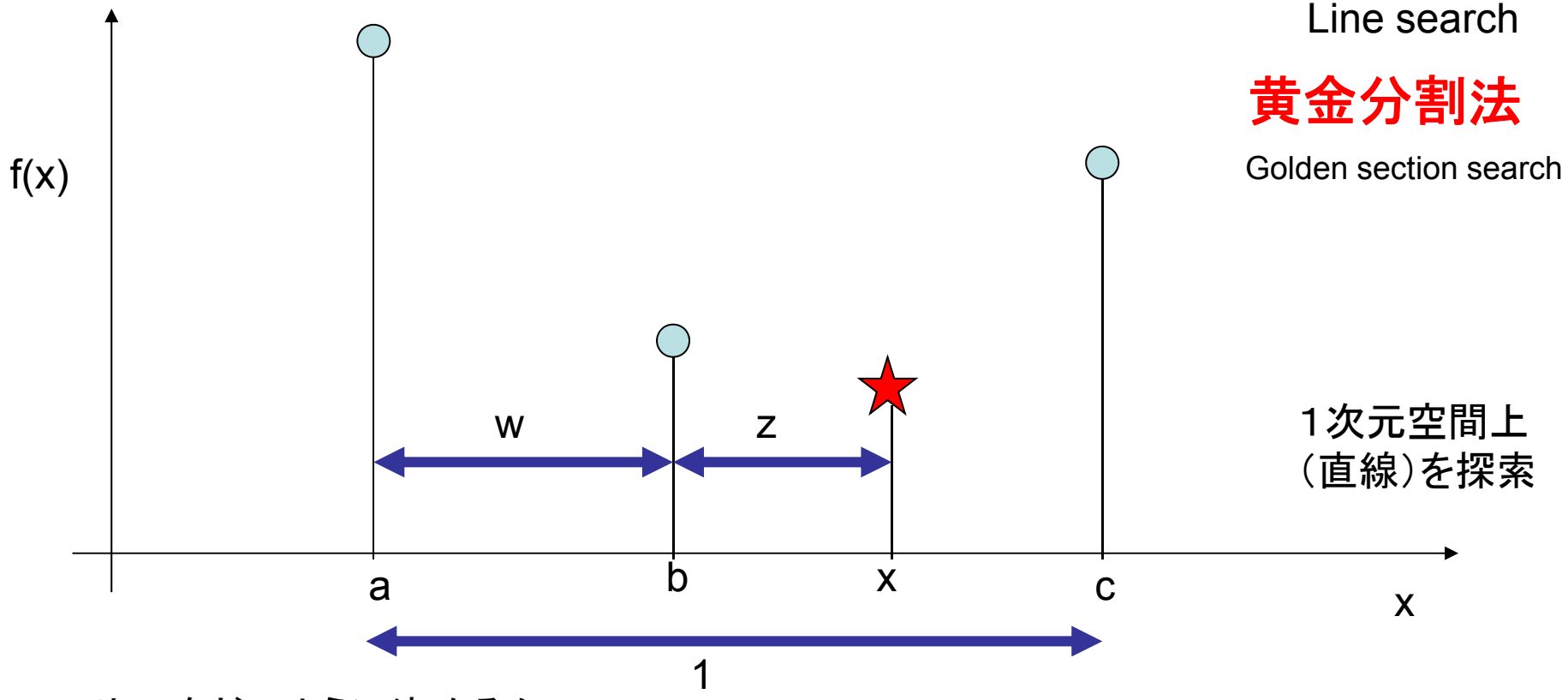
3） f(b) < f(x) なら、a < b < x に狭められた範囲に極小がある
f(x) < f(b) なら、b < x < c に狭められた範囲に極小がある→ b,x,c をa,b,c に置換

【制約のない関数最適化】1次元の最小化：ラインサーチ

Line search

黄金分割法

Golden section search



b や x をどのように決めるか？

次の新しい区間長は w+z または 1-w 。これらを等しくすると z = 1 - 2w

区間[b,c]に対するxの位置関係は、区間[a,c]に対するbの位置関係に等しい

これらの連立方程式を解くと、

$$w = \frac{3-\sqrt{5}}{2}$$

黄金比

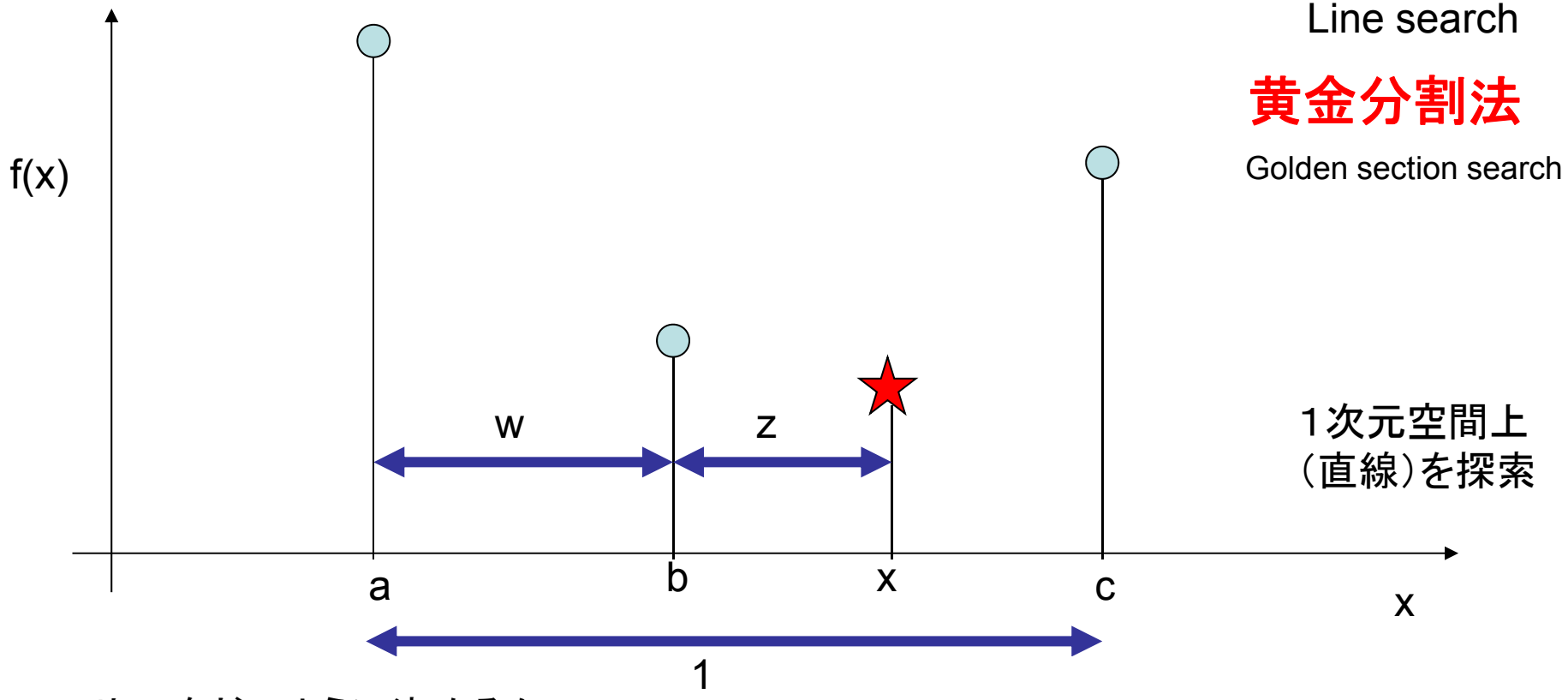
Golden section

【制約のない関数最適化】1次元の最小化：ラインサーチ

Line search

黄金分割法

Golden section search



b や x をどのように決めるか？

次の新しい区間長は w+z または 1-w 。これらを等しくすると z = 1 - 2w

区間[b,c]に対するxの位置関係は、区間[a,c]に対するbの位置関係に等しい

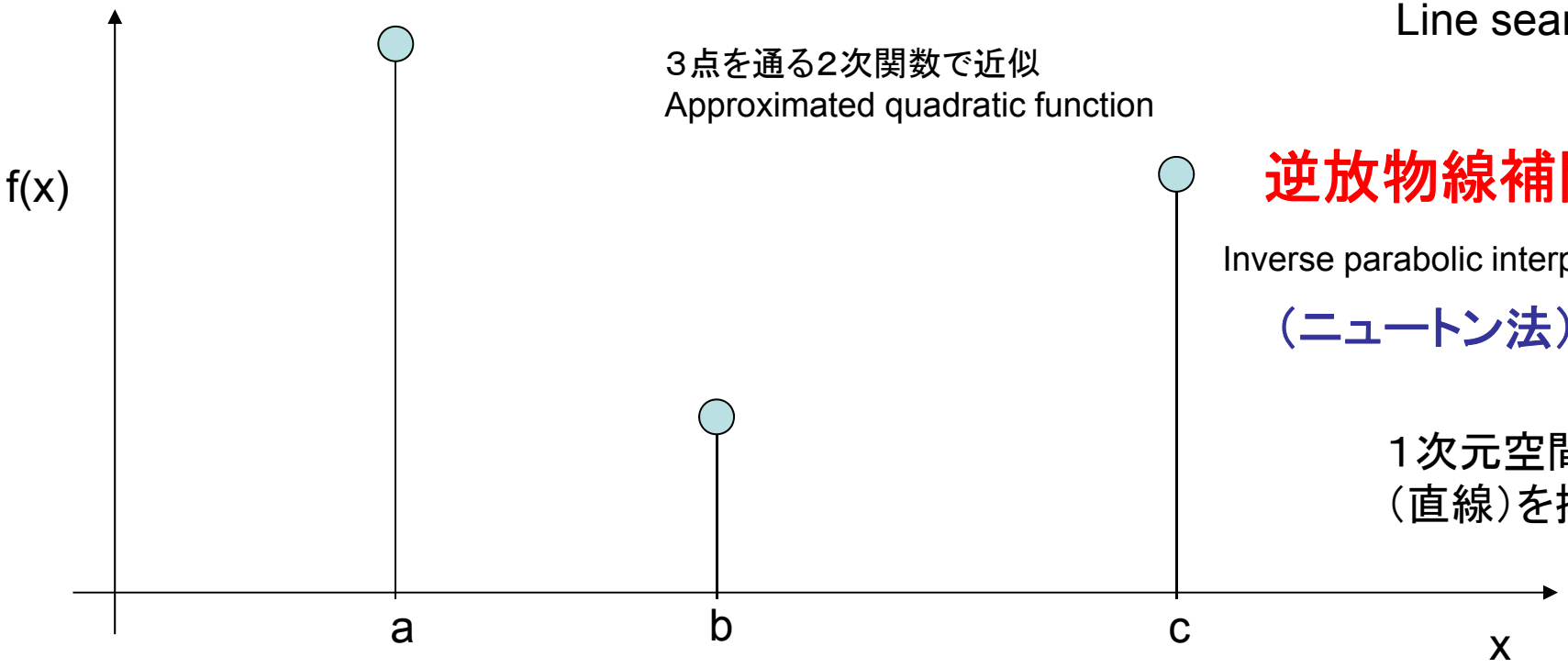
1 : w = (1 – w) : z

これらの連立方程式を解くと、 $w^2 - 3w + 1 = 0$ ⇒ $w = \frac{3-\sqrt{5}}{2}$

黄金比 Golden section

1次元の最小化：ラインサーチ

Line search



3点を通る2次関数で近似
Approximated quadratic function

逆放物線補間

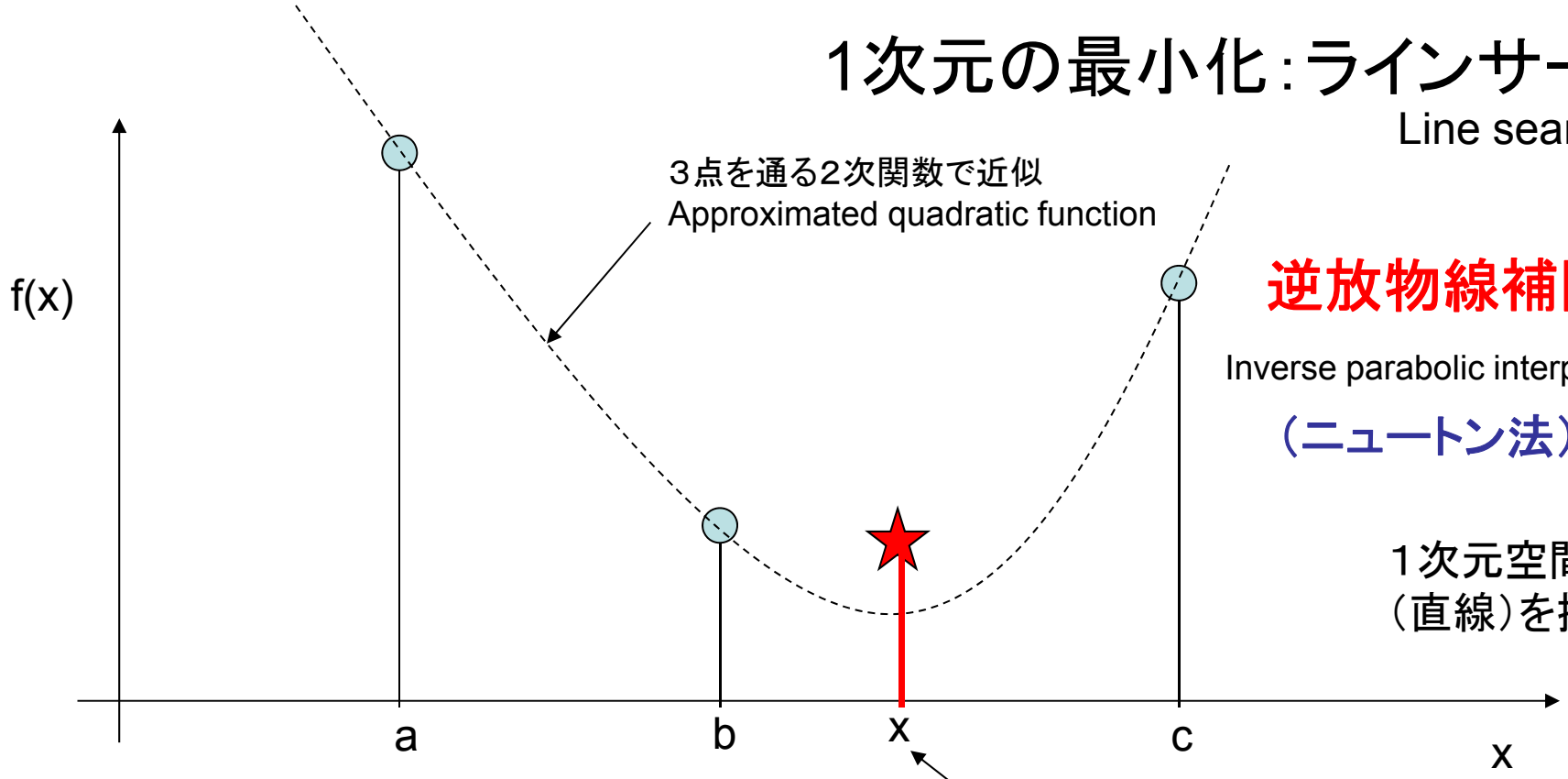
Inverse parabolic interpolation

（ニュートン法）

1次元空間上
（直線）を探索

1次元の最小化：ラインサーチ

Line search



逆放物線補間

Inverse parabolic interpolation

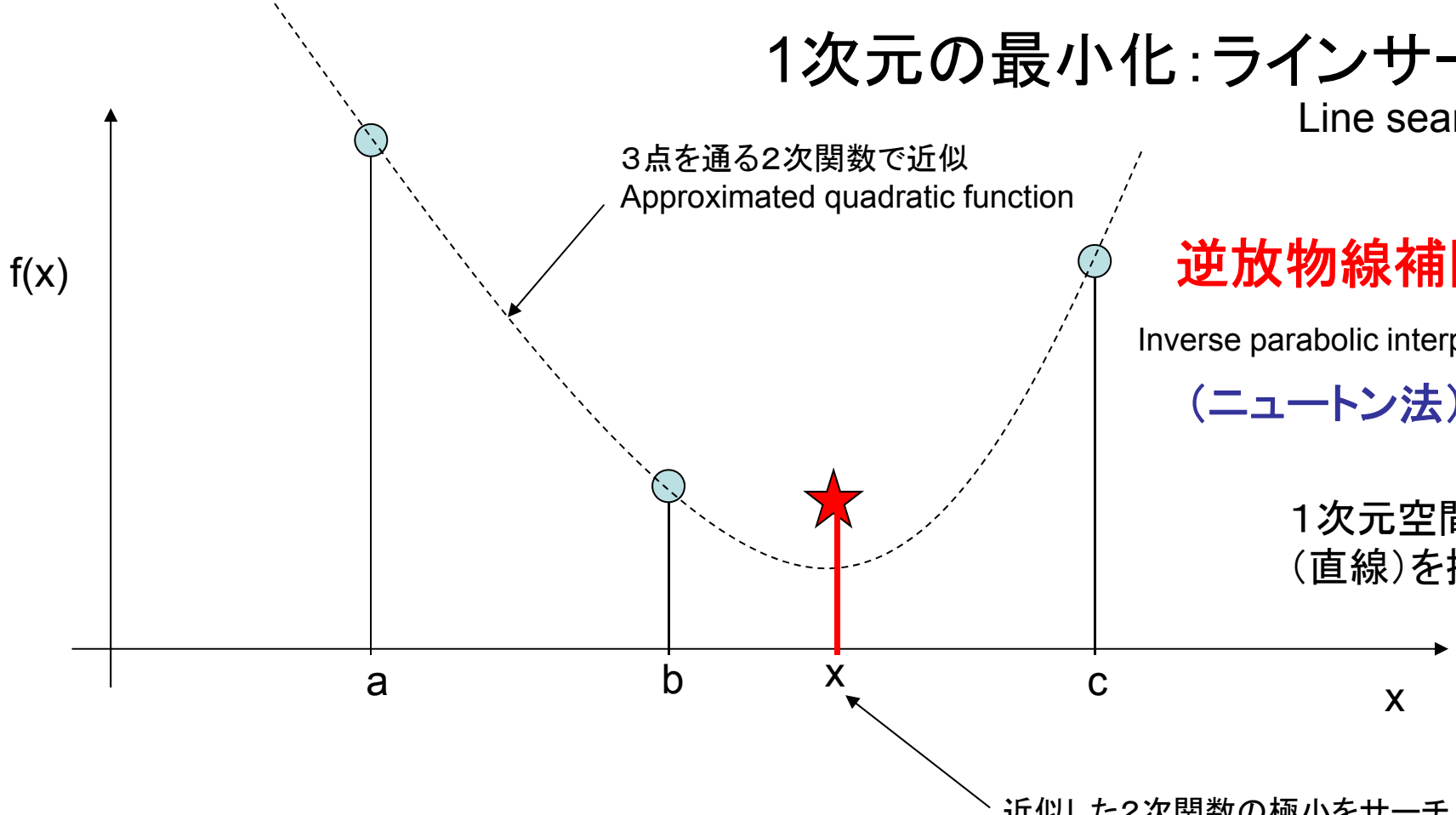
（ニュートン法）

1次元空間上
（直線）を探索

近似した2次関数の極小をサーチ
Search this point where gradient of the approximated function is zero.

$x =$

1次元の最小化：ラインサーチ
Line search



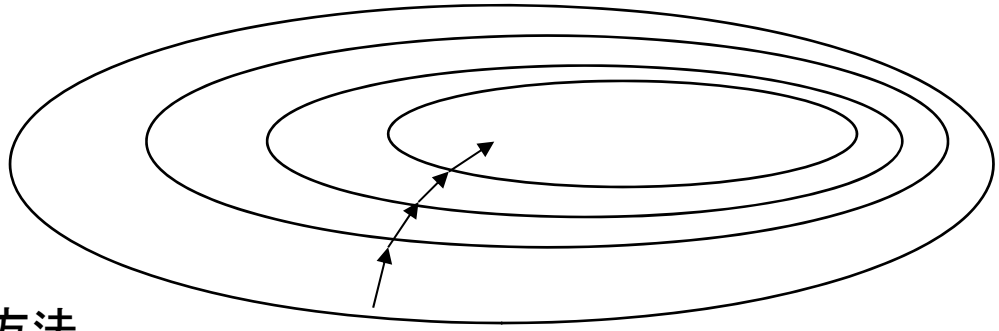
$$x = b - \frac{1}{2}\frac{(b-a)^2\big(f(b)-f(c)\big)-(b-c)^2\big(f(b)-f(a)\big)}{(b-a)\big(f(b)-f(c)\big)-(b-c)\big(f(b)-f(a)\big)}$$

ただし、3点が1直線にあると、分母がゼロになってこの式は使えない
This equation is void when the tree points are on a same straight line.

【制約のない関数最適化】 n次元関数の最小化

optimization in n-dimensional function

探索点での関数の値（と勾配の大きさ）に基づいて次の探索点を順次決定する手順により、あたかも山の頂上を目指すように、関数の極大値（あるいは極小値）をすみやかに発見する手法



1）**勾配情報を用いる方法**：

・最大勾配法（最急降下法）

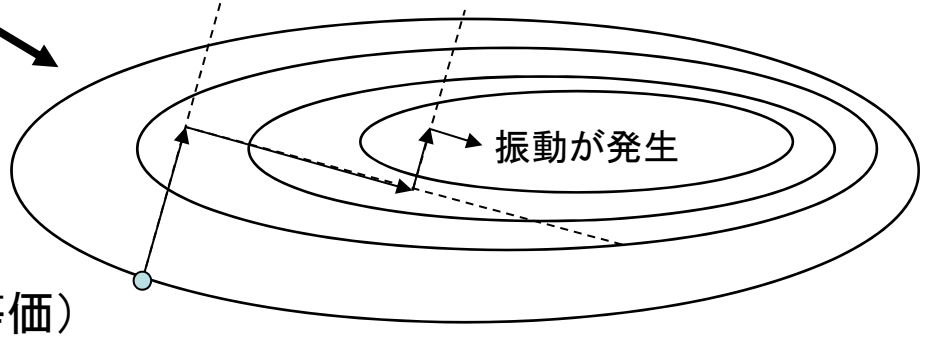
勾配方向へ少しずつ解を改善

ステップ幅を勾配に比例させて極小点で停止

●ラインサーチと方向転換を繰り返す方法

・最適勾配法：勾配方向にラインサーチを行い、その極小点で再び勾配方向へラインサーチ

だいたいうまく行くが、2次関数において極めて効率の悪い場合がある



・ 共役勾配法 ：方向転換するとき、その点の勾配方向だけでなく、今まで下ってきた方向も考える

（関数の2階偏導関数を考慮に入れることと等価）

対象とする関数がn変数の2次形式である場合、

ラインサーチをn回繰返すことで最大値が求まることが保障される

2）**勾配情報を用いない方法**： 滑降シンプレックス法

たいした計算を要しないで手っ取り早く動くアルゴリズムが欲しい場合に最善の方法

【復習】 関数の勾配とは？

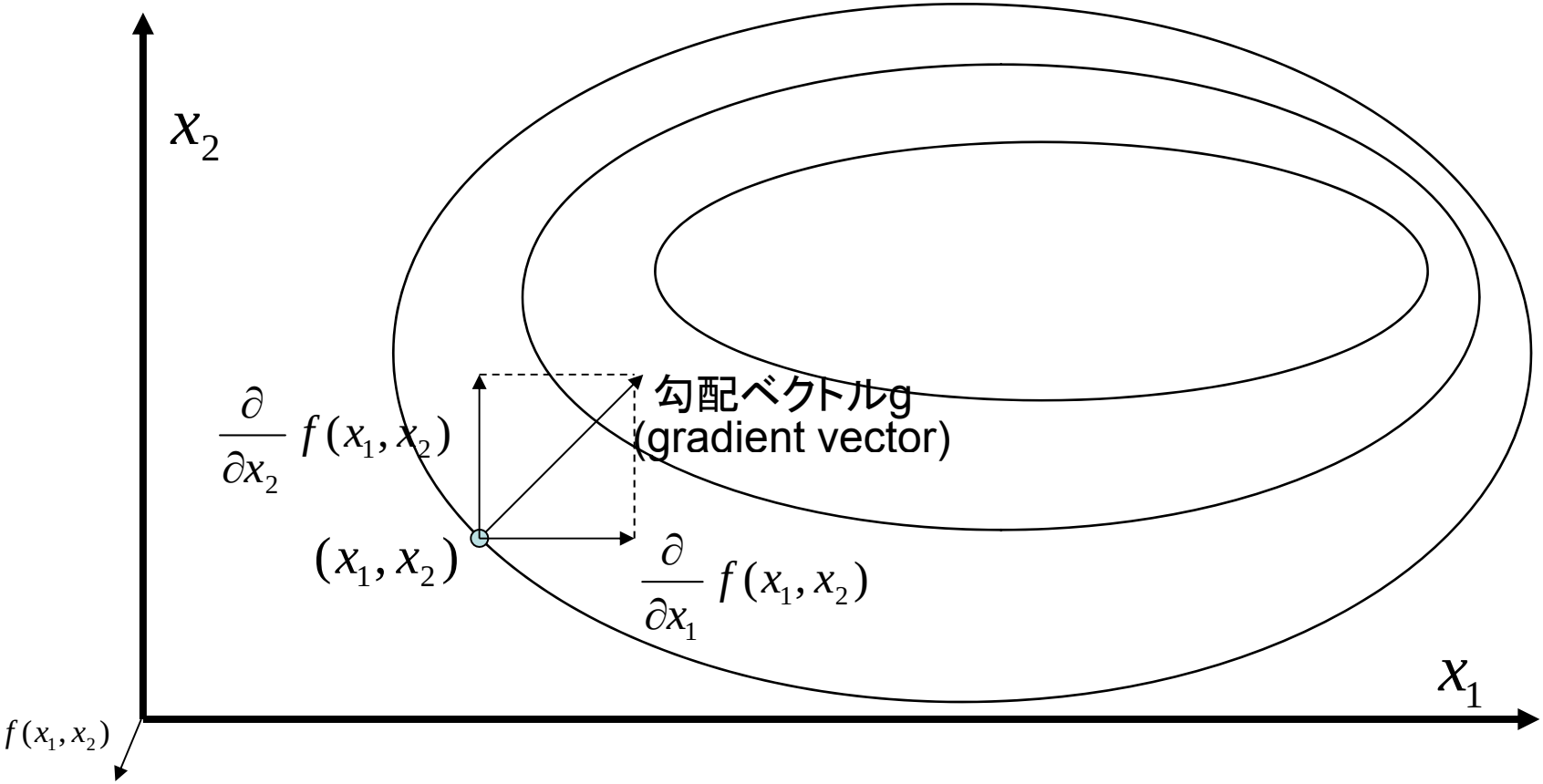
(gradient)

2変数関数 $f(x_1, x_2)$

探索すべき
パラメータベクトル $\mathbf{x} = (x_1, x_2)$

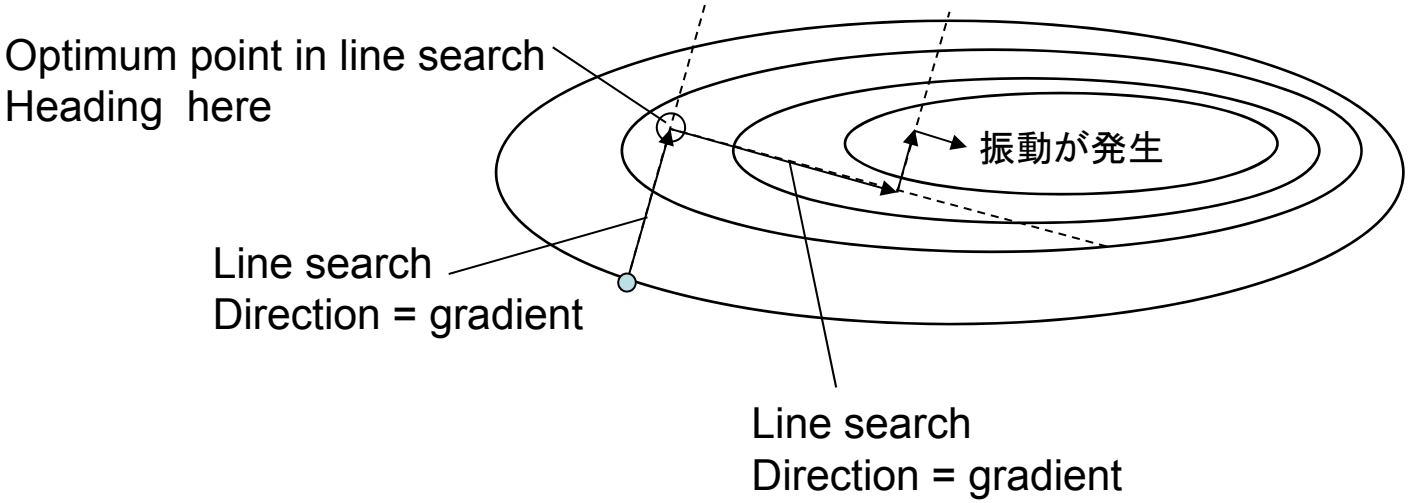
勾配ベクトル
(gradient vector)

$$\mathbf{g} = \nabla f(x_1, x_2) = \left(\frac{\partial}{\partial x_1} f(x_1, x_2), \frac{\partial}{\partial x_2} f(x_1, x_2) \right)$$



ラインサーチと方向転換を繰り返す方法（1）
最適勾配法（Optimal　gradient　method）

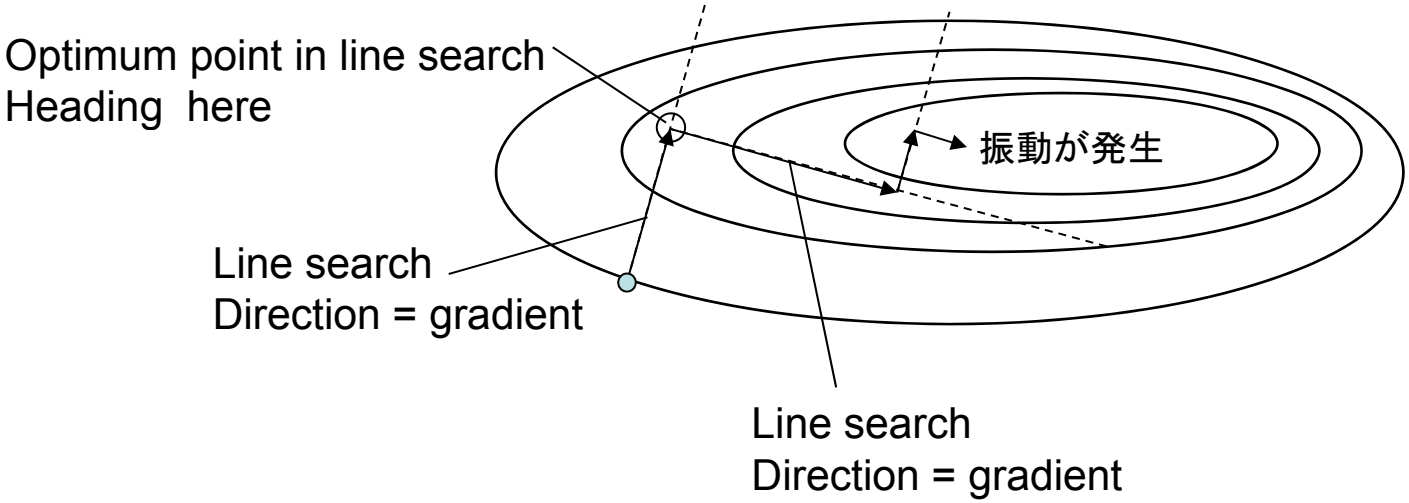
勾配方向にラインサーチを行い、極小点で再び勾配を求め、その方向へラインサーチする
Execute line search in direction of the gradient to its optimum point,
and turn to the new gradient direction, and execute line search again.



【問題点】
２次関数において、細かいステップで方向転換を繰り返しながら
同じような方向を何度もラインサーチを行い、谷底に着くまで多数のステップを要する
It needs many steps in quadratic functions repeating the line search
in similar direction.

ラインサーチと方向転換を繰り返す方法（1）
最適勾配法（Optimal　gradient　method）

勾配方向にラインサーチを行い、極小点で再び勾配を求め、その方向へラインサーチする
Execute line search in direction of the gradient to its optimum point,
and turn to the new gradient direction, and execute line search again.



【問題点】
２次関数において、細かいステップで方向転換を繰り返しながら
同じような方向を何度もラインサーチを行い、谷底に着くまで多数のステップを要する
It needs many steps in quadratic functions repeating the line search
in similar direction.

新しい探索方向は、今までの全ての探索方向とも干渉しない（直交・共役である）
ことが望ましい A new direction should be conjugate against all past directions.

なぜ「2次形式」が重要なのか？
(quadratic)

ある特定の点Pを原点とし、この点の近傍座標をxとする。
すると、**どんな関数 f もテイラー級数で近似できる**:
(Taylor series)

$$f(\mathbf{x}) = f(\mathbf{P}) + \sum_i \frac{\partial f}{\partial x_i} x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j} x_i x_j + \cdots$$

$$= f(\mathbf{P}) + \nabla f\,|_{\mathbf{P}} \cdot \mathbf{x} + \frac{1}{2} \mathbf{x}^t \cdot \mathbf{A} \cdot \mathbf{x} + \cdots$$

2次形式
(quadratic)

この行列A はf(x)の ..... 行列

各要素はPにおける2階編導関数

なぜ「2次形式」が重要なのか？
(quadratic)

ある特定の点Pを原点とし、この点の近傍座標をxとする。
すると、**どんな関数 f もテイラー級数で近似できる**:
(Taylor series)

$$f(\mathbf{x}) = f(\mathbf{P}) + \sum_i \frac{\partial f}{\partial x_i} x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j} x_i x_j + \cdots$$

$$= f(\mathbf{P}) + \nabla f \,|_{\mathbf{P}} \cdot \mathbf{x} + \frac{1}{2} \mathbf{x}^t \cdot \mathbf{A} \cdot \mathbf{x} + \cdots$$

2次形式
(quadratic)

この行列A はf(x)の Hesse 行列

Hessian matrix

各要素はPにおける2階編導関数

$$[\mathbf{A}]_{ij} \equiv \left. \frac{\partial^2 f}{\partial x_i \partial x_j} \right|_{\mathbf{P}}$$

【復習】「2次形式」と「共役」な方向とは？

(quadratic)　(conjugate)

$$f(x_1, x_2, \cdots, x_n) = \boxed{}$$

← **2次形式** (quadratic)

ただし

探索すべき
パラメータベクトル
(parameter vector)

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Symmetrical matrix

N×N 対称行列 $\mathbf{A}$ に対して、2つの方向を表すベクトル $\mathbf{p}$ と $\mathbf{q}$ が

$\mathbf{p}^t\mathbf{A}\mathbf{q} = 0$ を満たすとき、 $\mathbf{p}$ と $\mathbf{q}$ は $\mathbf{A}$ に関して互いに**共役**であるという

(conjugate)

参考：$\mathbf{A}$ を単位行列とすると、上の式は直交条件となり、共役性は直交性の拡張概念

また、**「複素共役」とは無関係**

【復習】「2次形式」と「共役」な方向とは？

(quadratic)　(conjugate)

$$f(x_1, x_2, \cdots, x_n) = \boxed{\mathbf{X}^t\mathbf{A}\mathbf{X} + \mathbf{B}^t\mathbf{X} + C}$$

← 2次形式 (quadratic)

ただし

探索すべき
パラメータベクトル
(parameter vector)

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Symmetrical matrix

N×N 対称行列 $\mathbf{A}$ に対して、2つの方向を表すベクトル $\mathbf{p}$ と $\mathbf{q}$ が

$\mathbf{p}^t\mathbf{A}\mathbf{q} = 0$ を満たすとき、 $\mathbf{p}$ と $\mathbf{q}$ は $\mathbf{A}$ に関して互いに**共役**であるという

(conjugate)

参考：$\mathbf{A}$ を単位行列とすると、上の式は直交条件となり、共役性は直交性の拡張概念

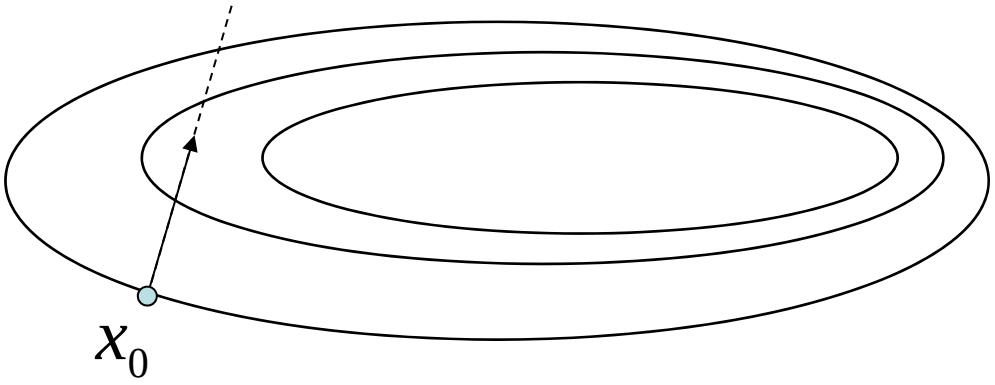
また、「**複素共役**」とは無関係

【制約のない関数最適化】共役勾配法 Conjugate gradient method

x_i i 番目の点の座標ベクトル
Coordinate of the i[th] search point

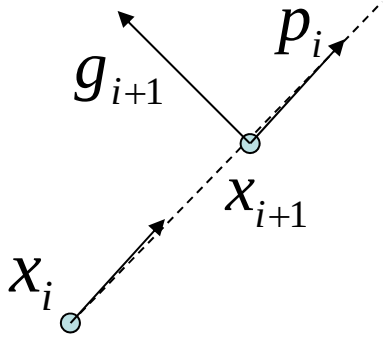
g_i 点 x_i における勾配ベクトル
Gradient vector at x_i

p_i 点 x_i からラインサーチを行う方向ベクトル
Direction of line search at x_i



まず最初は勾配の反対方向へラインサーチ

$$p_0 = -g_0$$



x_i の次の点 x_{i+1} を、ラインサーチで見つけた最小点とする

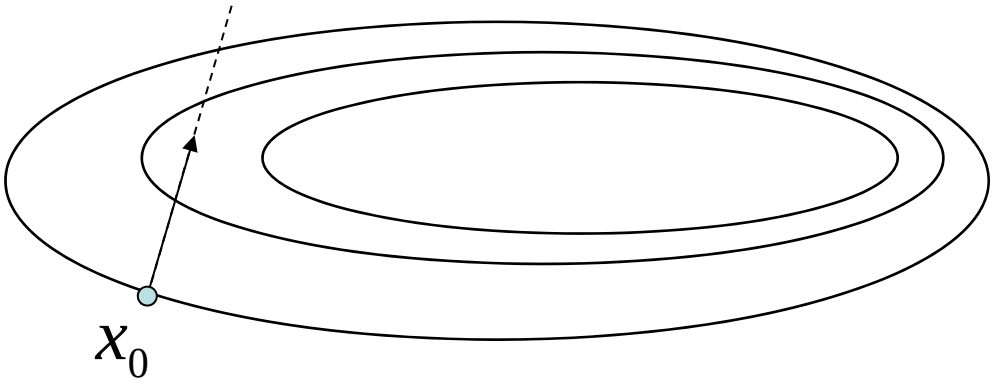
$$x_{i+1} = x_i + \alpha_i \, p_i$$ Optimum point in line search

【制約のない関数最適化】共役勾配法 Conjugate gradient method

x_i i 番目の点の座標ベクトル
Coordinate of the i[th] search point

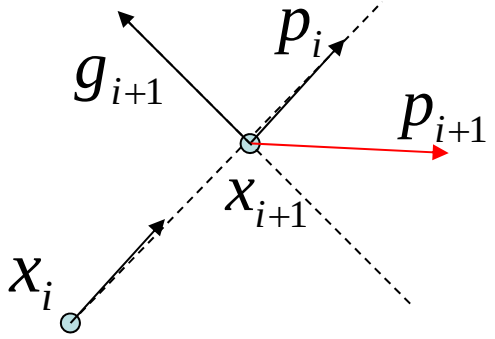
g_i 点 x_i における勾配ベクトル
Gradient vector at x_i

p_i 点 x_i からラインサーチを行う方向ベクトル
Direction of line search at x_i



まず最初は勾配の反対方向へラインサーチ

$$p_0 = -g_0$$



x_i の次の点 x_{i+1} を、ラインサーチで見つけた最小点とする

$$x_{i+1} = x_i + \alpha_i \, p_i$$ Optimum point in line search

次の点 x_{i+1} での**新しい探索方向** p_{i+1} **は、前の探索方向** p_i **と共役**であるようにする。
そのため、点 x_{i+1} での勾配と前の探索方向 p_i とを結合する:

$$p_{i+1} = -g_{i+1} + \beta_{i+1} \, p_i$$

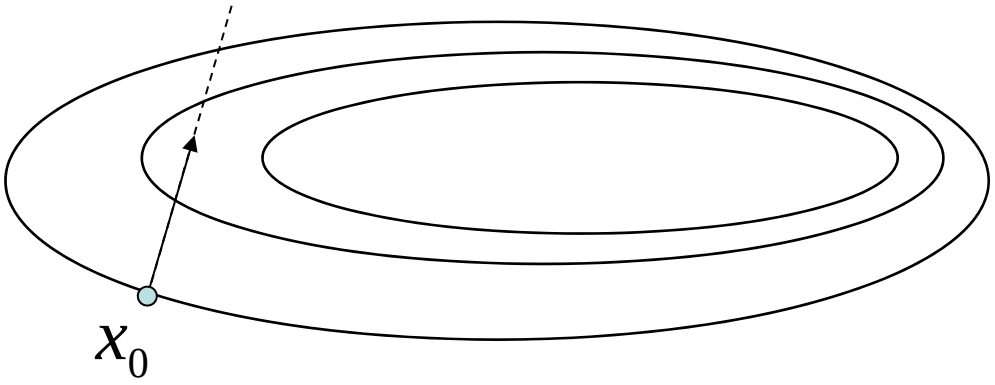
重み係数

【制約のない関数最適化】共役勾配法 Conjugate gradient method

x_i i 番目の点の座標ベクトル
Coordinate of the i[th] search point

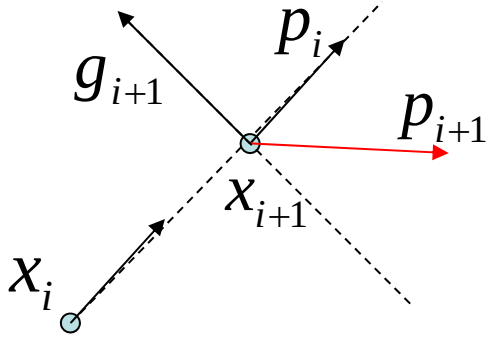
g_i 点 x_i における勾配ベクトル
Gradient vector at x_i

p_i 点 x_i からラインサーチを行う方向ベクトル
Direction of line search at x_i



まず最初は勾配の反対方向へラインサーチ

$$p_0 = -g_0$$



x_i の次の点 x_{i+1} を、ラインサーチで見つけた最小点とする

$$x_{i+1} = x_i + \alpha_i\ p_i$$ Optimum point in line search

次の点 x_{i+1} での**新しい探索方向** p_{i+1} **は、前の探索方向** p_i **と共役**であるようにする。
そのため、点 x_{i+1} での勾配と前の探索方向 p_i とを結合する:

$$p_{i+1} = -g_{i+1} + \beta_{i+1}\ p_i$$

ただし、重み係数 β_{i+1} は次の式のどちらかで与える:

$\beta_{i+1} =$ [ ] （Fletcher-Reeves法）

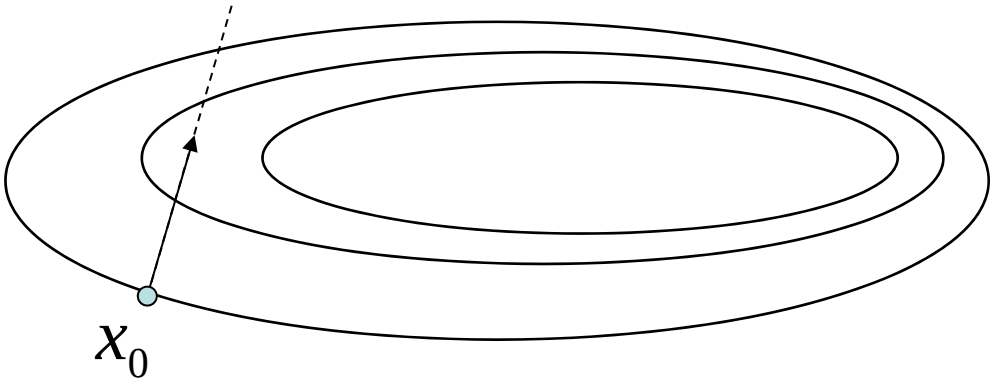
（Polak-Ribiere法） $\beta_{i+1} =$ [ ]

【制約のない関数最適化】共役勾配法 Conjugate gradient method

x_i i 番目の点の座標ベクトル
Coordinate of the i[th] search point

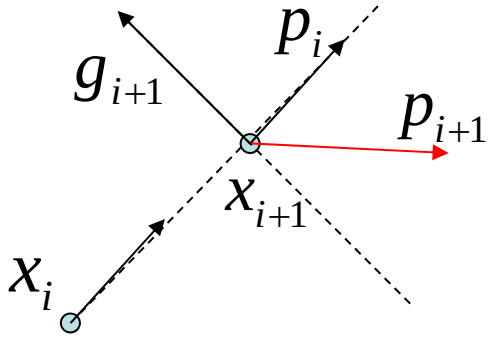
g_i 点 x_i における勾配ベクトル
Gradient vector at x_i

p_i 点 x_i からラインサーチを行う方向ベクトル
Direction of line search at x_i



まず最初は勾配の反対方向へラインサーチ

$$p_0 = -g_0$$



x_i の次の点 x_{i+1} を、ラインサーチで見つけた最小点とする

$$x_{i+1} = x_i + \alpha_i \, p_i \quad \text{Optimum point in line search}$$

次の点 x_{i+1} での**新しい探索方向** p_{i+1} **は、前の探索方向** p_i **と共役**であるようにする。
そのため、点 x_{i+1} での勾配と前の探索方向 p_i とを結合する:

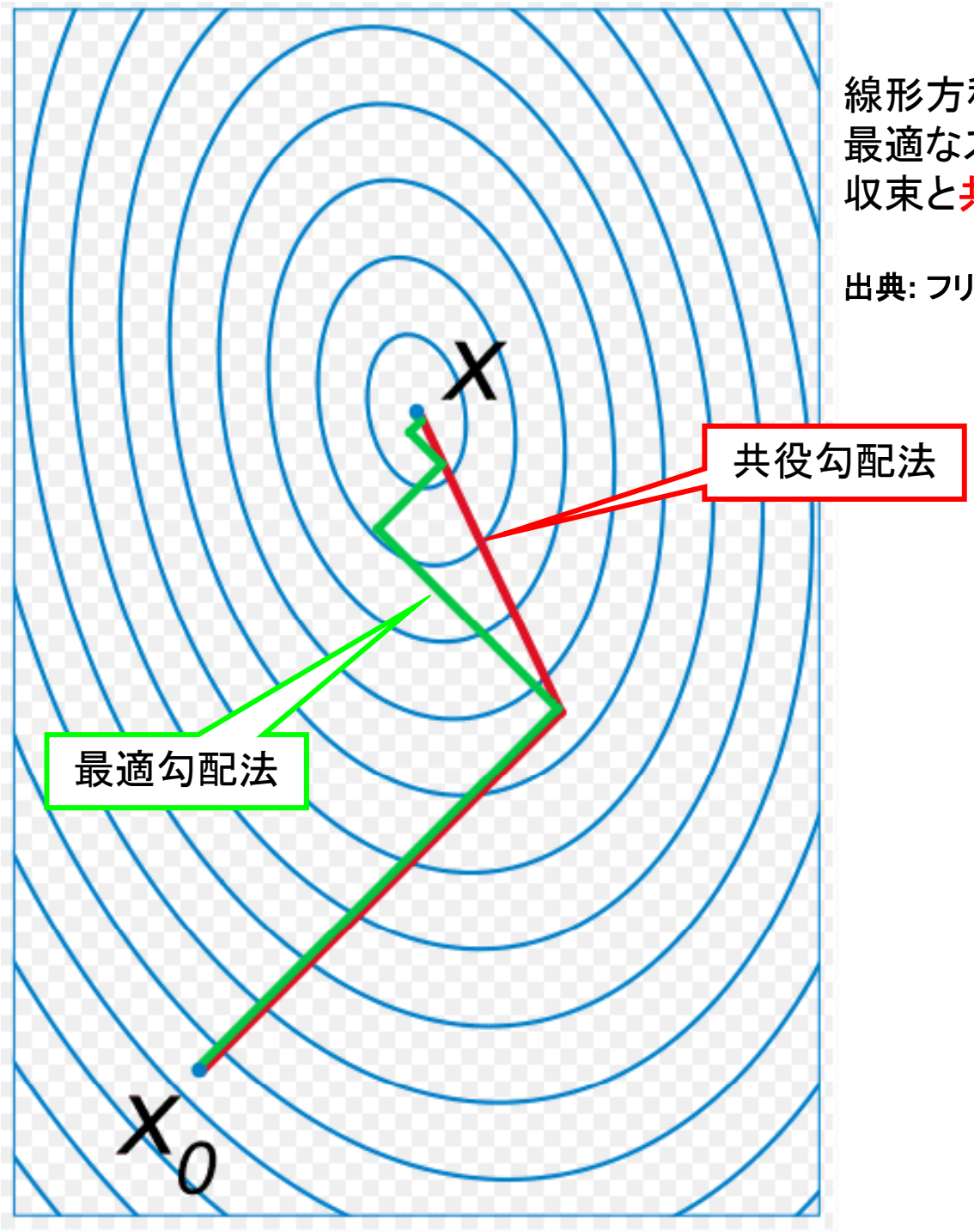
$$p_{i+1} = -g_{i+1} + \beta_{i+1} \, p_i$$

ただし、重み係数 β_{i+1} は次の式のどちらかで与える:

$$\beta_{i+1} = \frac{g_{i+1}^T \, g_{i+1}}{g_i^T \, g_i} = \frac{\|g_{i+1}\|^2}{\|g_i\|^2}$$ （Fletcher-Reeves法）

（Polak-Ribiere法）

$$\beta_{i+1} = \frac{(g_{i+1} - g_i)^T \, g_{i+1}}{g_i^T \, g_i}$$



線形方程式の2次形式を最小化するための、
最適なステップサイズによる**最適勾配法**の
収束と**共役勾配法**の収束の比較

出典: フリー百科事典『ウィキペディア（Wikipedia）』

【共役勾配法が2次形式関数でn回のラインサーチで最適解を見つける理由】

次式を満たす n 個の n 次元ベクトル $\mathbf{Z}_1, \mathbf{Z}_2, \cdots \mathbf{Z}_n$ を考える:

$$\mathbf{Z}_i{}^t \mathbf{A} \mathbf{Z}_j = \begin{cases} const. & for \quad i = j \\ 0 & for \quad i \neq j \end{cases}$$

⇒ $\mathbf{Z}_1, \mathbf{Z}_2, \cdots \mathbf{Z}_n$ は共役ベクトル集合

次に、ベクトル $\mathbf{X}$ を $\mathbf{X} = \sum_{j=1}^{n} \alpha_j \mathbf{Z}_j$ と表すと、2次形式の評価式は

$$f(x_1, x_2, \cdots, x_n) = \mathbf{X}^t \mathbf{A} \mathbf{X} + \mathbf{B}^t \mathbf{X} + C$$

$$= \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \mathbf{Z}_i{}^t \mathbf{A} \mathbf{Z}_j \right) + \left(\sum_{i=1}^{n} \alpha_i \mathbf{B}^t \mathbf{Z}_i \right) + C = \sum_{i=1}^{n} \alpha_i{}^2 \mathbf{Z}_i{}^t \mathbf{A} \mathbf{Z}_i + \alpha_i \mathbf{B}^t \mathbf{Z}_i + C$$

【共役勾配法が2次形式関数でn回のラインサーチで最適解を見つける理由】

次式を満たす n 個の n 次元ベクトル $\mathbf{Z}_1, \mathbf{Z}_2, \cdots \mathbf{Z}_n$ を考える:

$$\mathbf{Z}_i{}^t \mathbf{A} \mathbf{Z}_j = \begin{cases} const. & for \quad i = j \\ 0 & for \quad i \neq j \end{cases}$$

$\Rightarrow$ $\mathbf{Z}_1, \mathbf{Z}_2, \cdots \mathbf{Z}_n$ は共役ベクトル集合

次に、ベクトル $\mathbf{X}$ を $\mathbf{X} = \sum_{j=1}^{n} \alpha_j \mathbf{Z}_j$ と表すと、2次形式の評価式は

$$f(x_1, x_2, \cdots, x_n) = \mathbf{X}^t \mathbf{A} \mathbf{X} + \mathbf{B}^t \mathbf{X} + C$$

$$= \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \mathbf{Z}_i{}^t \mathbf{A} \mathbf{Z}_j \right) + \left(\sum_{i=1}^{n} \alpha_i \mathbf{B}^t \mathbf{Z}_i \right) + C = \sum_{i=1}^{n} \alpha_i{}^2 \mathbf{Z}_i{}^t \mathbf{A} \mathbf{Z}_i + \alpha_i \mathbf{B}^t \mathbf{Z}_i + C$$

ここで $g_i(\alpha_i) = \alpha_i{}^2 \mathbf{Z}_i{}^t \mathbf{A} \mathbf{Z}_i + \alpha_i \mathbf{B}^t \mathbf{Z}_i$ とおくと、上の式は以下のようになる:

各共役ベクトル毎にインデックス表示

【共役勾配法が2次形式関数でn回のラインサーチで最適解を見つける理由】

次式を満たす n 個の n 次元ベクトル $\mathbf{Z}_1, \mathbf{Z}_2, \cdots \mathbf{Z}_n$ を考える:

$$\mathbf{Z}_i{}^t \mathbf{A} \mathbf{Z}_j = \begin{cases} const. & for \quad i = j \\ 0 & for \quad i \neq j \end{cases}$$

⇒ $\mathbf{Z}_1, \mathbf{Z}_2, \cdots \mathbf{Z}_n$ は共役ベクトル集合

次に、ベクトル $\mathbf{X}$ を $\mathbf{X} = \sum_{j=1}^{n} \alpha_j \mathbf{Z}_j$ と表すと、2次形式の評価式は

$$f(x_1, x_2, \cdots, x_n) = \mathbf{X}^t \mathbf{A} \mathbf{X} + \mathbf{B}^t \mathbf{X} + C$$

$$= \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \mathbf{Z}_i{}^t \mathbf{A} \mathbf{Z}_j \right) + \left(\sum_{i=1}^{n} \alpha_i \mathbf{B}^t \mathbf{Z}_i \right) + C = \sum_{i=1}^{n} \alpha_i{}^2 \mathbf{Z}_i{}^t \mathbf{A} \mathbf{Z}_i + \alpha_i \mathbf{B}^t \mathbf{Z}_i + C$$

ここで $g_i(\alpha_i) = \alpha_i{}^2 \mathbf{Z}_i{}^t \mathbf{A} \mathbf{Z}_i + \alpha_i \mathbf{B}^t \mathbf{Z}_i$ とおくと、上の式は以下のようになる:

⇒ これより、関数 f はそれぞれ α_i に対する n 個の関数の代数和で表される

よって $g_1(\alpha_1), g_2(\alpha_2), \cdots g_n(\alpha_n)$ のそれぞれについて最小値を求めれば良い

探索開始点から順次 $\mathbf{Z}_1, \mathbf{Z}_2, \cdots \mathbf{Z}_n$ 方向の関数の断面について $\alpha_1, \alpha_2 \cdots \alpha_n$ の最小化を行えばn回のラインサーチで最適解を得る

【共役勾配法が2次形式関数でn回のラインサーチで最適解を見つける理由】

次式を満たす n 個の n 次元ベクトル $\mathbf{Z}_1, \mathbf{Z}_2, \cdots \mathbf{Z}_n$ を考える:

$$\mathbf{Z}_i{}^t \mathbf{A} \mathbf{Z}_j = \begin{cases} const. & for \quad i = j \\ 0 & for \quad i \neq j \end{cases}$$

⇒ $\mathbf{Z}_1, \mathbf{Z}_2, \cdots \mathbf{Z}_n$ は共役ベクトル集合

次に、ベクトル $\mathbf{X}$ を $\mathbf{X} = \sum_{j=1}^{n} \alpha_j \mathbf{Z}_j$ と表すと、2次形式の評価式は

$$f(x_1, x_2, \cdots, x_n) = \mathbf{X}^t \mathbf{A} \mathbf{X} + \mathbf{B}^t \mathbf{X} + C$$

$$= \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \mathbf{Z}_i{}^t \mathbf{A} \mathbf{Z}_j \right) + \left(\sum_{i=1}^{n} \alpha_i \mathbf{B}^t \mathbf{Z}_i \right) + C = \sum_{i=1}^{n} \alpha_i{}^2 \mathbf{Z}_i{}^t \mathbf{A} \mathbf{Z}_i + \alpha_i \mathbf{B}^t \mathbf{Z}_i + C$$

ここで $g_i(\alpha_i) = \alpha_i{}^2 \mathbf{Z}_i{}^t \mathbf{A} \mathbf{Z}_i + \alpha_i \mathbf{B}^t \mathbf{Z}_i$ とおくと、上の式は以下のようになる:

$$f(x_1, x_2, \cdots, x_n) = \sum_{i=1}^{n} g_i(\alpha_i) + C$$

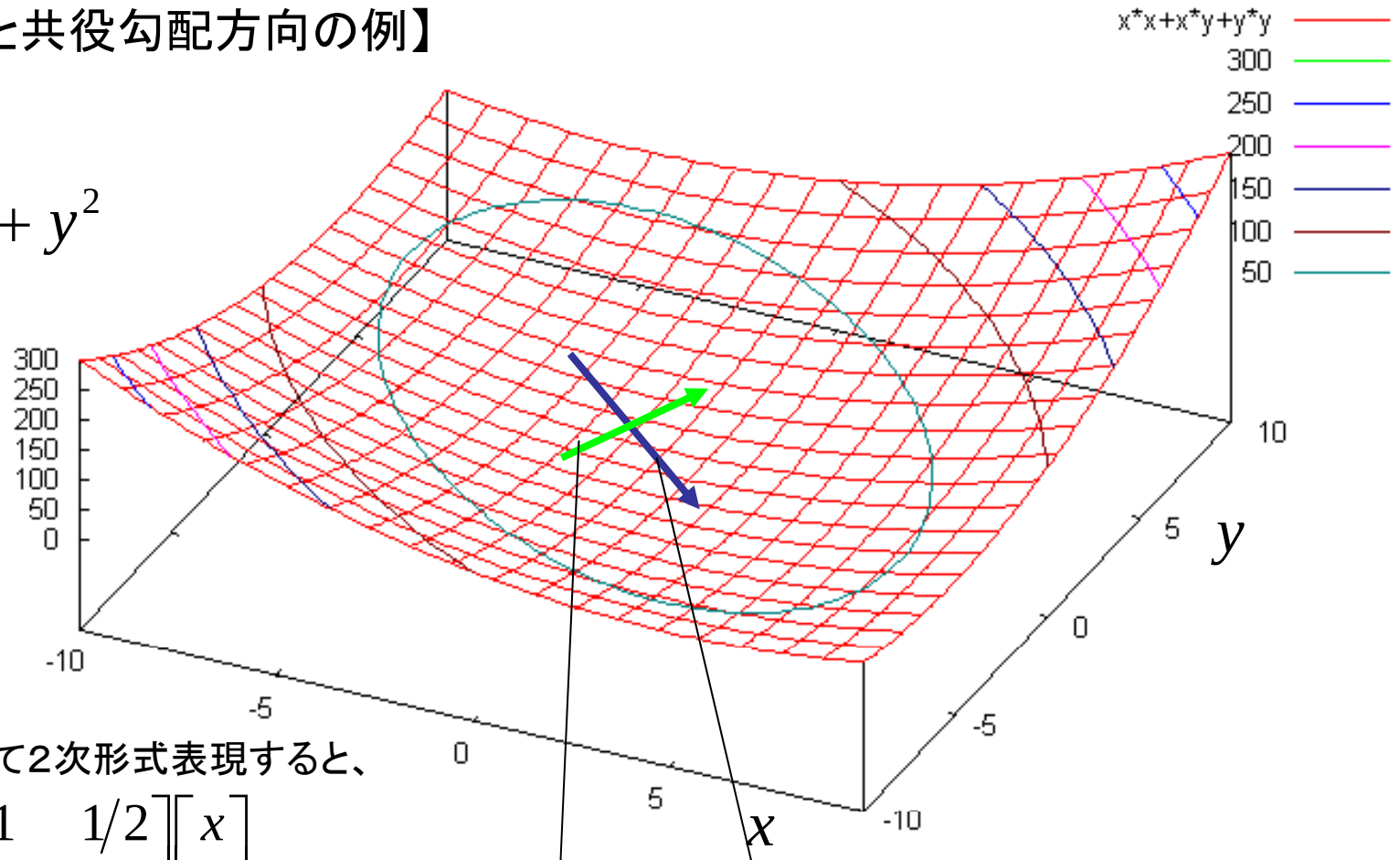
⇒ これより、関数 f はそれぞれ α_i に対する n 個の関数の代数和で表される

よって $g_1(\alpha_1), g_2(\alpha_2), \cdots g_n(\alpha_n)$ のそれぞれについて最小値を求めれば良い

探索開始点から順次 $\mathbf{Z}_1, \mathbf{Z}_2, \cdots \mathbf{Z}_n$ 方向の関数の断面について $\alpha_1, \alpha_2 \cdots \alpha_n$ の最小化を行えばn回のラインサーチで最適解を得る

【2次形式関数と共役勾配方向の例】

$$f = x^2 + xy + y^2$$



対称行列Aを用いて2次形式表現すると、

$$f = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 1/2 \\ 1/2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

このとき

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1/2 \\ 1/2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = 0$$

短軸方向

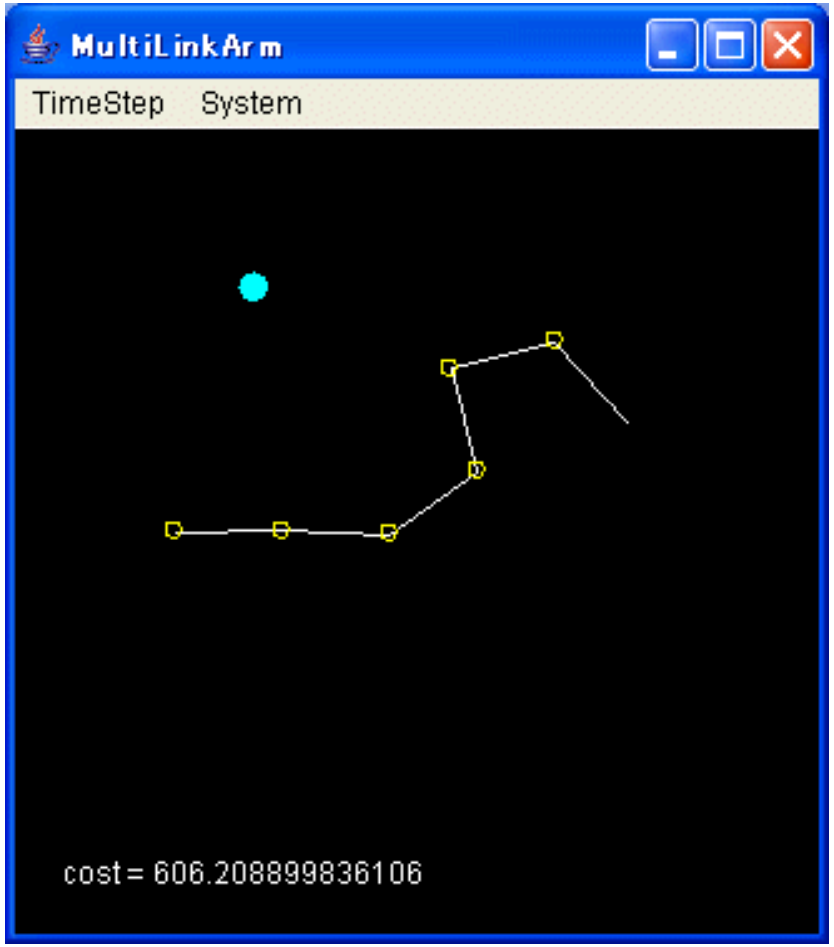
長軸方向

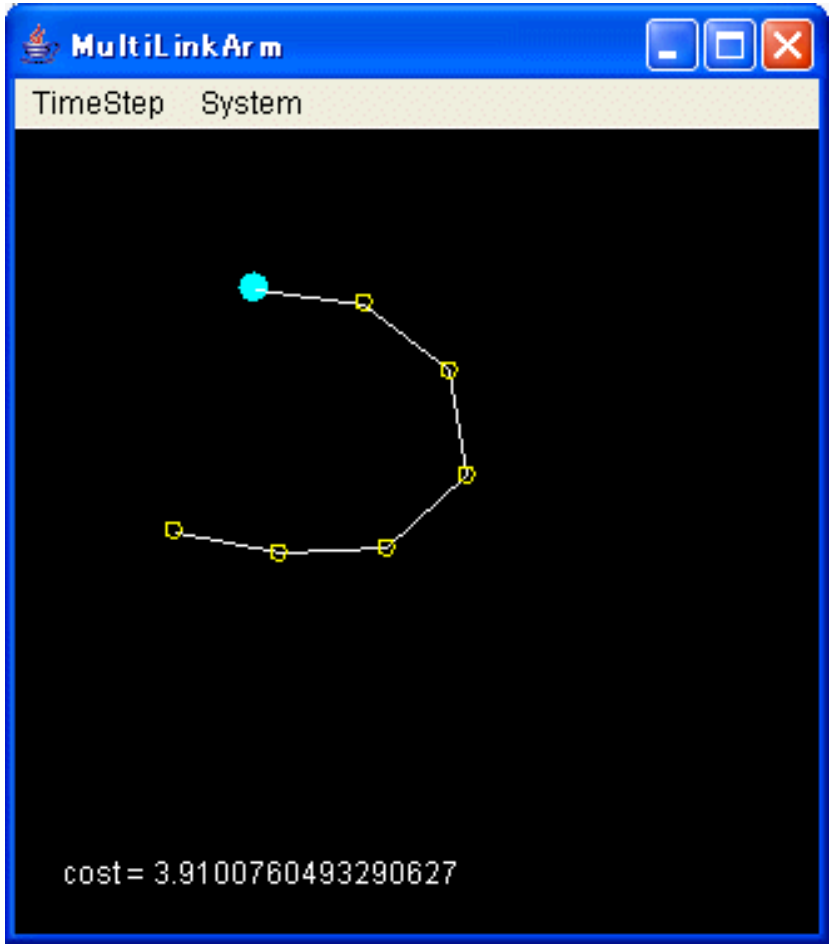
となるので（1，1）と（1、－1）が対となる共役ベクトル

【例：冗長自由度アームのリーチング問題】

●各関節の角度をパラメータとして、アーム先端がターゲット座標に一致するパラメータを求める問題。ただし、各関節はリンク同士を±90度以内で繋ぐ制約があるので、なるべく関節の可動角度中央付近の角度にすることが好ましい。

コスト関数＝（アーム先端座標とターゲット座標の偏差の2乗）
＋（各関節の可動中央角度からの偏差の2乗）





流体の数値計算における共役勾配法の利用

圧力のポアソン方程式を離散化 → 大規模な連立1次方程式 $\mathbf{Ax}=\mathbf{b}$ を解く

係数行列：隣接する領域のみの関係なので、疎な実対称正定値行列

未知量ベクトル（圧力）
要素数は数万～数百万

2次形式

$f(\mathbf{x})=$ □ とおき、これを最小化する $\mathbf{x}$ を共役勾配法で求める

「最小点」では $df(\mathbf{x})/d\mathbf{x}=\mathbf{0}$ より $\mathbf{Ax}=\mathbf{b}$ を満たす

⇩

求めようとする解そのもの

大規模な連立方程式を数値的に解く場合、逆行列を直接計算するよりも
共役勾配法のほうが、はるかに効率良く計算できる場合がある

流体の数値計算における共役勾配法の利用

圧力のポアソン方程式を離散化 → 大規模な連立1次方程式 $\mathbf{Ax}=\mathbf{b}$ を解く

係数行列：隣接する領域のみの関係なので、疎な実対称正定値行列

未知量ベクトル（圧力）
要素数は数万～数百万

2次形式

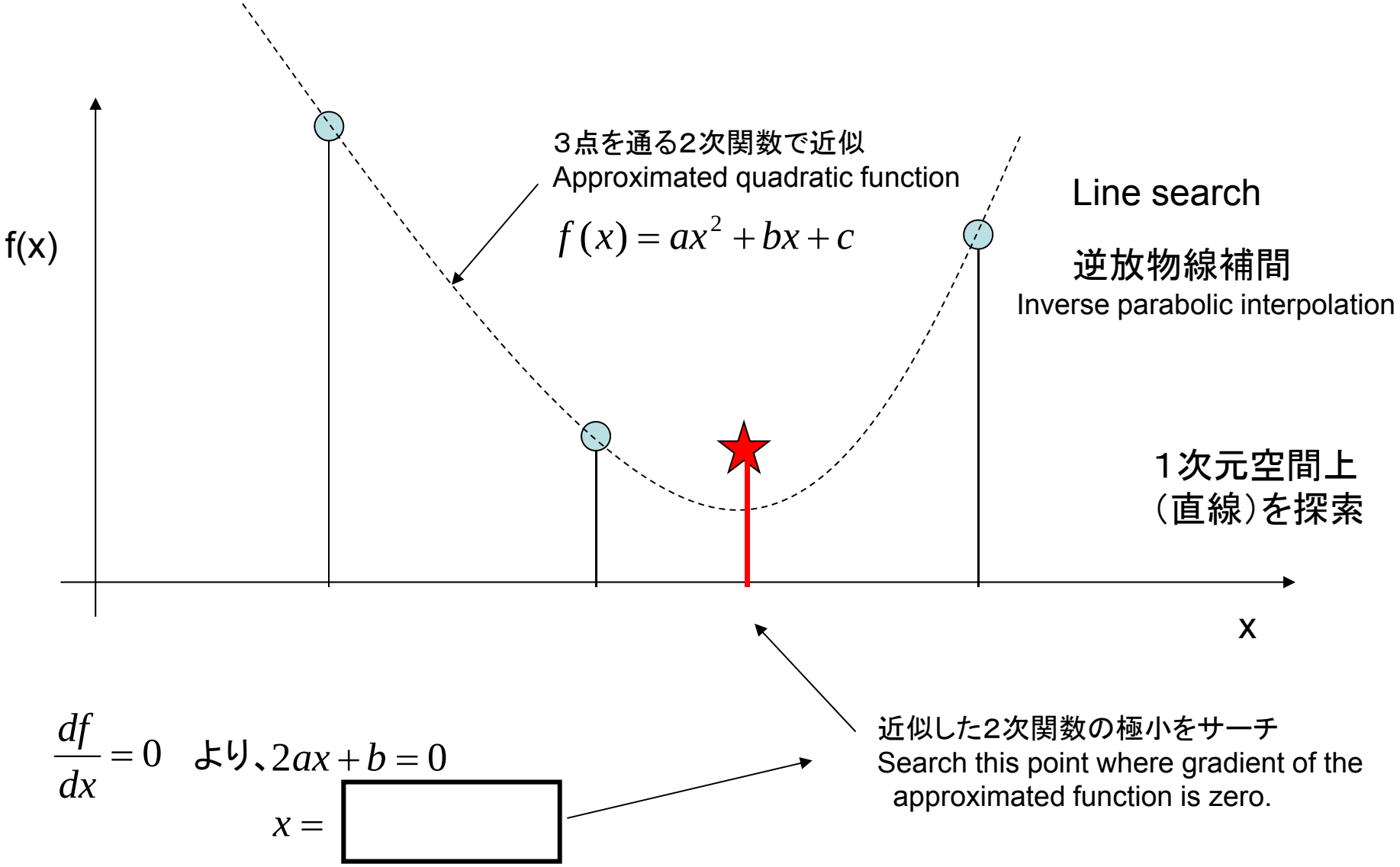
$f(\mathbf{x})=\mathbf{x}^{\mathrm{T}}\mathbf{Ax}-2\mathbf{x}^{\mathrm{T}}\mathbf{b}$ とおき、これを最小化する $\mathbf{x}$ を共役勾配法で求める

「最小点」では $df(\mathbf{x})/d\mathbf{x}=\mathbf{0}$ より $\mathbf{Ax}=\mathbf{b}$ を満たす

求めようとする解そのもの

大規模な連立方程式を数値的に解く場合、逆行列を直接計算するよりも
共役勾配法のほうが、はるかに効率良く計算できる場合がある

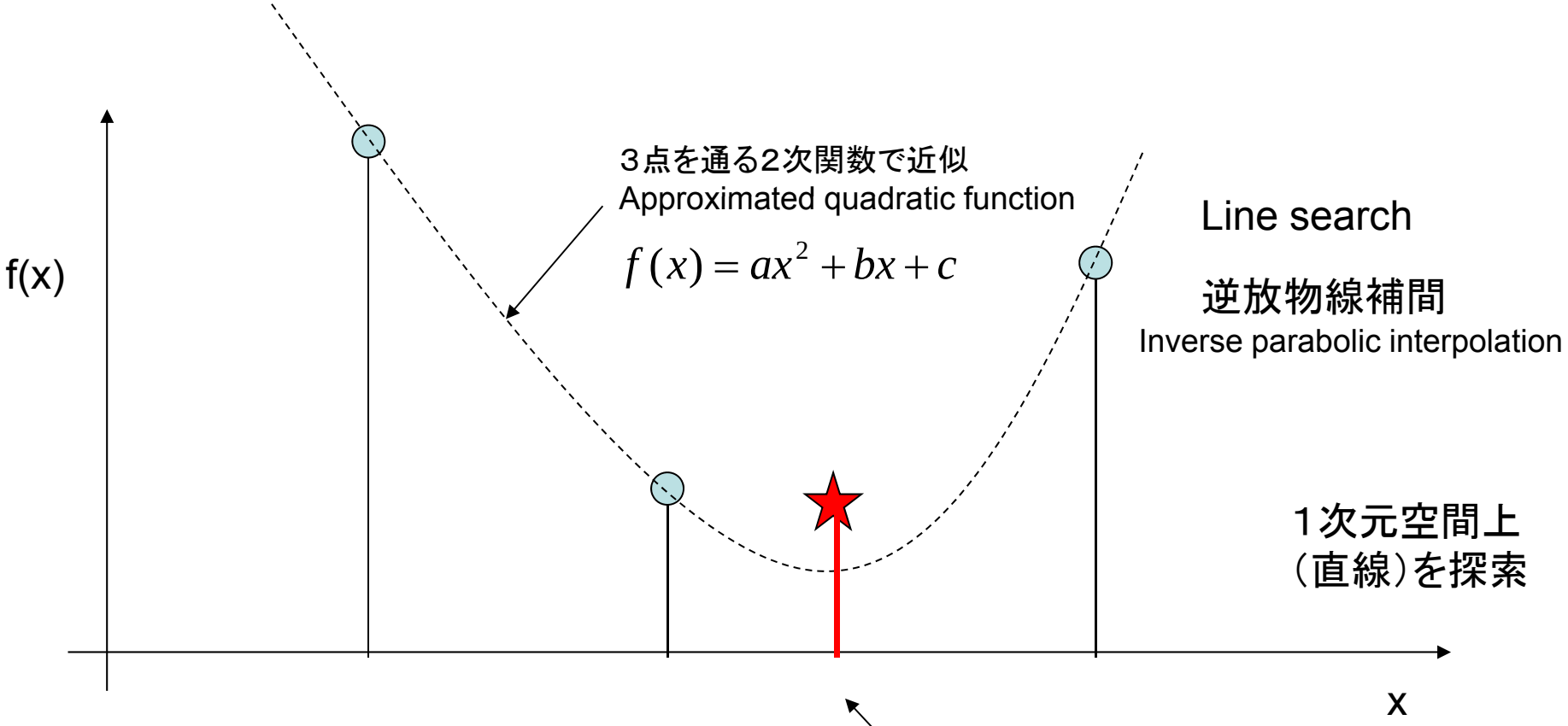
【制約のない関数最適化】ニュートン法 Newton method



$\frac{df}{dx} = 0$ より、$2ax + b = 0$

$x =$ [ ]

【制約のない関数最適化】ニュートン法 Newton method



$\frac{df}{dx} = 0$ より、$2ax + b = 0$

$$x = \boxed{(2a)^{-1}(-b)}$$

近似した2次関数の極小をサーチ
Search this point where gradient of the approximated function is zero.

Xの1次項の係数

【制約のない関数最適化】準ニュートン法

Quasi-Newton method

関数 f(x) をテイラー級数で近似
(Taylor series)

$$f(\mathbf{x}) = f(\mathbf{P}) + \sum_i \frac{\partial f}{\partial x_i} x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j} x_i x_j + \cdots$$

$$= f(\mathbf{P}) + \nabla f \,|_{\mathbf{P}} \cdot \mathbf{x} + \frac{1}{2} \mathbf{x} \cdot \mathbf{A} \cdot \mathbf{x} + \cdots$$

この行列A はf(x)のHesse行列
(Hessian matrix)
各要素はPにおける2階編導関数

$$[\mathbf{A}]_{ij} \equiv \left. \frac{\partial^2}{\partial x_i \partial x_j} \right|_{\mathbf{P}}$$

2次形式
(quadratic)

【制約のない関数最適化】準ニュートン法

Quasi-Newton method

関数 f(x) をテイラー級数で近似
(Taylor series)

$$f(\mathbf{x}) = f(\mathbf{P}) + \sum_i \frac{\partial f}{\partial x_i} x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j} x_i x_j + \cdots$$

$$= f(\mathbf{P}) + \nabla f\,|_{\mathbf{P}} \cdot \mathbf{x} + \frac{1}{2} \mathbf{x} \cdot \mathbf{A} \cdot \mathbf{x} + \cdots$$

この行列A はf(x)のHesse行列
(Hessian matrix)
各要素はPにおける2階編導関数

$$[\mathbf{A}]_{ij} \equiv \left. \frac{\partial^2}{\partial x_i \partial x_j} \right|_{\mathbf{P}}$$

2次形式
(quadratic)

極小点では勾配ゼロだから上の式を微分すると

$$\nabla f(\mathbf{x}) = \nabla f\,|_{\mathbf{P}} + \mathbf{A} \cdot \mathbf{x} = \mathbf{0}$$

これを解くと

次はこの点を探査すべし

【制約のない関数最適化】準ニュートン法

Quasi-Newton method

関数 f(x) をテイラー級数で近似
(Taylor series)

$$f(\mathbf{x}) = f(\mathbf{P}) + \sum_i \frac{\partial f}{\partial x_i} x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j} x_i x_j + \cdots$$

$$= f(\mathbf{P}) + \nabla f\,|_{\mathbf{P}} \cdot \mathbf{x} + \frac{1}{2} \mathbf{x} \cdot \mathbf{A} \cdot \mathbf{x} + \cdots$$

この行列A はf(x)のHesse行列
(Hessian matrix)
各要素はPにおける2階編導関数

$$[\mathbf{A}]_{ij} \equiv \left. \frac{\partial^2}{\partial x_i \partial x_j} \right|_{\mathbf{P}}$$

2次形式
(quadratic)

極小点では勾配ゼロだから上の式を微分すると

$$\nabla f(\mathbf{x}) = \nabla f\,|_{\mathbf{P}} + \mathbf{A} \cdot \mathbf{x} = \mathbf{0}$$

これを解くと

次はこの点を探査すべし

しかし、行列Aは未知 なので、**探索の過程から逐次的に近似**する

Since matrix A is unknown, it is estimated from the process of the optimization incrementally.

（計算には勾配の情報とラインサーチを用いる）

詳細は省略。最適化するなら共役勾配法を使え。

【制約のない関数最適化】準ニュートン法

Quasi-Newton method

関数 f(x) をテイラー級数で近似
(Taylor series)

$$f(\mathbf{x}) = f(\mathbf{P}) + \sum_i \frac{\partial f}{\partial x_i} x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j} x_i x_j + \cdots$$

$$= f(\mathbf{P}) + \nabla f\,|_{\mathbf{P}} \cdot \mathbf{x} + \frac{1}{2} \mathbf{x} \cdot \mathbf{A} \cdot \mathbf{x} + \cdots$$

この行列A はf(x)のHesse行列
(Hessian matrix)
各要素はPにおける2階編導関数

$$[\mathbf{A}]_{ij} \equiv \left. \frac{\partial^2}{\partial x_i \partial x_j} \right|_{\mathbf{P}}$$

2次形式
(quadratic)

極小点では勾配ゼロだから上の式を微分すると

$$\nabla f(\mathbf{x}) = \nabla f\,|_{\mathbf{P}} + \mathbf{A} \cdot \mathbf{x} = \mathbf{0}$$

これを解くと

$$\mathbf{x} = \mathbf{A}^{-1}(-\nabla f\,|_{\mathbf{P}})$$

次はこの点を探査すべし

しかし、行列Aは未知 なので、**探索の過程から逐次的に近似**する

Since matrix A is unknown, it is estimated from the process of the optimization incrementally.

（計算には勾配の情報とラインサーチを用いる）

詳細は省略。最適化するなら共役勾配法を使え。

勾配法の補足：

関数の厳密な勾配の計算ができない場合は、
差分近似で代用すること。

まとめ

・1次元関数の最小化（ラインサーチ）
〇黄金比分割法　△ニュートン法

・n次元関数の最小化：
勾配を用いる方法
×最急降下法　△最適勾配法　〇共役勾配法
逆放物線補間を用いる方法
〇準ニュートン法（ただし計算過程で勾配とラインサーチを用いる）

【演習問題】　2016.11.04　　学籍番号　　氏名

最適化アルゴリズムの1つである共役勾配法の処理手順について説明せよ。
ただし最適化対象のコスト関数をf(x), 最適化すべきパラメータ群をベクトルxと表す。

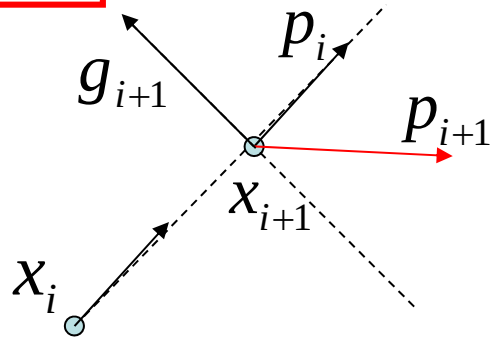
【演習問題】　　学籍番号　　氏名

最適化アルゴリズムの1つである共役勾配法の処理手順について説明せよ。
ただし最適化対象のコスト関数をf(x), 最適化すべきパラメータ群をベクトルxと表す。

x_i	i 番目の点の座標ベクトル
$g_i = \nabla f(x_1)$	点 x_i における勾配ベクトル
p_i	点 x_i からラインサーチを行う方向ベクトル

まず最初は勾配の反対方向へラインサーチ

$$p_0 = -g_0$$



x_i の次の点 x_{i+1} を、ラインサーチで見つけた最小点とする

$$x_{i+1} = x_i + \alpha_i \, p_i \quad \text{Optimum point in line search}$$

次の点 x_{i+1} での**新しい探索方向** p_{i+1} **は、前の探索方向** p_i **と共役**であるようにする。
そのため、点 x_{i+1} での勾配と前の探索方向 p_i とを結合する:

$$p_{i+1} = -g_{i+1} + \beta_{i+1} \, p_i$$

ただし、重み係数 β_{i+1} は次の式のどちらかで与える:

$$\beta_{i+1} = \frac{g_{i+1}^T g_{i+1}}{g_i^T g_i} = \frac{\|g_{i+1}\|^2}{\|g_i\|^2}$$ （Fletcher-Reeves法）

（Polak-Ribiere法）

$$\beta_{i+1} = \frac{(g_{i+1} - g_i)^T g_{i+1}}{g_i^T g_i}$$