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）

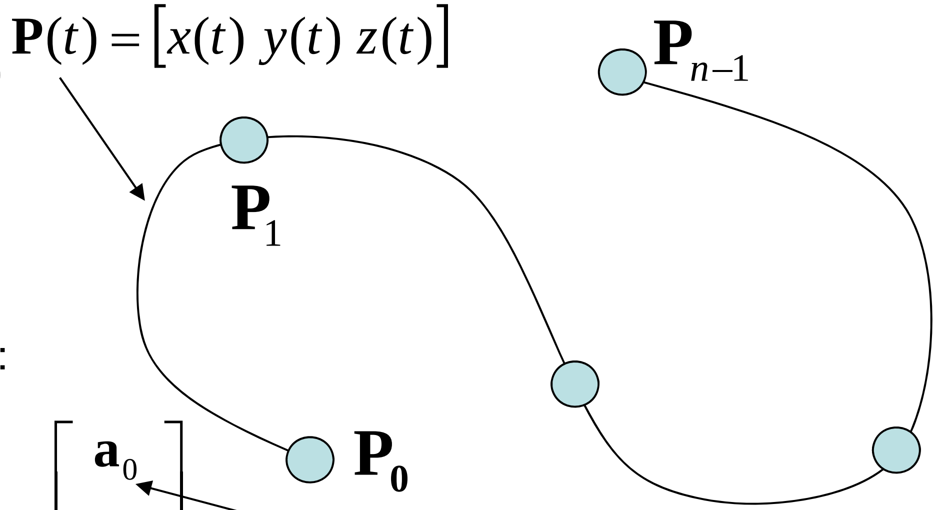
(3) 補間・関数近似と計算幾何学

場所： 船1講義室

授業の資料等は

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

多項式曲線（2～3次元）



曲線を通過させたい点列 $\mathbf{P}_i$ に対して、
曲線の長さに関連したパラメータ t
を与え、曲線を以下の多項式で表現する:

$$\mathbf{P}(t) = \begin{bmatrix} x(t) & y(t) & z(t) \end{bmatrix}$$

$$= \boxed{} = \begin{bmatrix} 1 & t & t^2 & \cdots & t^j & \cdots & t^{m-1} \end{bmatrix} \begin{bmatrix} \mathbf{a}_0 \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_j \\ \vdots \\ \mathbf{a}_{m-1} \end{bmatrix}$$

係数ベクトル
x,y,zにそれぞれ
対応した値を持つ

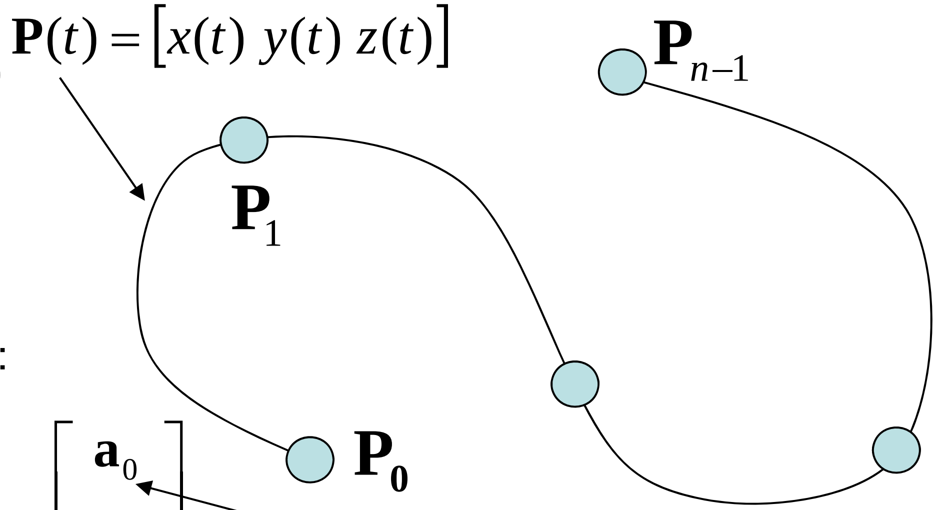
これを決めれば、曲線上のt
で指定される(x,y,z)座標が
与えられる（曲線補間）

通過点列 $\mathbf{P}_i$ に対して、係数ベクトルは
以下の方程式を満たす:

$$\boxed{} \begin{bmatrix} \mathbf{a}_0 \\ \vdots \\ \mathbf{a}_j \\ \vdots \\ \mathbf{a}_{m-1} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_0 \\ \vdots \\ \mathbf{P}_i \\ \vdots \\ \mathbf{P}_{n-1} \end{bmatrix}$$

よって、m=n ならば、
係数行列は左辺の行列の
逆行列を計算して行列Pを
乗ずることで得る。

多項式曲線（2～3次元）



曲線を通過させたい点列 $\mathbf{P}_i$ に対して、
曲線の長さに関連したパラメータ　t
を与え、**曲線を以下の多項式で表現**する：

$$\mathbf{P}(t)=[x(t)\ y(t)\ z(t)]$$

$$=\boxed{\sum_{j=0}^{m-1}\mathbf{a}_j t^j}=\begin{bmatrix}1 & t & t^2 & \cdots & t^j & \cdots & t^{m-1}\end{bmatrix}\begin{bmatrix}\mathbf{a}_0\\ \mathbf{a}_1\\ \mathbf{a}_2\\ \vdots\\ \mathbf{a}_j\\ \vdots\\ \mathbf{a}_{m-1}\end{bmatrix}$$

係数ベクトル
x,y,zにそれぞれ
対応した値を持つ

これを決めれば、曲線上のt
で指定される（x,y,z）座標が
与えられる（**曲線補間**）

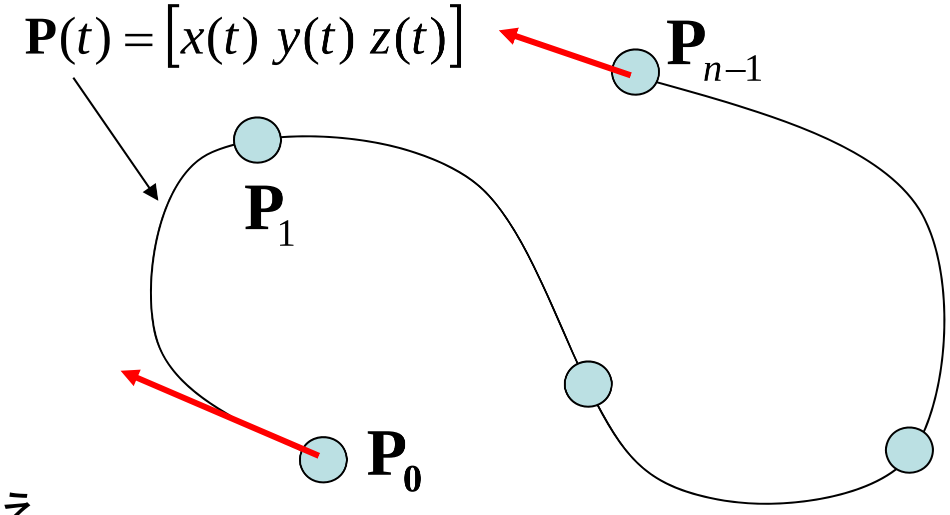
通過点列 $\mathbf{P}_i$ に対して、係数ベクトルは
以下の方程式を満たす：

$$\boxed{\begin{bmatrix}1 & t_0 & \cdots & t_0^{\ j} & \cdots & t_0^{\ m-1}\\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots\\ 1 & t_i & \cdots & t_i^{\ j} & \cdots & t_i^{\ m-1}\\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots\\ 1 & t_{n-1} & \cdots & t_{n-1}^{\ j} & \cdots & t_{n-1}^{\ m-1}\end{bmatrix}}\begin{bmatrix}\mathbf{a}_0\\ \vdots\\ \mathbf{a}_j\\ \vdots\\ \mathbf{a}_{m-1}\end{bmatrix}=\begin{bmatrix}\mathbf{P}_0\\ \vdots\\ \mathbf{P}_i\\ \vdots\\ \mathbf{P}_{n-1}\end{bmatrix}$$

ちなみに**m＜nの場合**は逆行列は計算できないが
疑似逆行列（多重回帰参照）で無理やり解ける
この場合は**点Piを通らず**最小二乗法になる

よって、m=n ならば、
係数行列は左辺の行列の
逆行列を計算して行列Pを
乗ずることで得る。

多項式曲線



さらに、曲線の両端末での
接線ベクトルの方向を指定する場合：

$$\dot{\mathbf{P}}(t) = \frac{d}{dt}\mathbf{P}(t) = \boxed{}$$

より、

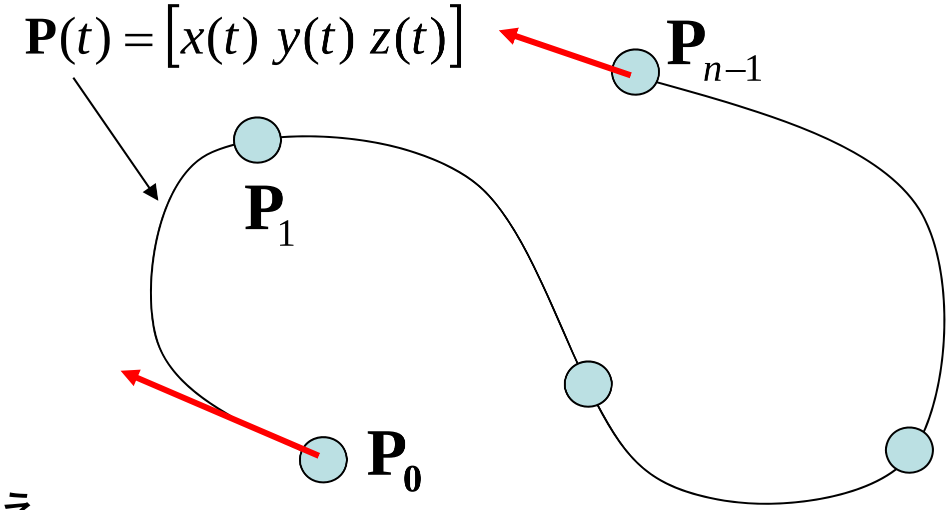
係数ベクトルの方程式の拘束条件が2つ増え、
以下の方程式が成り立つ：

$$\begin{bmatrix} 1 & t_0 & \cdots & t_0^{\ j} & \cdots & t_0^{\ m-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & t_i & \cdots & t_i^{\ j} & \cdots & t_i^{\ m-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & t_{n-1} & \cdots & t_{n-1}^{\ j} & \cdots & t_{n-1}^{\ m-1} \\ & & & & & \\ & & & & & \end{bmatrix} \begin{bmatrix} \mathbf{a}_0 \\ \vdots \\ \mathbf{a}_j \\ \vdots \\ \mathbf{a}_{m-1} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_0 \\ \vdots \\ \mathbf{P}_i \\ \vdots \\ \mathbf{P}_{n-1} \\ \\ \\ \end{bmatrix}$$

端部の
接線の条件

よって、m=n＋2 ならば、係数行列は左辺の行列の
逆行列を計算して行列Pを乗ずることで得る。

多項式曲線



さらに、曲線の両端末での
接線ベクトルの方向を指定する場合：

$$\dot{\mathbf{P}}(t) = \frac{d}{dt}\mathbf{P}(t) = \boxed{\sum_{j=0}^{m-1} j\mathbf{a}_j t^{j-1}}$$

より、

係数ベクトルの方程式の拘束条件が2つ増え、
以下の方程式が成り立つ：

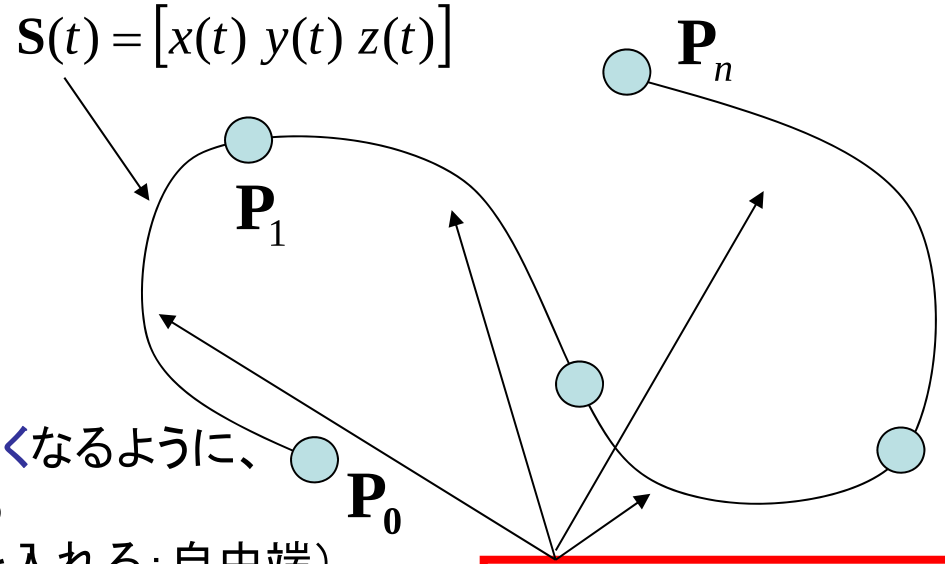
$$\begin{bmatrix} 1 & t_0 & \cdots & t_0^{\,j} & \cdots & t_0^{\,m-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & t_i & \cdots & t_i^{\,j} & \cdots & t_i^{\,m-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & t_{n-1} & \cdots & t_{n-1}^{\,j} & \cdots & t_{n-1}^{\,m-1} \\ 0 & 1 & \cdots & j\,t_0^{\,j-1} & \cdots & (m-1)\,t_0^{\,m-2} \\ 0 & 1 & \cdots & j\,t_{n-1}^{\,j-1} & \cdots & (m-1)\,t_{n-1}^{\,m-2} \end{bmatrix} \begin{bmatrix} \mathbf{a}_0 \\ \vdots \\ \mathbf{a}_j \\ \vdots \\ \mathbf{a}_{m-1} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_0 \\ \vdots \\ \mathbf{P}_i \\ \vdots \\ \mathbf{P}_{n-1} \\ \dot{\mathbf{P}}_0 \\ \dot{\mathbf{P}}_{n-1} \end{bmatrix}$$

端部の
接線の条件

よって、m=n＋2 ならば、係数行列は左辺の行列の逆行列を計算して行列Pを乗ずることで得る。

スプライン曲線

自由端3次スプライン曲線



1）隣り合う2つの通過点間を
　３次多項式曲線で表現

2）**通過点における接線方向ベクトルが等しく**なるように、
隣接する3次多項式曲線の端点を設定する
（ただし端点では2次微分＝0の拘束条件を入れる: 自由端）

それぞれの区間で
別々の3次多項式曲線
を求める

未知数4コ

３次多項式曲線

$$\mathbf{S}(t) = [x(t)\ y(t)\ z(t)] = \mathbf{a} + \mathbf{b}t + \mathbf{c}t^2 + \mathbf{d}t^3$$

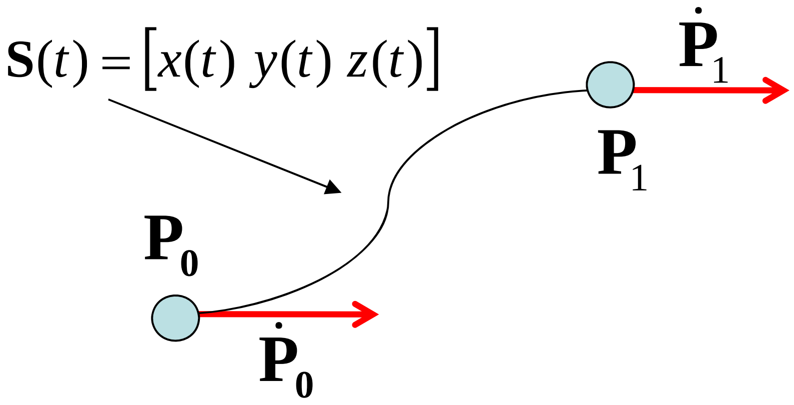
端点の座標　　　＝　2つの拘束条件

端点の接線方向　＝　2つの拘束条件

4つの拘束条件より、
4つの未知数を求める

隣り合う区間での3次多項式曲線の接線が等しいとした連立方程式を立てて解く
→　３重対角行列の逆行列を解く問題へ帰着

スプライン曲線

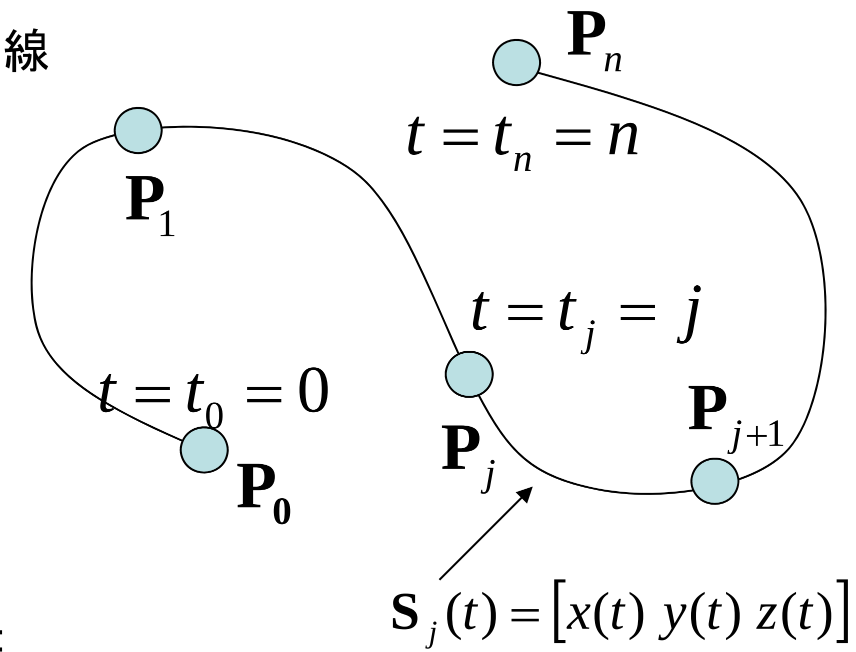


P0からP1までの区間で媒介変数　t　が0～1の値をとる場合、
3次スプライン曲線は以下のように簡単に与えられる:

$$\mathbf{S}(t) = [x(t)\ y(t)\ z(t)] = \mathbf{a} + \mathbf{b}t + \mathbf{c}t^2 + \mathbf{d}t^3$$

$$= \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \dot{\mathbf{P}}_0 \\ \dot{\mathbf{P}}_1 \end{bmatrix}$$

スプライン曲線 自由端3次スプライン曲線



各区間で3次スプライン曲線を以下のように定義する:

$$\mathbf{S}_j(t) = [x(t)\ y(t)\ z(t)]$$

$$= \mathbf{a}_j + \mathbf{b}_j(t - t_j) + \mathbf{c}_j(t - t_j)^2 + \mathbf{d}_j(t - t_j)^3$$

各区間では0〜1

上記のスプライン曲線に以下の拘束条件を与える:

$$\begin{cases} \mathbf{S}_j(t = j) = \mathbf{P}_j = [x_j \quad y_j \quad z_j] \\ \mathbf{S}_j(t = j+1) = \mathbf{S}_{j+1}(j+1) = \mathbf{P}_{j+1} = [x_{j+1} \quad y_{j+1} \quad z_{j+1}] \\ \dot{\mathbf{S}}_j(t = j+1) = \dot{\mathbf{S}}_{j+1}(j+1) \\ \ddot{\mathbf{S}}_j(j+1) = \ddot{\mathbf{S}}_{j+1}(j+1) \\ \ddot{\mathbf{S}}_0(0) = \ddot{\mathbf{S}}_{n-1}(n) = 0 \end{cases}$$

通過する点の座標

隣接する曲線の傾きと2次微分が同じ

曲線の両端の2次微分＝0

これらの方程式を係数について解くと次スライドのようになる:

参考資料:「簡略化した3次スプライン曲線の生成方法」
http://www5d.biglobe.ne.jp/~stssk/maze/spline.html

$$\mathbf{a}_j = \mathbf{P}_j$$

$$\mathbf{b}_j = \mathbf{a}_{j+1} - \mathbf{a}_j - \frac{\left(\mathbf{c}_{j+1} + 2\mathbf{c}_j\right)}{3}$$

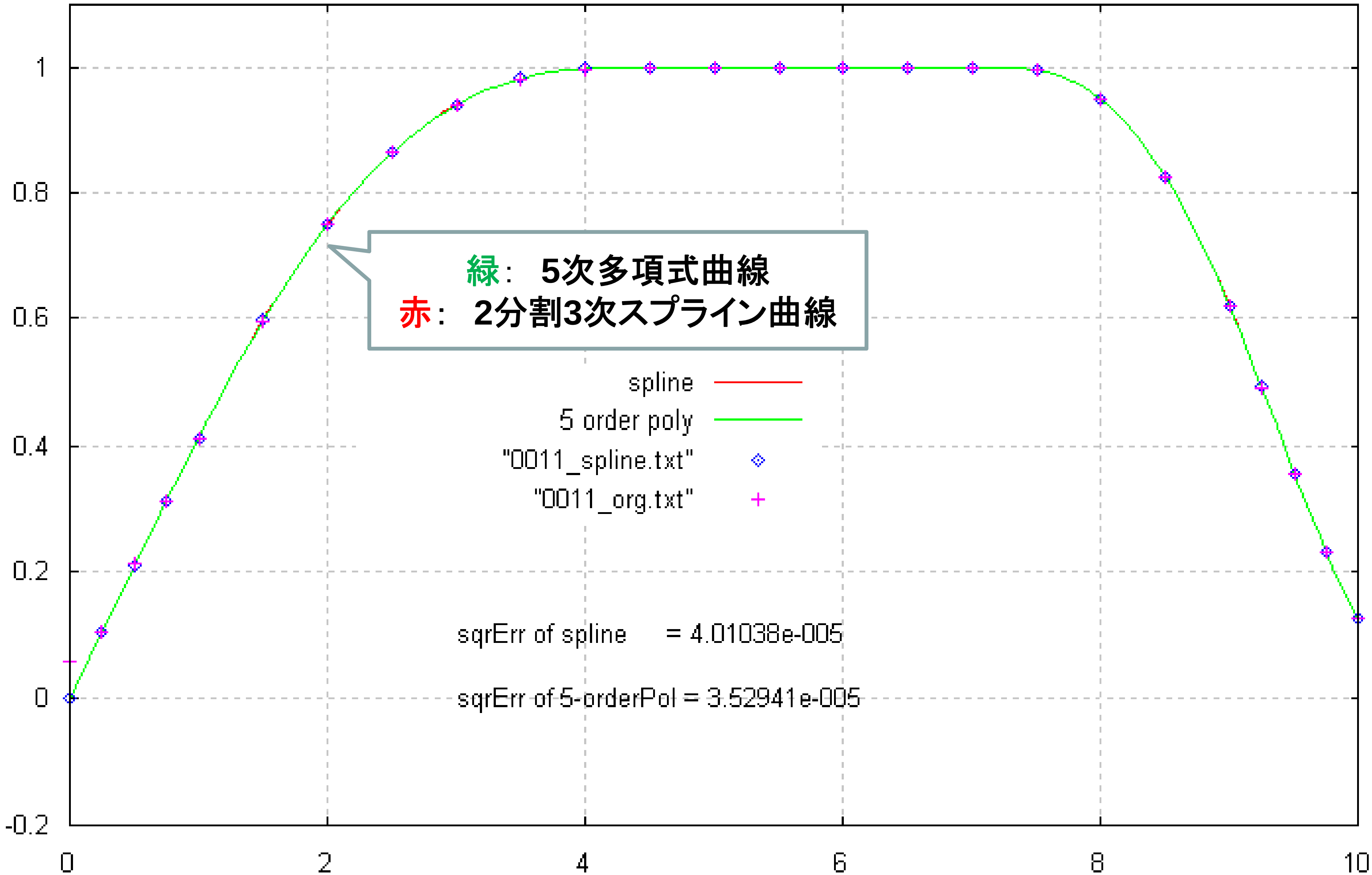
$$\mathbf{d}_j = \frac{\left(\mathbf{c}_{j+1} + 2\mathbf{c}_j\right)}{3}$$

$\mathbf{c}_j$ については、以下の連立方程式を解く:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \\ \vdots \\ \mathbf{c}_{n-2} \\ \mathbf{c}_{n-1} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 3\left(\mathbf{a}_2 - 2\mathbf{a}_1 + \mathbf{a}_0\right) \\ 3\left(\mathbf{a}_3 - 2\mathbf{a}_2 + \mathbf{a}_1\right) \\ 3\left(\mathbf{a}_4 - 2\mathbf{a}_3 + \mathbf{a}_2\right) \\ \vdots \\ 3\left(\mathbf{a}_{n-1} - 2\mathbf{a}_{n-2} + \mathbf{a}_{n-3}\right) \\ 3\left(\mathbf{a}_n - 2\mathbf{a}_{n-1} + \mathbf{a}_{n-2}\right) \\ \mathbf{0} \end{bmatrix}$$

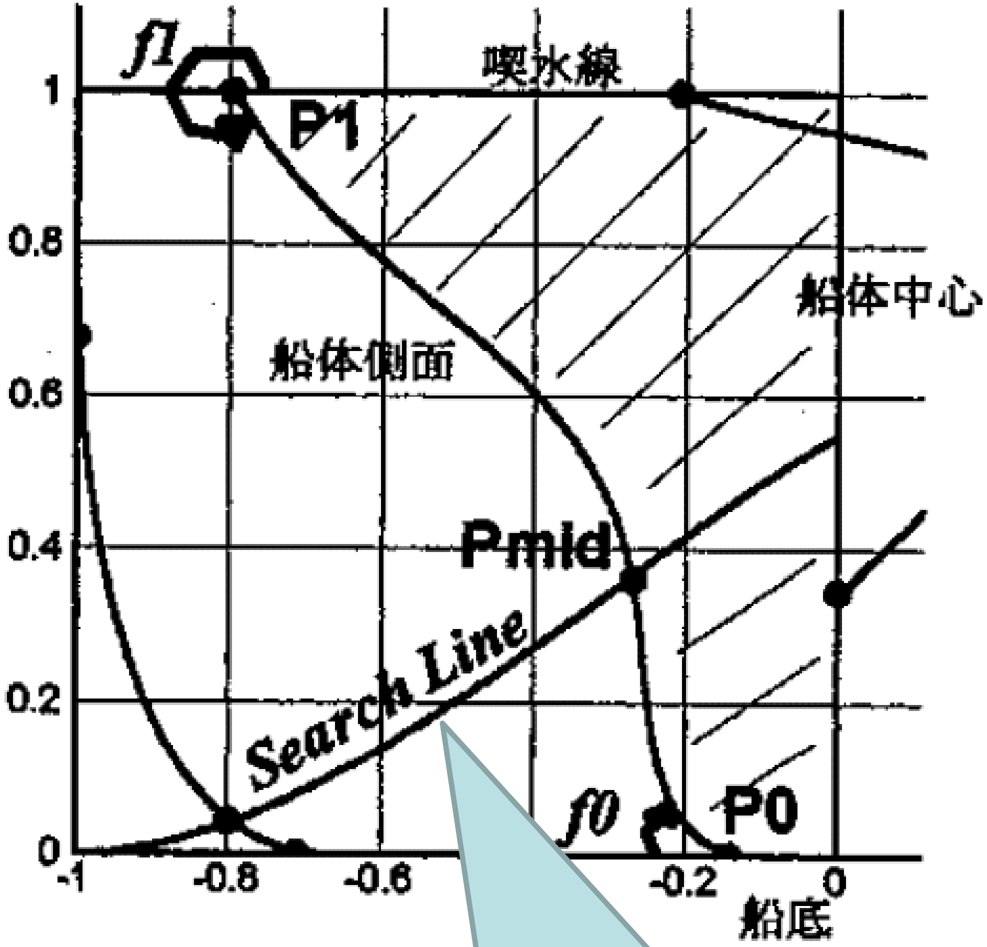
これで全ての区間のスプライン曲線の係数が求められる。

多項式曲線・スプライン曲線の使用例：Cpカーブの表現



スプライン曲線の使用例：線図創生法

船型設計システムについて―線図創生―
SRC News No.62, Jan., pp.8--9, 2005,
造船技術センター(SRC)



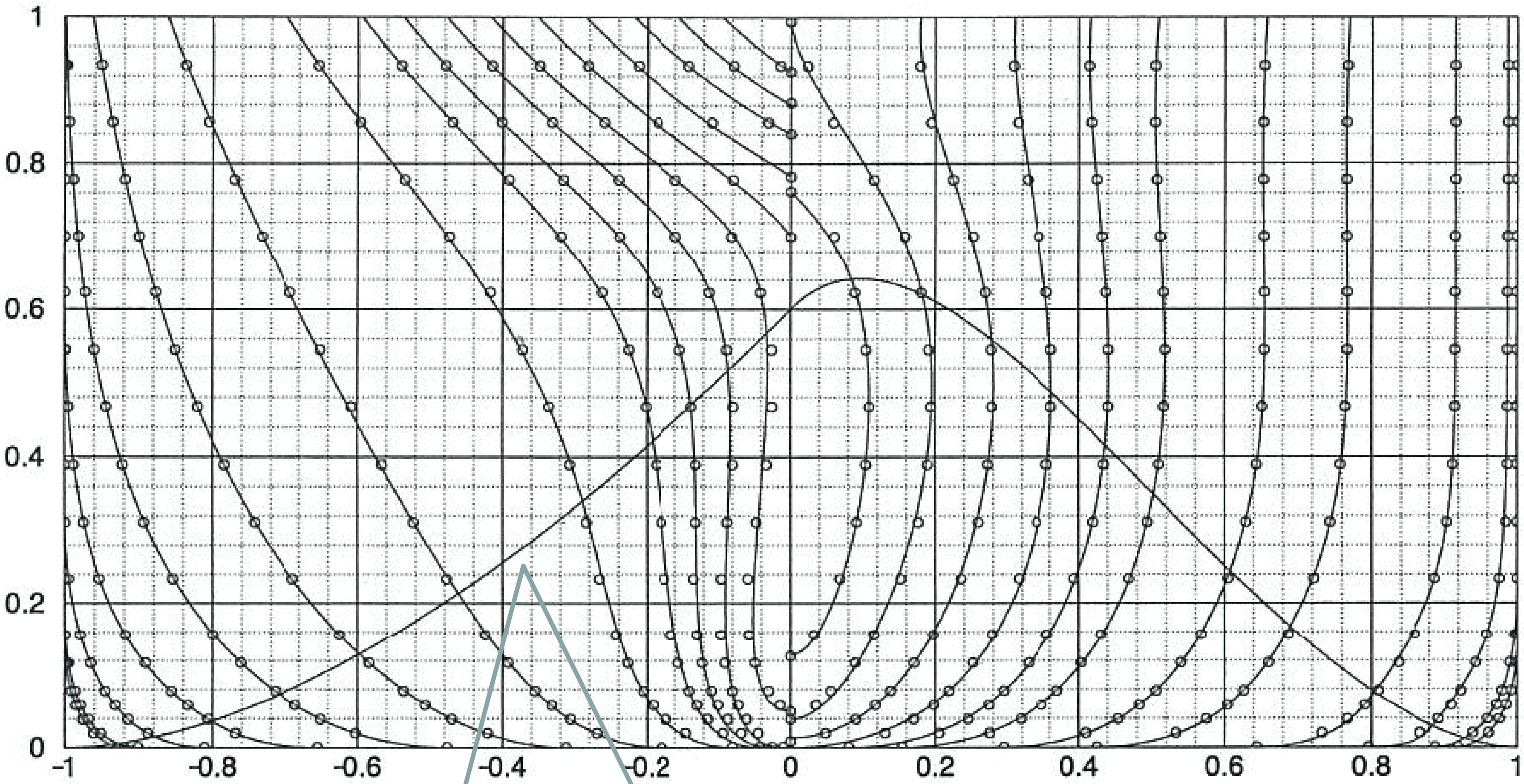
これも**パラメトリック3次スプライン曲線**
パラメータ　t　(0＜t＜1)で位置を指定

- 肋骨線形状をサーチライン上に分割点を持つ2分割**パラメトリック3次スプライン曲線**で表現する。
- 肋骨線の両端の点の角度と曲率を与える（隣り合う肋骨線同士での値がなめらかに変化するよう配慮）
- 端点P1, P2の座標（サイドタンジェンシャル・フラットボトム）を適宜与える
- 船体断面の面積が該当位置のCpカーブの値×Cmと一致するよう、サーチライン上でスプライン曲線の分割点を探索する←**ラインサーチ**

パラメトリック3次スプライン曲線：

$$\begin{cases} x = a_x t^3 + b_x t^2 + c_x t + d_x \\ y = a_y t^3 + b_y t^2 + c_y t + d_y \end{cases}$$

線図創生法で生成された線図の一例

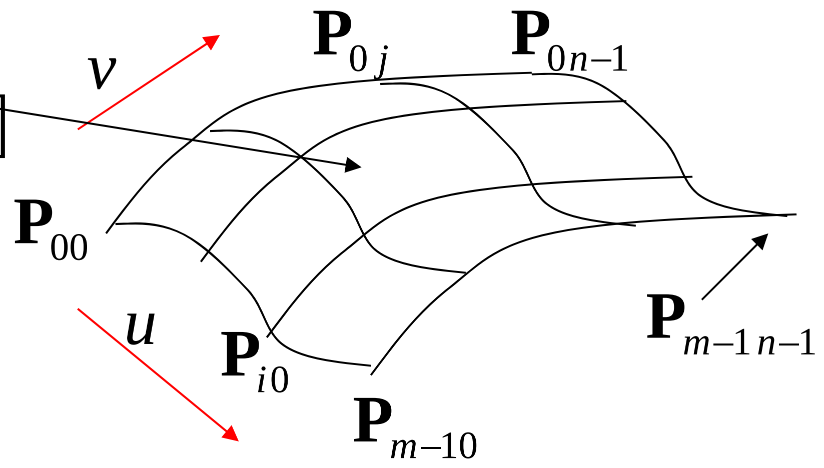


サーチラインは船体の前部と後部で
別のものを用いるが、並行部付近
（船体下部）では同じ角度と曲率にする

多項式曲面

（3次元）

$$\mathbf{P}(u,v) = \left[x(u,v)\ y(u,v)\ z(u,v) \right]$$



曲面を通過させたい点列 $\mathbf{P}_{ij}$ に対して、曲線の長さに関連したパラメータ u とv を与え、**曲面を以下の多項式で表現**する:

$$\mathbf{P}(u,v) = \sum_{i=0}^{m-1}\sum_{j=0}^{m-1} \mathbf{a}_{ij} u^i v^j$$

$$= \begin{bmatrix} 1 & \cdots & u^i & \cdots & u^{m-1} \end{bmatrix} \begin{bmatrix} \mathbf{a}_{00} & \cdots & \mathbf{a}_{0j} & \cdots & \mathbf{a}_{0\,m-1} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{a}_{i0} & \cdots & \mathbf{a}_{ij} & \cdots & \mathbf{a}_{i\,m-1} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{a}_{m-1\,0} & \cdots & \mathbf{a}_{m-1\,j} & \cdots & \mathbf{a}_{m-1\,m-1} \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ v^j \\ \vdots \\ v^{m-1} \end{bmatrix} = \mathbf{U}\mathbf{A}\mathbf{V}^{\mathrm{T}}$$

曲面係数マトリクスA

これを決めれば、曲面上の(u,v)で指定されるx,y,z座標が与えられる（**曲面補間**）

→ 曲面の通過点列から曲面係数マトリクスを求めるには？

多項式曲面

曲面通過点列 $\mathbf{P}_{i\,j}$ についての行列を定義:

$$
\mathbf{P} = \begin{array}{cc} & v \longrightarrow \\ u \downarrow & \begin{bmatrix} \mathbf{P}_{0\,0} & \cdots & \mathbf{P}_{0\,j} & \cdots & \mathbf{P}_{0\,n-1} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{P}_{i\,0} & \cdots & \mathbf{P}_{i\,j} & \cdots & \mathbf{P}_{i\,n-1} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{P}_{n-1\,0} & \cdots & \mathbf{P}_{n-1\,j} & \cdots & \mathbf{P}_{n-1\,n-1} \end{bmatrix} \end{array}
$$

各通過点に対応する u, v ベクトルの行列を定義:

$$
U = \begin{bmatrix} U_0 \\ \vdots \\ U_i \\ \vdots \\ U_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & u_0 & \cdots & u_0{}^j & \cdots & u_0{}^{m-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & u_i & \cdots & u_i{}^j & \cdots & u_i{}^{m-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & u_{n-1} & \cdots & u_{n-1}{}^j & \cdots & u_{n-1}{}^{m-1} \end{bmatrix}
$$

$$
V = \begin{bmatrix} V_0 \\ \vdots \\ V_i \\ \vdots \\ V_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & v_0 & \cdots & v_0{}^j & \cdots & v_0{}^{m-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & v_i & \cdots & v_i{}^j & \cdots & v_i{}^{m-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & v_{n-1} & \cdots & v_{n-1}{}^j & \cdots & v_{n-1}{}^{m-1} \end{bmatrix}
$$

多項式曲面はこれらの点列を
通るので、以下の式が成り立つ:

曲面係数マトリクス

曲線の場合同様、端部の拘束条件を
入れることもできる

★CADで船殻の曲面を設計する場合には
(Non-Uniform Rational B-Spline)
が使われる

多項式曲面

曲面通過点列$\mathbf{P}_{i\,j}$についての行列を定義:

$$
\mathbf{P} = \begin{bmatrix}
\mathbf{P}_{0\,0} & \cdots & \mathbf{P}_{0\,j} & \cdots & \mathbf{P}_{0\,n-1} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\mathbf{P}_{i\,0} & \cdots & \mathbf{P}_{i\,j} & \cdots & \mathbf{P}_{i\,n-1} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\mathbf{P}_{n-1\,0} & \cdots & \mathbf{P}_{n-1\,j} & \cdots & \mathbf{P}_{n-1\,n-1}
\end{bmatrix}
$$

各通過点に対応するu, v ベクトルの行列を定義:

$$
U = \begin{bmatrix} U_0 \\ \vdots \\ U_i \\ \vdots \\ U_{n-1} \end{bmatrix} = \begin{bmatrix}
1 & u_0 & \cdots & u_0{}^j & \cdots & u_0{}^{m-1} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
1 & u_i & \cdots & u_i{}^j & \cdots & u_i{}^{m-1} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
1 & u_{n-1} & \cdots & u_{n-1}{}^j & \cdots & u_{n-1}{}^{m-1}
\end{bmatrix}
$$

$$
V = \begin{bmatrix} V_0 \\ \vdots \\ V_i \\ \vdots \\ V_{n-1} \end{bmatrix} = \begin{bmatrix}
1 & v_0 & \cdots & v_0{}^j & \cdots & v_0{}^{m-1} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
1 & v_i & \cdots & v_i{}^j & \cdots & v_i{}^{m-1} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
1 & v_{n-1} & \cdots & v_{n-1}{}^j & \cdots & v_{n-1}{}^{m-1}
\end{bmatrix}
$$

多項式曲面はこれらの点列を
通るので、以下の式が成り立つ:

$$\boxed{\mathbf{P} = U\mathbf{A}V^T}$$

曲面係数マトリクス $\boxed{\mathbf{A} = U^{-1}\mathbf{P}(V^T)^{-1}}$

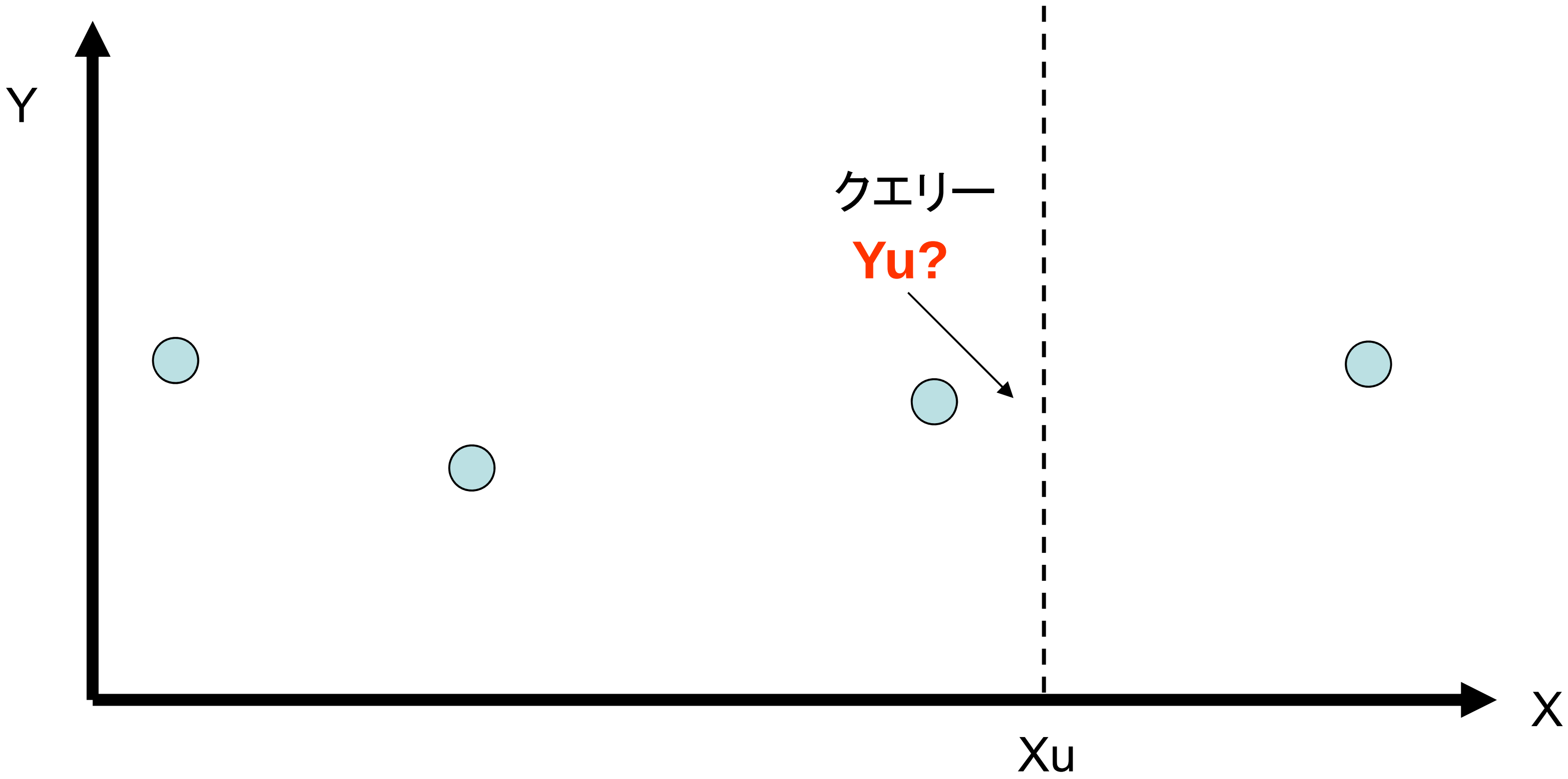
曲線の場合同様、端部の拘束条件を
入れることもできる

★CADで船殻の曲面を設計する場合には
NURBS曲面 (Non-Uniform Rational B-Spline)
が使われる

【関数近似の問題設定】

多次元空間X
データ群は**まばらで少ない**

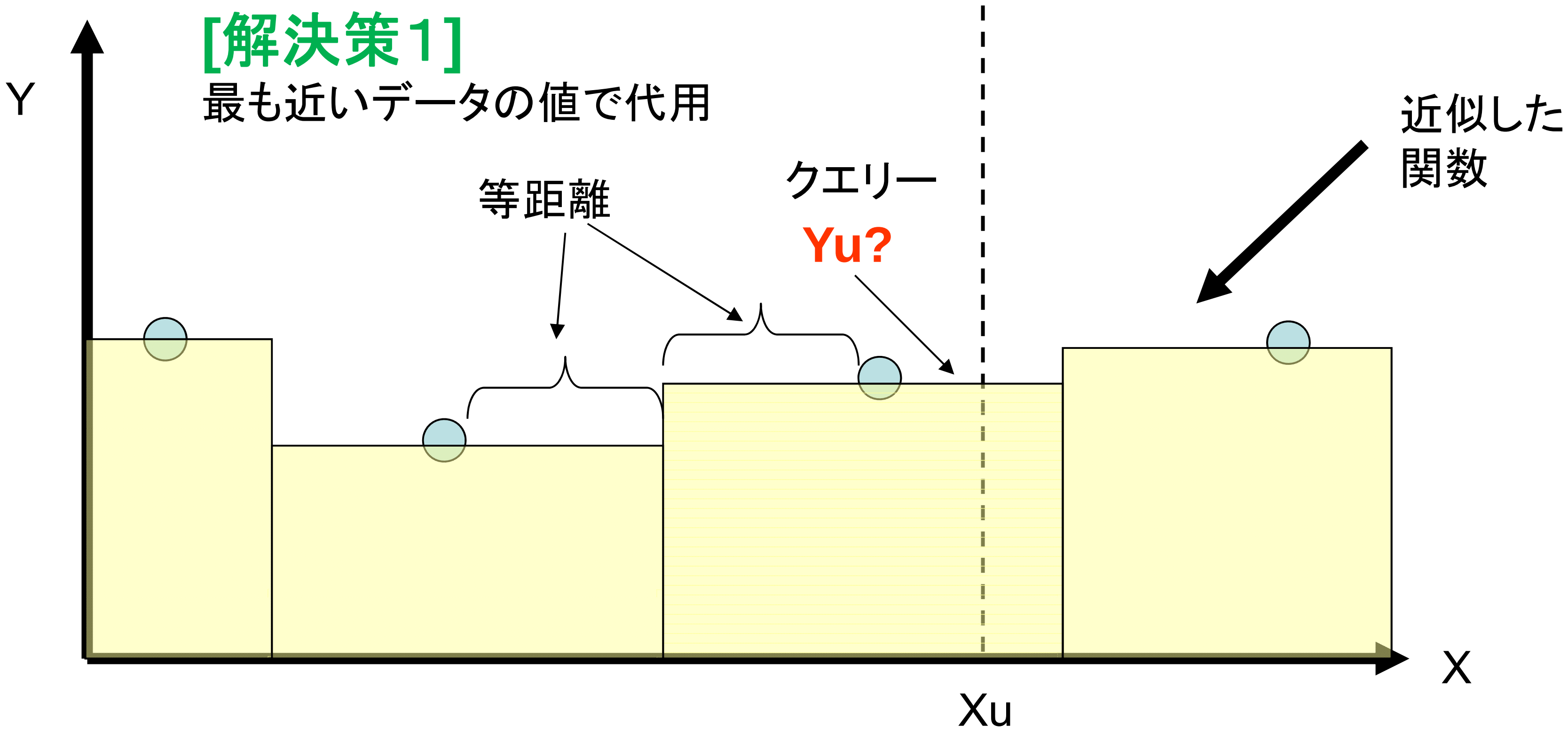
データ群 (X1,Y1), (X2,Y2), … (Xn,Yn) が与えられているもとで、
未知なる Xu についてのYuの値を求める
ただし X1～Xn, Y1～Yn, Xu,Yu は全て有界な値（スカラー or ベクトル量）



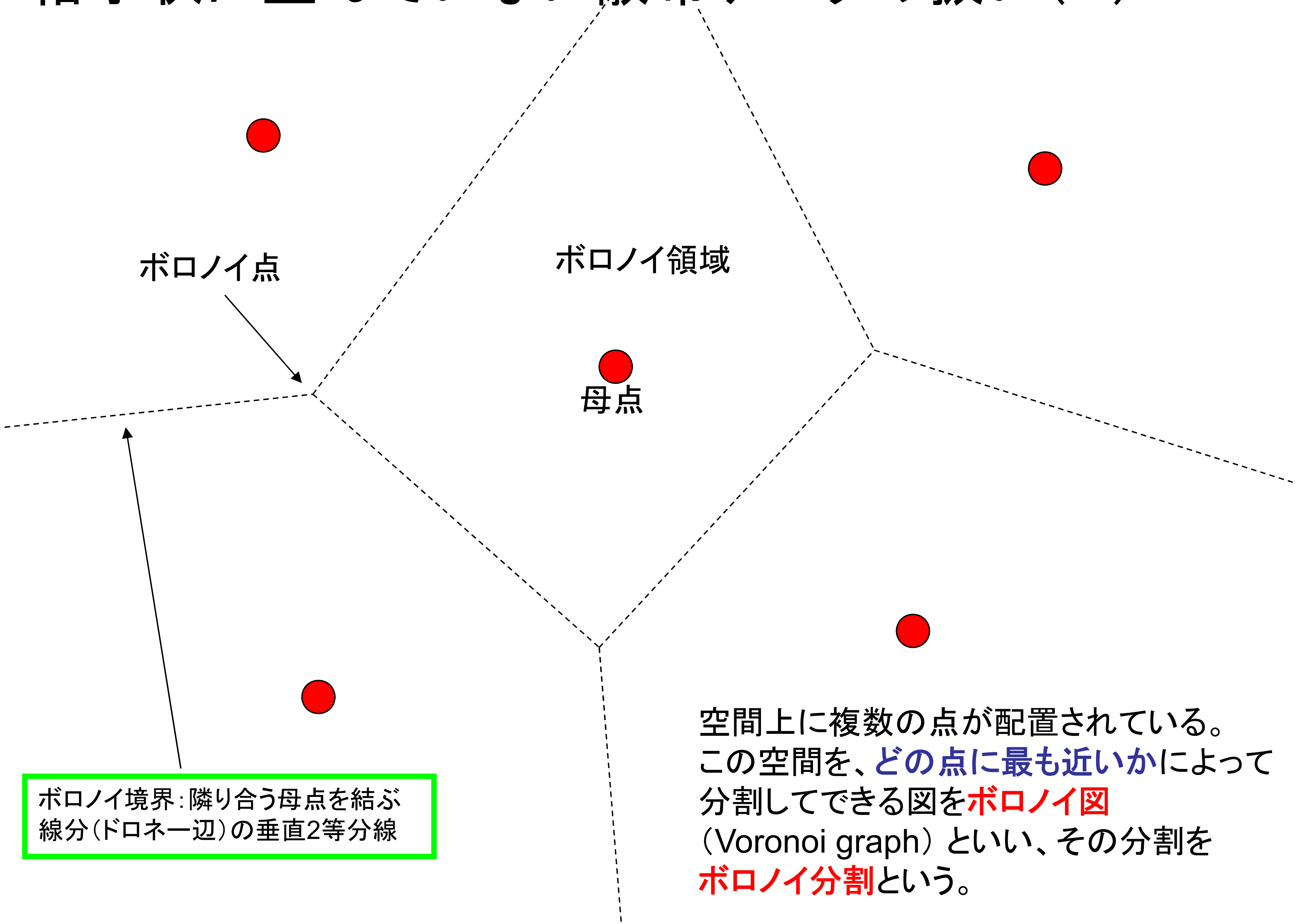
【関数近似の問題設定】

多次元空間X
データ群は**まばらで少ない**

データ群 (X1,Y1), (X2,Y2), … (Xn,Yn) が与えられているもとで、
未知なる Xu についてのYuの値を求める
ただし X1～Xn, Y1～Yn, Xu,Yu は全て有界な値（スカラー or ベクトル量）



格子状に並んでいない散布データの扱い（1）

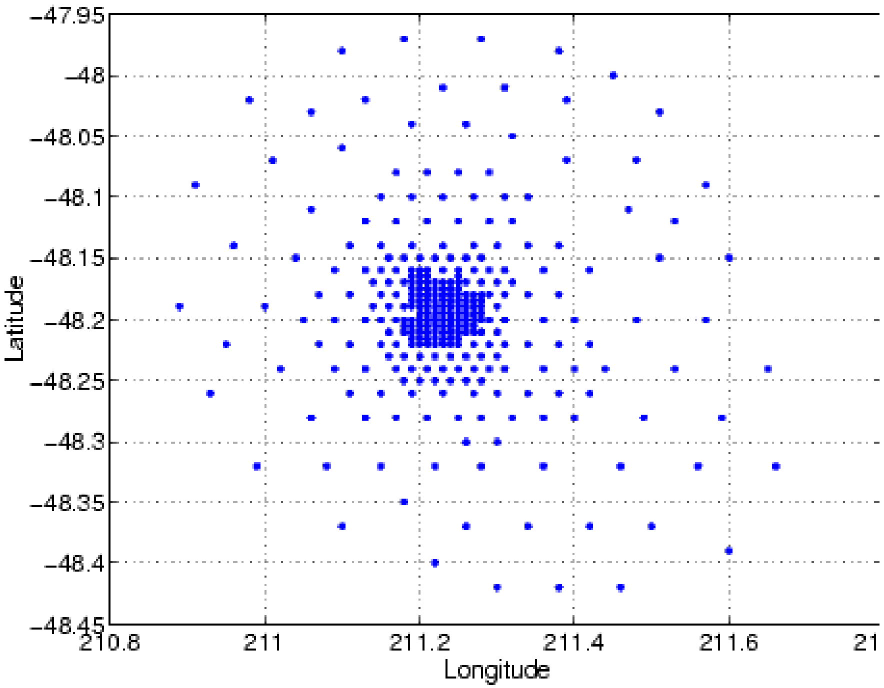


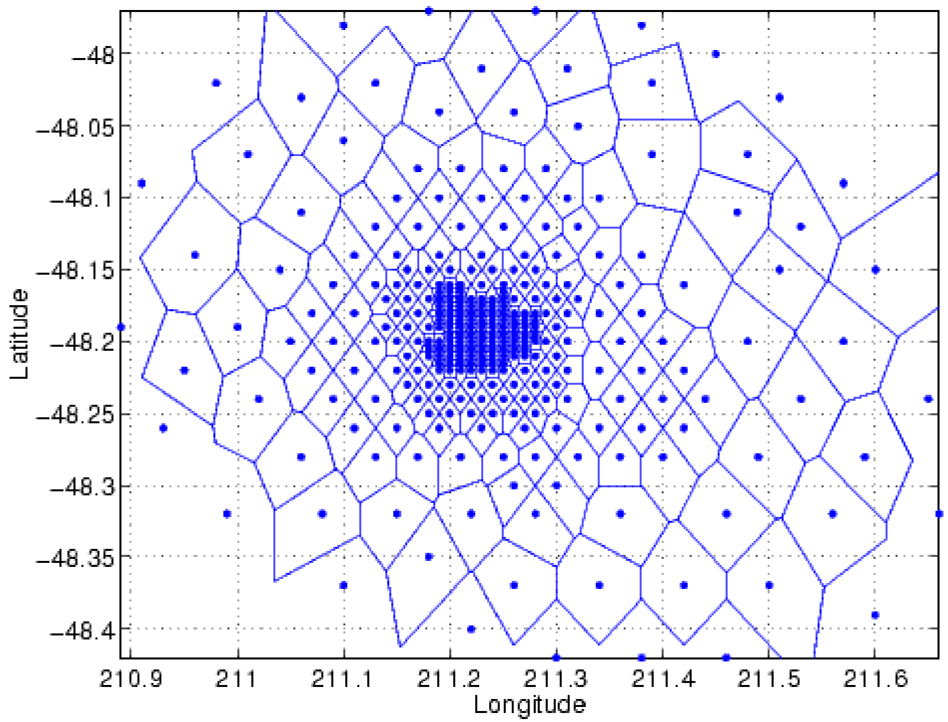
空間上に複数の点が配置されている。
この空間を、**どの点に最も近いか**によって
分割してできる図を**ボロノイ図**
（Voronoi graph）といい、その分割を
ボロノイ分割という。

ボロノイ分割の応用:

・パターン認識(識別・クラスタリング)
・データの符号化(量子化)
・施設・サービス等の勢力図の分析など

関数近似や補間よりも「離散化」に適する



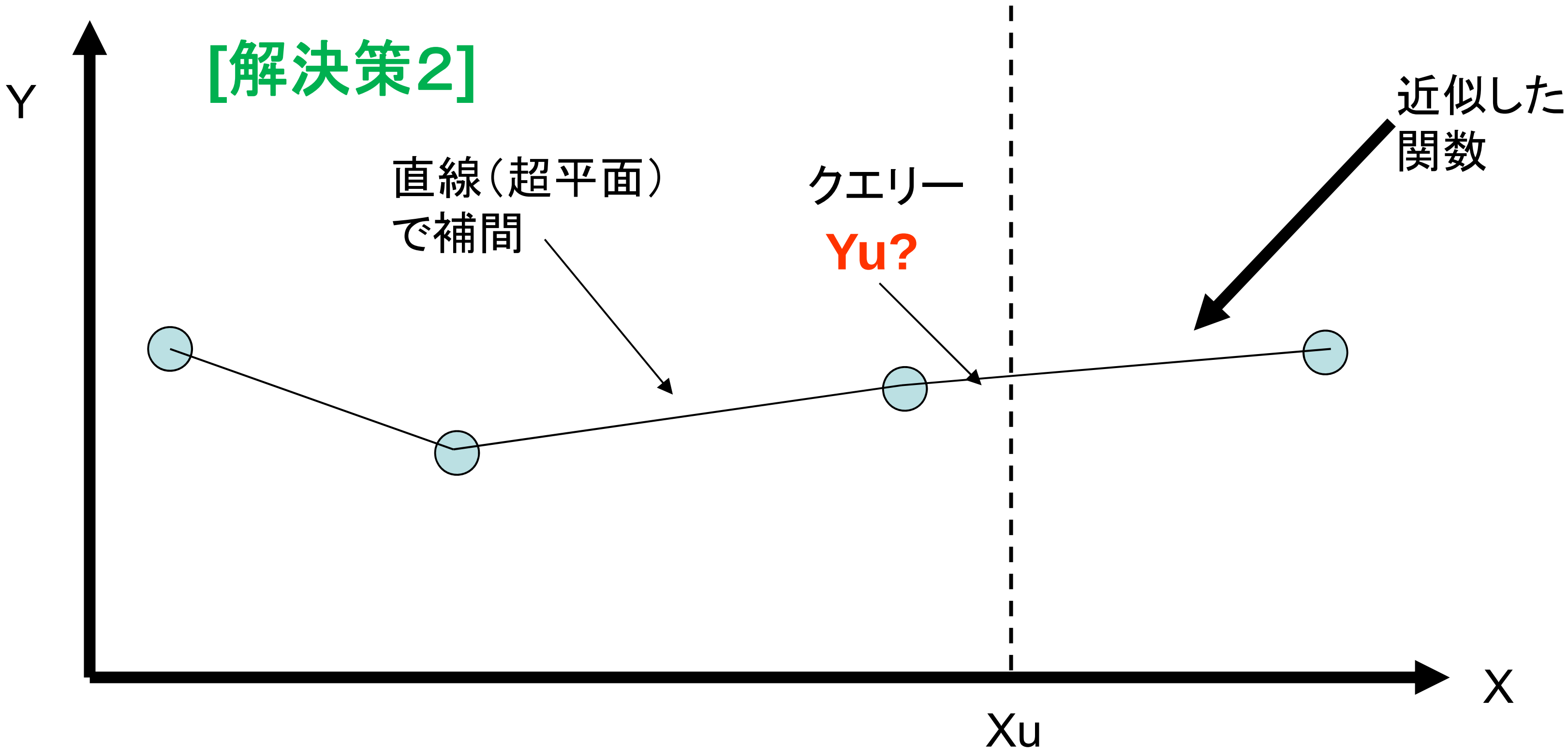


http://dl.cybernet.co.jp/matlab/support/manual/r14/toolbox/matlab/math/?/matlab /support/manual/r14/toolbox/matlab/math/poly_i18.shtml より

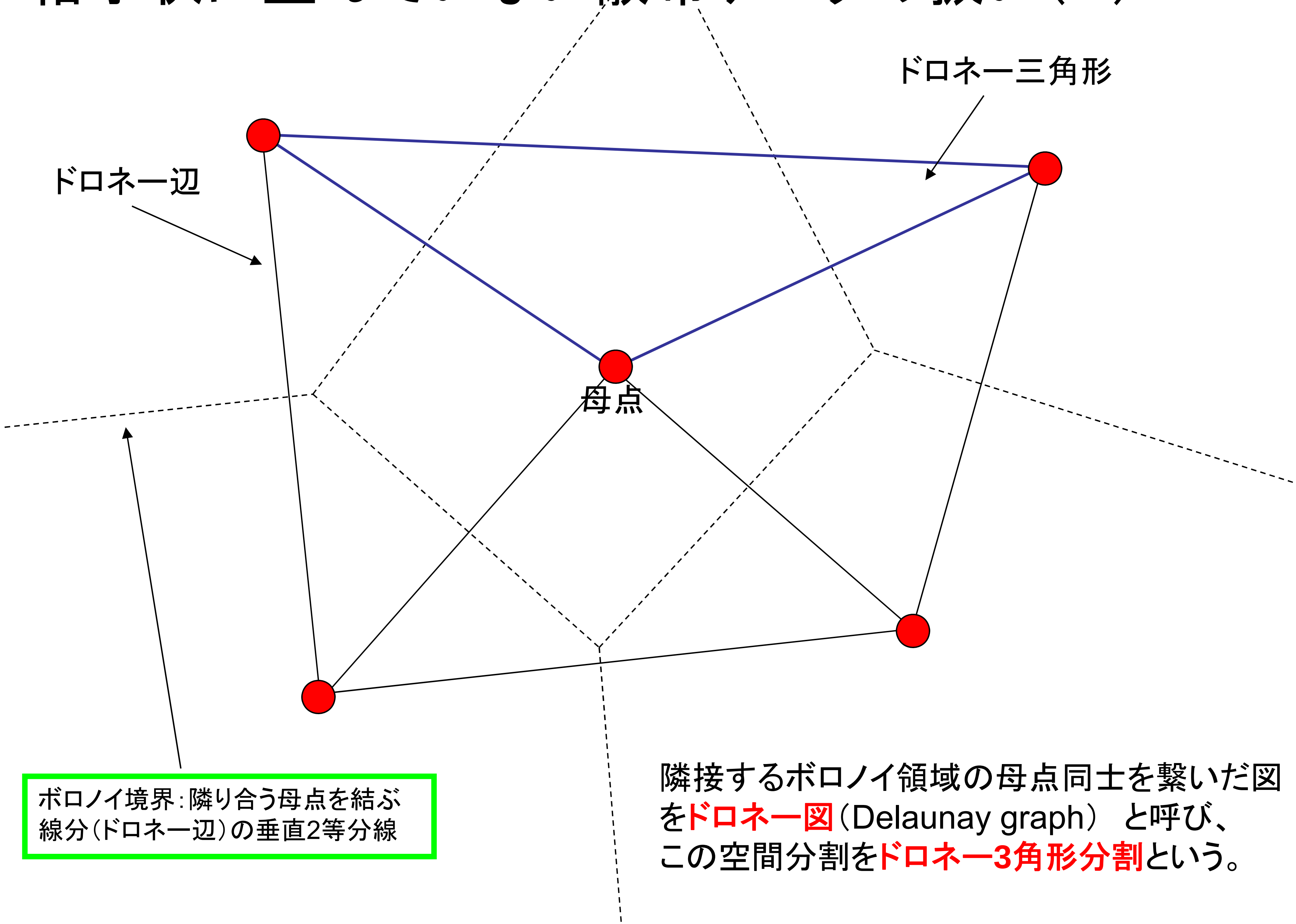
【関数近似の問題設定】

多次元空間X
データ群はまばらで少ない

データ群 (X1,Y1), (X2,Y2), … (Xn,Yn) が与えられているもとで、
未知なる Xu についてのYuの値を求める
ただし X1～Xn, Y1～Yn, Xu,Yu は全て有界な値（スカラー or ベクトル量）

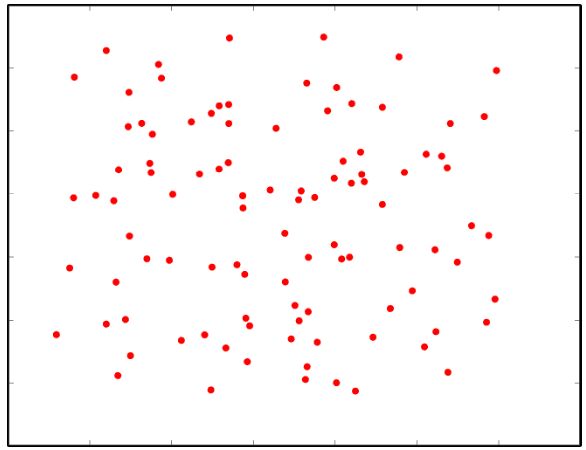


格子状に並んでいない散布データの扱い（2）



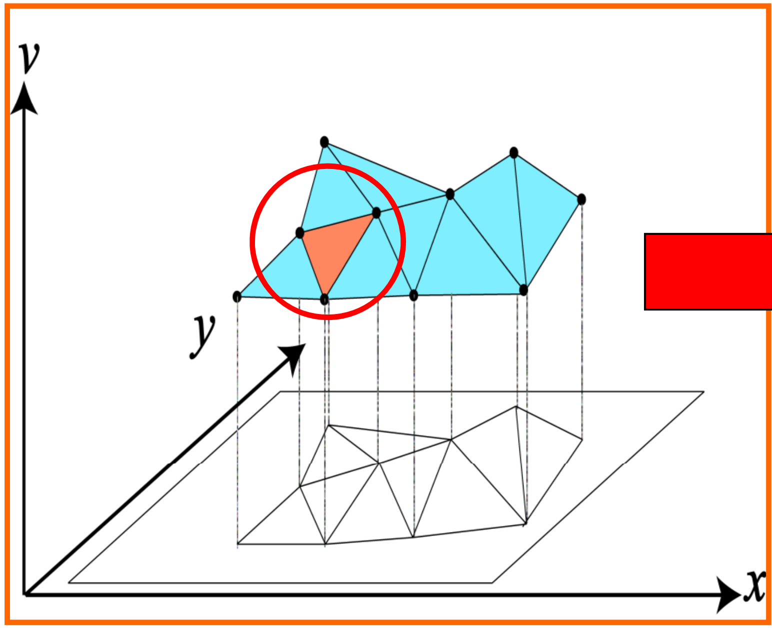
隣接するボロノイ領域の母点同士を繋いだ図を**ドロネー図**（Delaunay graph） と呼び、この空間分割を**ドロネー3角形分割**という。

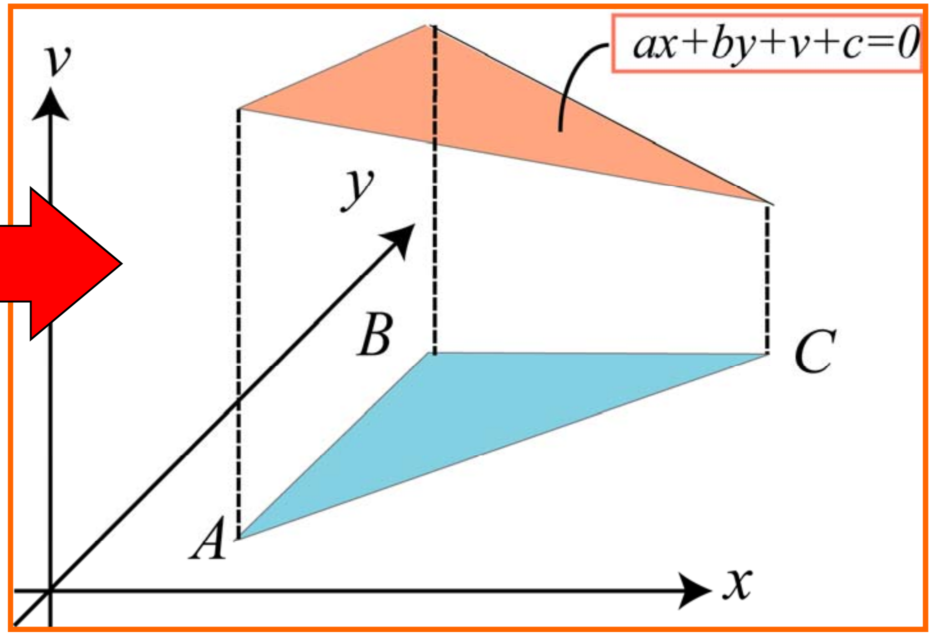
格子状に並んでいないデータの扱い



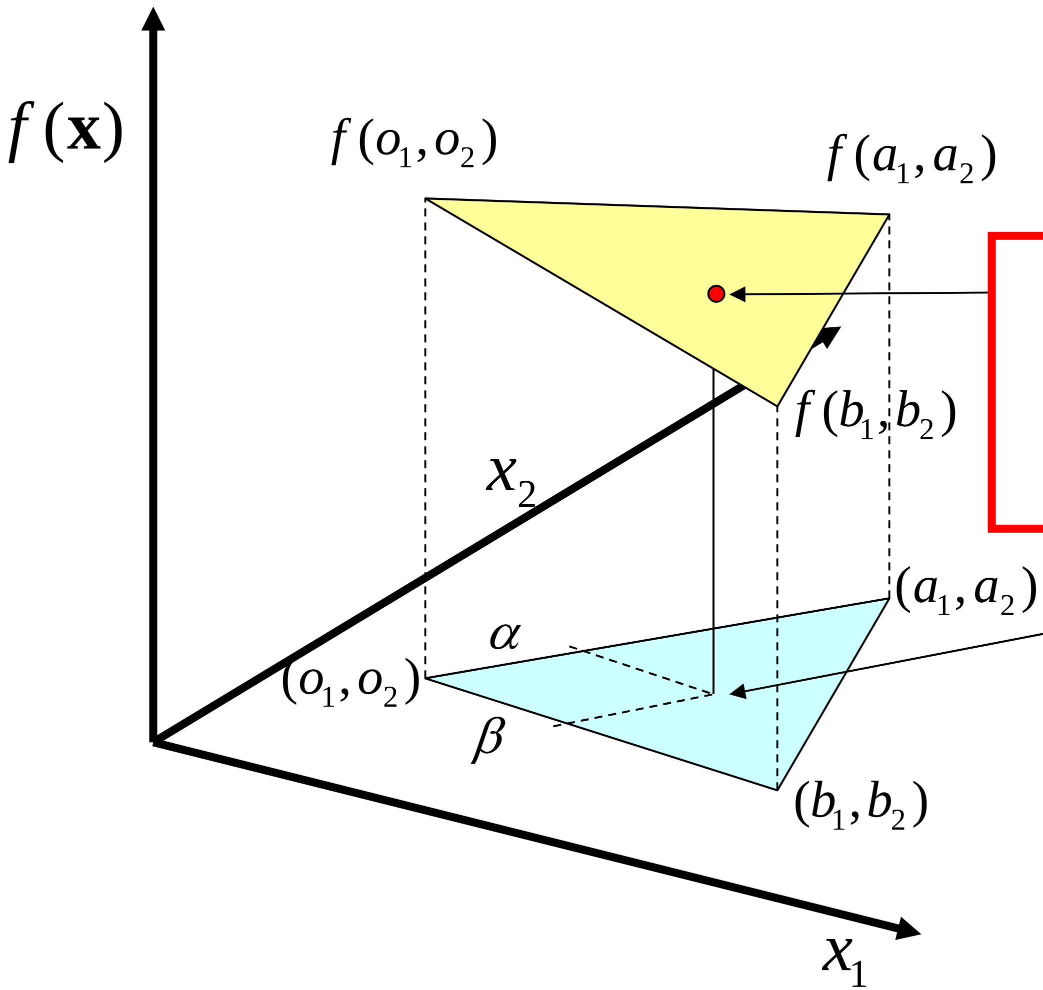
格子状に並んでいない測定データ
が2次元空間上に広がっている場合

3角形パネルに分割して補間





補間の計算方法



(x_1, x_2) この点の関数値を3角形の頂点の関数値の補間により計算する

$$\begin{bmatrix} x_1 - o_1 \\ x_2 - o_2 \end{bmatrix} = \begin{bmatrix} a_1 - o_1 & b_1 - o_1 \\ a_2 - o_2 & b_1 - o_2 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

より、

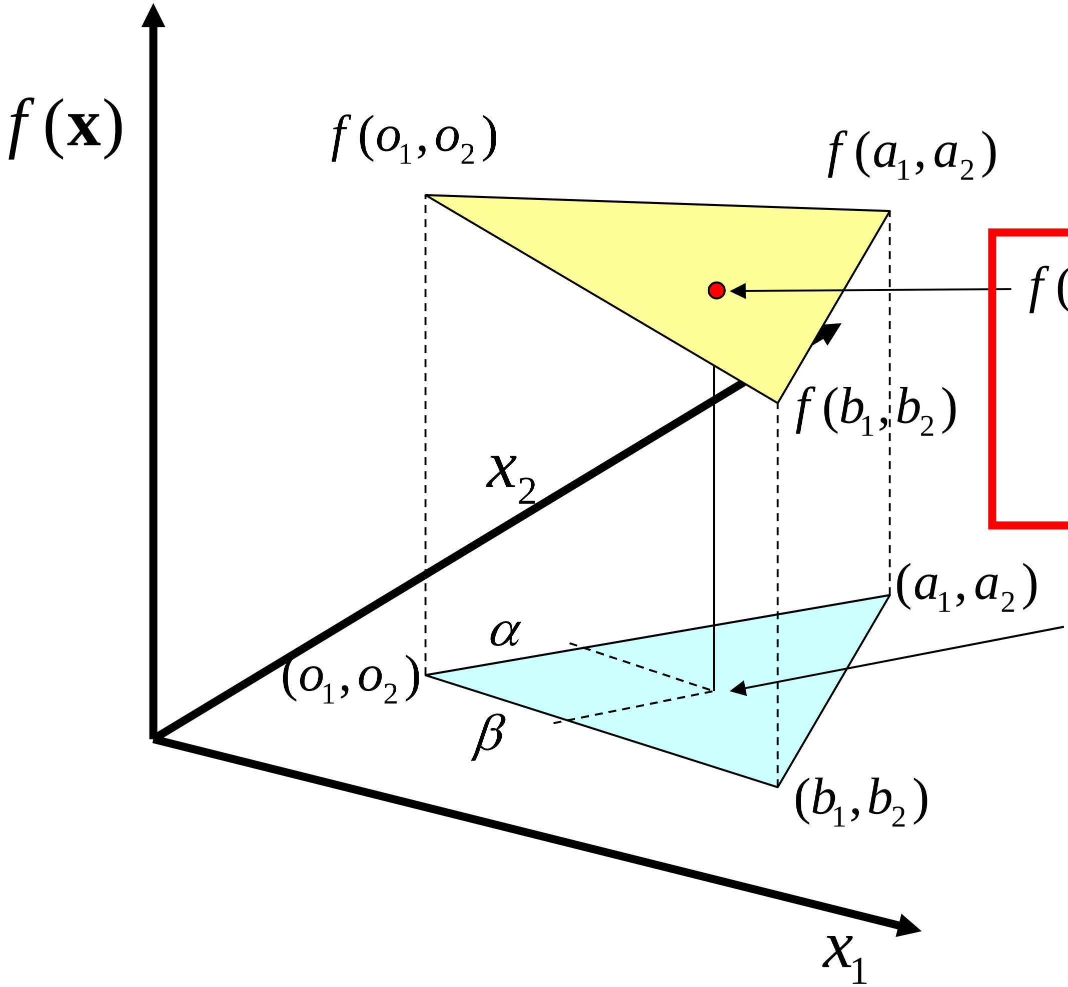
$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} a_1 - o_1 & b_1 - o_1 \\ a_2 - o_2 & b_1 - o_2 \end{bmatrix}^{-1} \begin{bmatrix} x_1 - o_1 \\ x_2 - o_2 \end{bmatrix}$$

ただし3角形内部に点がある条件は、以下の不等式を全て満たすこと:

$$0 \le \alpha \le 1, \quad 0 \le \beta \le 1, \quad \alpha + \beta \le 1$$

上記は2次元空間の例だが、n次元空間でも全く同じ

補間の計算方法



$$\boxed{\begin{aligned} f(x_1, x_2) = f(o_1, o_2) & \\ +\alpha\big(f(a_1, a_2) - f(o_1, o_2)\big) & \\ +\beta\big(f(b_1, b_2) - f(o_1, o_2)\big) & \end{aligned}}$$

(x_1, x_2) この点の関数値を3角形の頂点の関数値の補間により計算する

$$\begin{bmatrix} x_1 - o_1 \\ x_2 - o_2 \end{bmatrix} = \begin{bmatrix} a_1 - o_1 & b_1 - o_1 \\ a_2 - o_2 & b_1 - o_2 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

より、

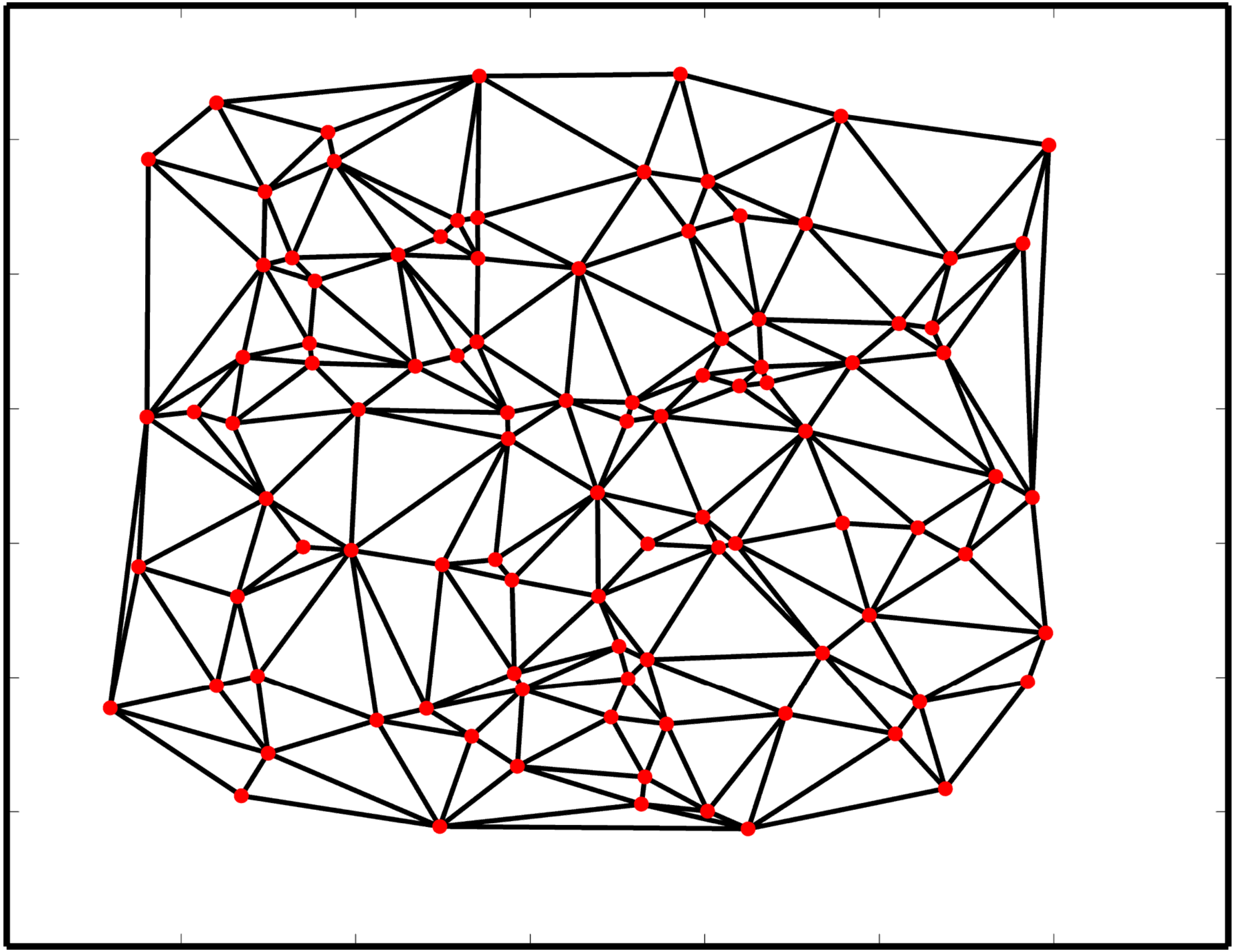
$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} a_1 - o_1 & b_1 - o_1 \\ a_2 - o_2 & b_1 - o_2 \end{bmatrix}^{-1} \begin{bmatrix} x_1 - o_1 \\ x_2 - o_2 \end{bmatrix}$$

上記は2次元空間の例だが、
n次元空間でも全く同じ

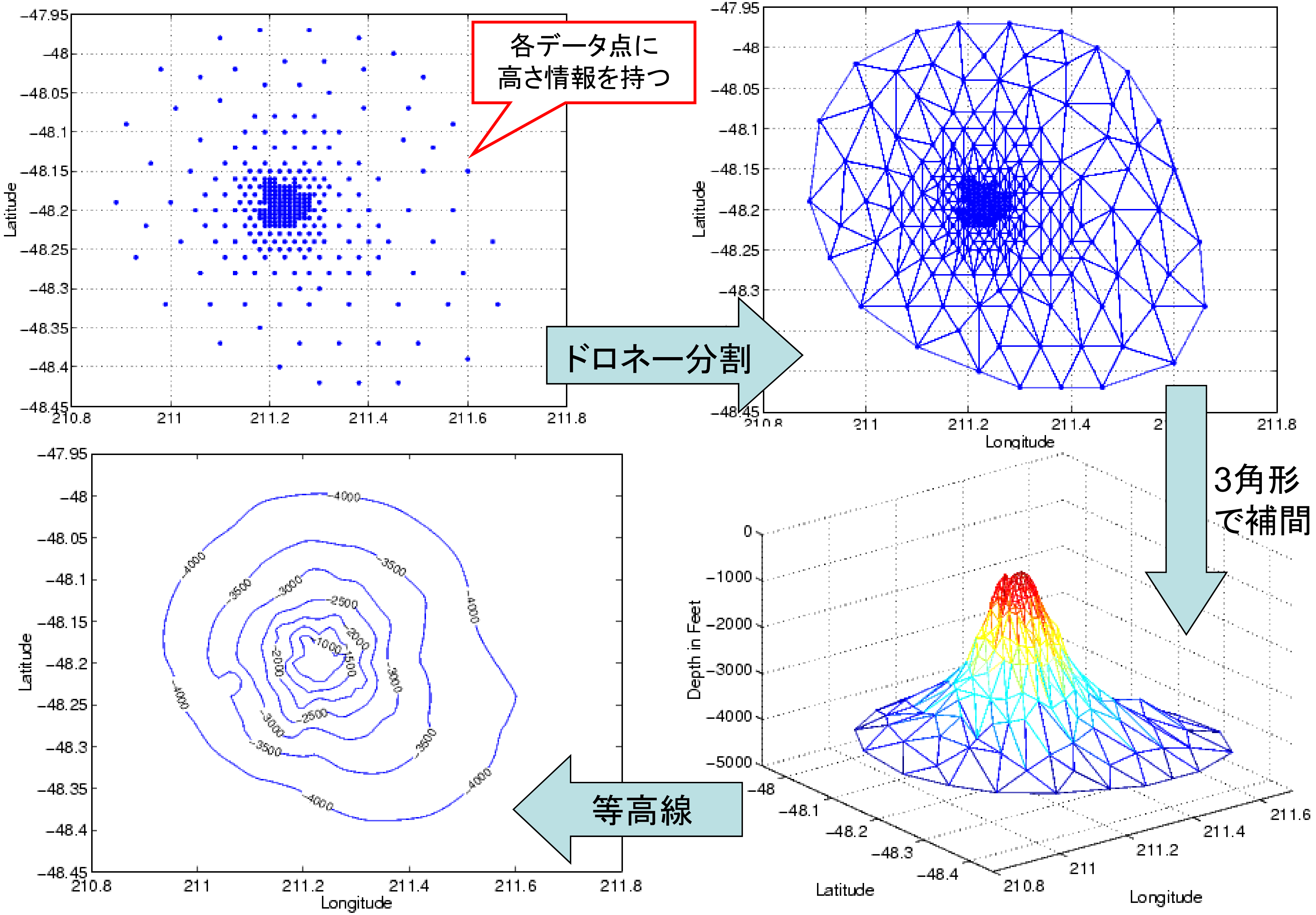
ただし3角形内部に点がある条件は、
以下の不等式を全て満たすこと:

$$0 \le \alpha \le 1, \quad 0 \le \beta \le 1, \quad \alpha + \beta \le 1$$

Delaunay三角形分割(ランダム点における)



ドロネー図の応用： 散布データの補間



ドロネー図・ボロノイ図の性質

・ドロネー辺の垂直2等分線がボロノイ辺に一致する

・ボロノイ分割における各ボロノイ点（ボロノイ分割の多角形の頂点）
　が、 ドロネー分割による3角形の外接円の中心に等しい。

・ドロネー3角形の外接円の内部に、他の母点は含まれない

・ボロノイ図とドロネー図は**双対**の関係にある

ドロネー図の生成（2次元平面の場合）

・逐次添加法など効率の良いアルゴリズムが提案されている。

ドロネー図・ボロノイ図の性質

・ドロネー辺の垂直2等分線がボロノイ辺に一致する

・ボロノイ分割における各ボロノイ点（ボロノイ分割の多角形の頂点）
　が、 ドロネー分割による3角形の外接円の中心に等しい。

・ドロネー3角形の外接円の内部に、他の母点は含まれない

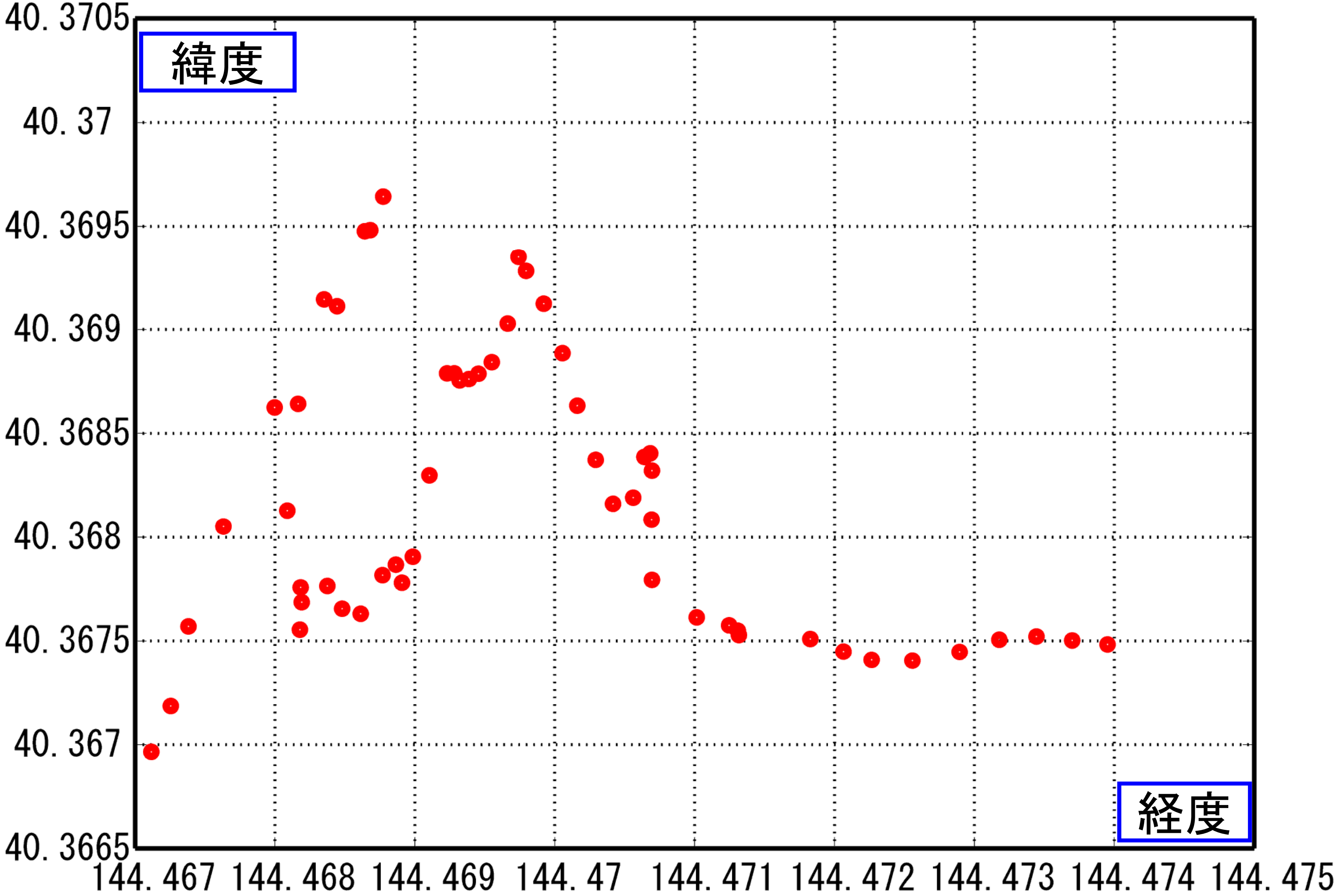
・ボロノイ図とドロネー図は**双対**の関係にある

ドロネー図の生成（2次元平面の場合）

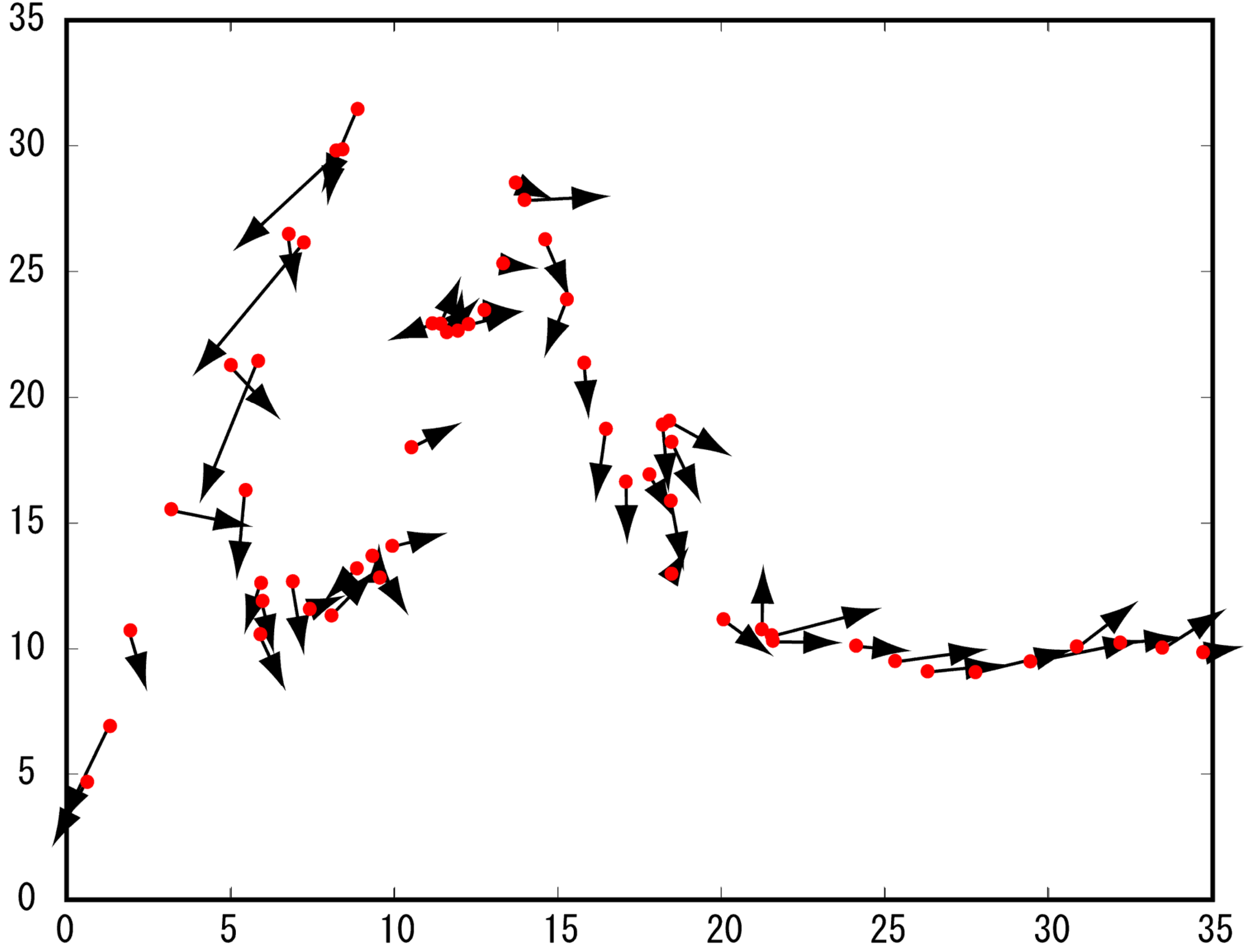
全ての点の中から3点を選び、その**外接円**を描く。
このとき、**円内にその3点以外の点が含まれなければ**、
それらを3角形として結ぶ。
この作業を全ての3点の組合せについて行ったとき、
最終的に得られる3角形分割はドロネー分割になっている。

・逐次添加法など効率の良いアルゴリズムが提案されている。

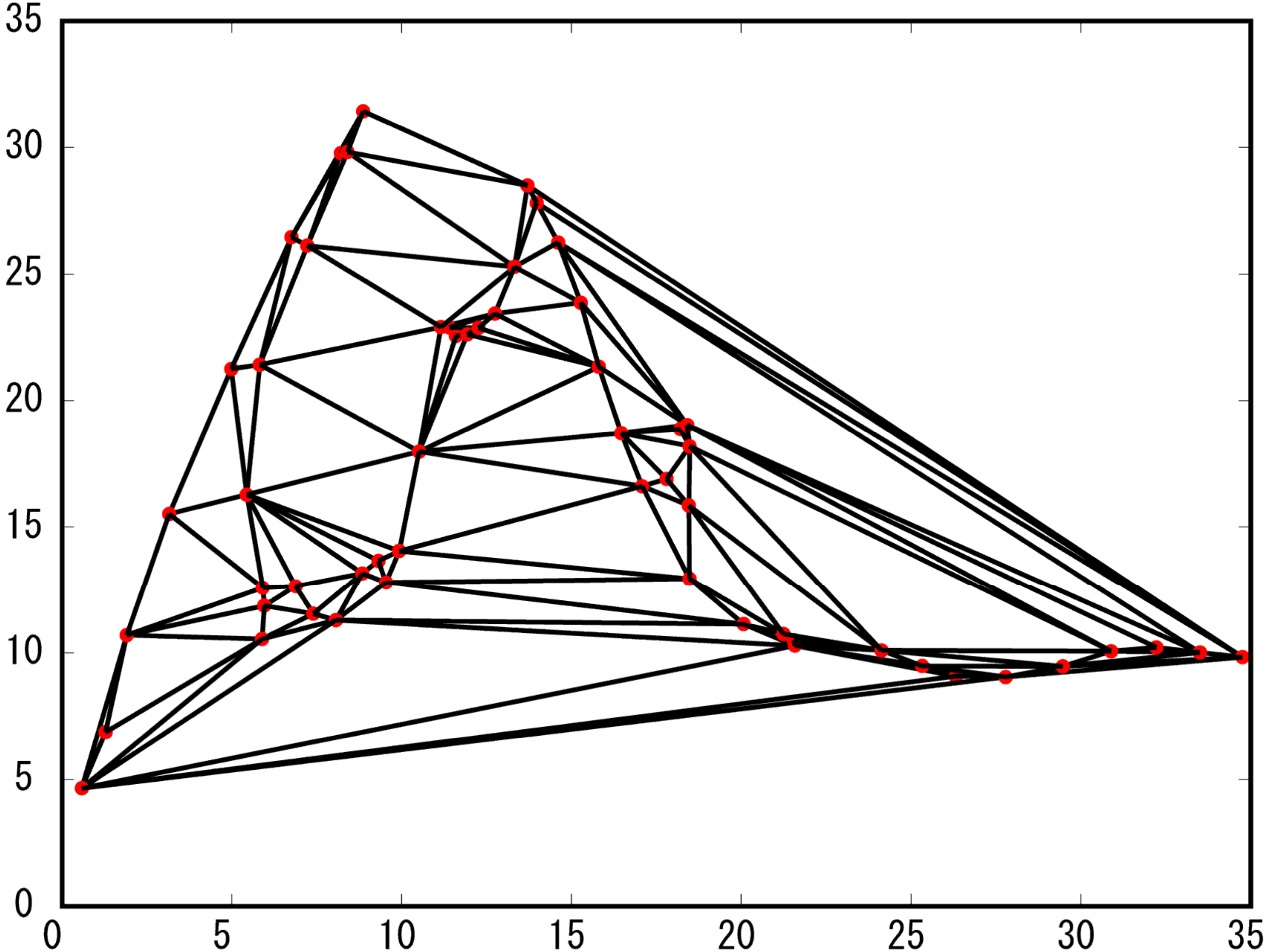
実際の潮流計測データ（JAMSTEC提供）



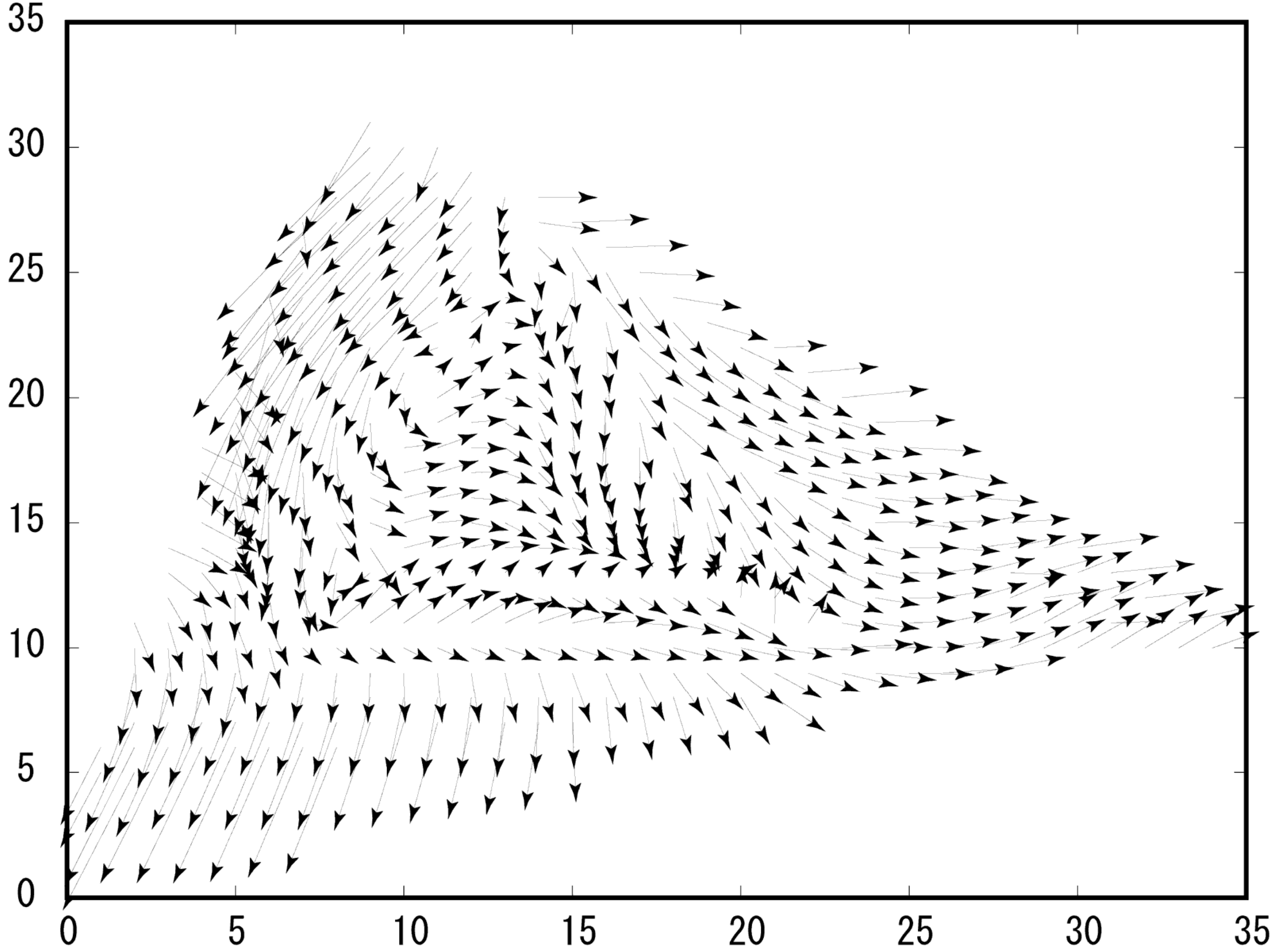
潮流場(観測データ)



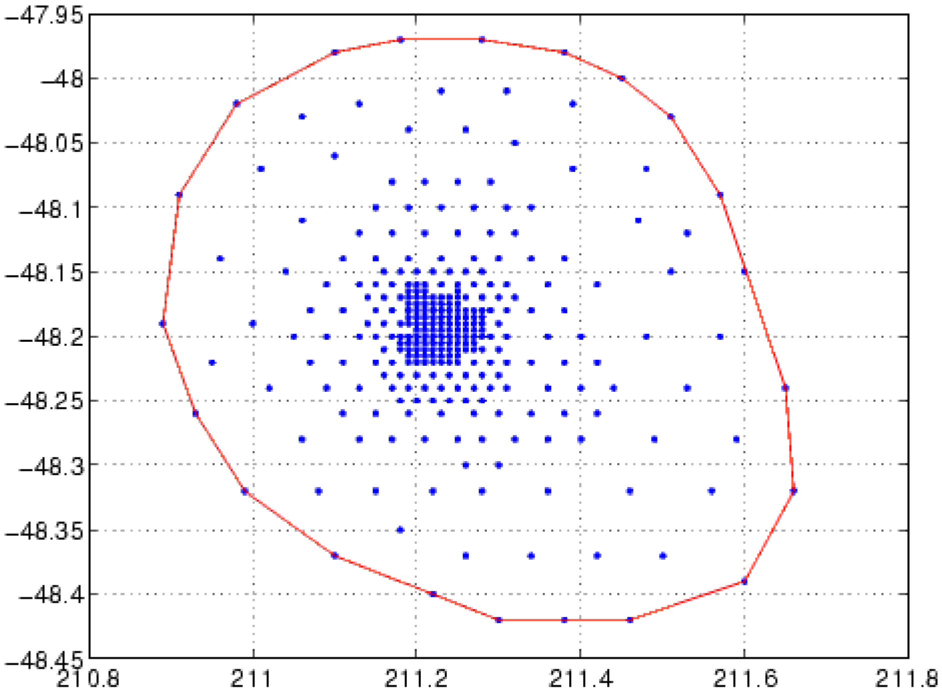
領域をDelaunay分割



未観測領域の潮流を補間

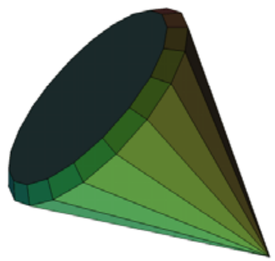


凸包：　データ点を全て覆う凸多角形



凸包の応用：
クエリー点がデータ点で覆われる
空間の中にあるかどうかを判定
（中にあれば精度良く補間が可能）

多次元空間でのボロノイ分割・ドロネー分割・凸包を行う計算機プログラム：　Qhull

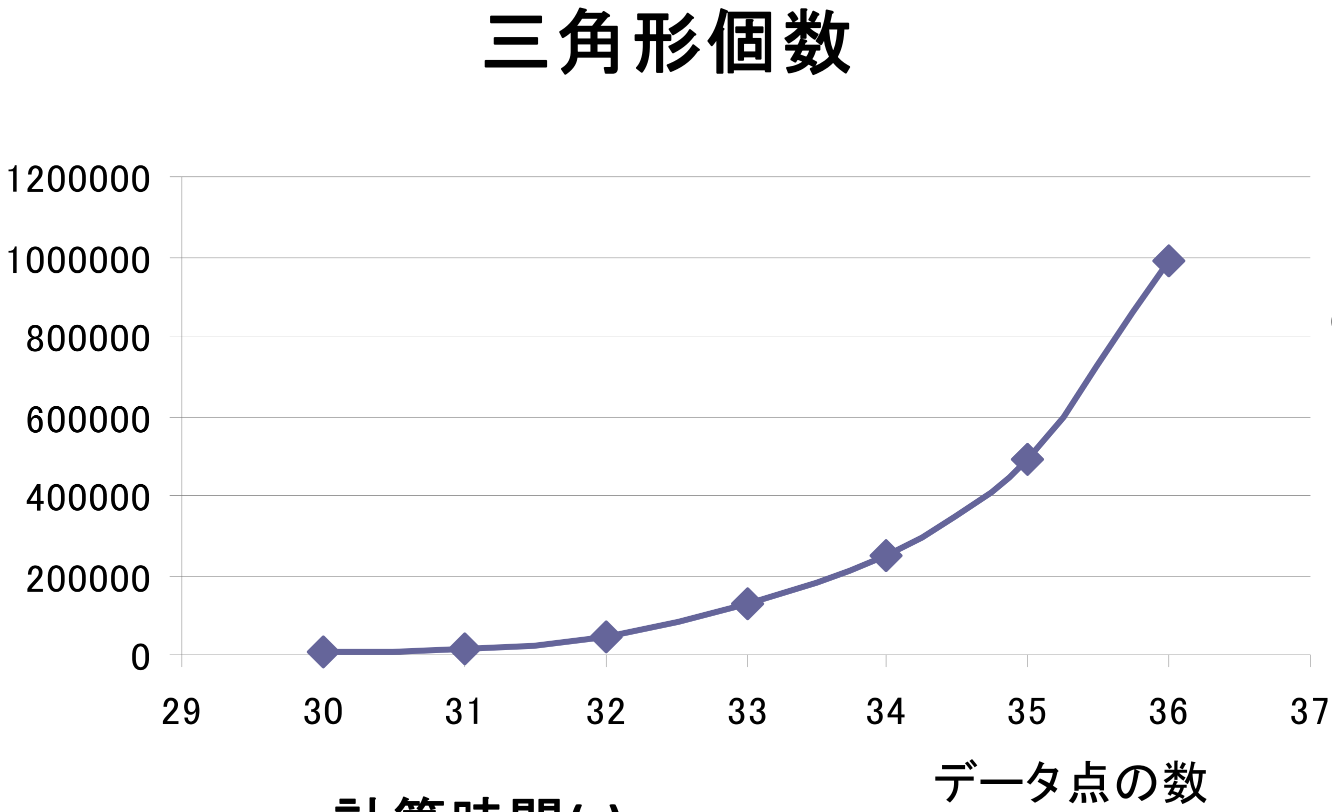


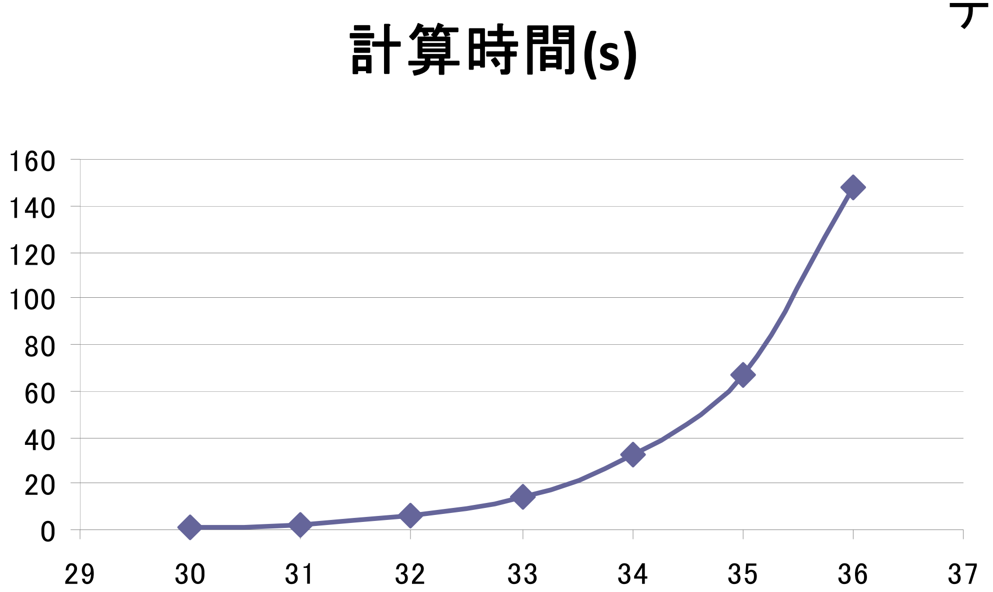
http://www.qhull.org/

データを標準入力から入力し、計算結果は標準出力から出力
Windows版やUnix版のバイナリやソースも提供されているのでおすすめ
詳しくは上記サイトを参照

ドロネー分割の問題点１

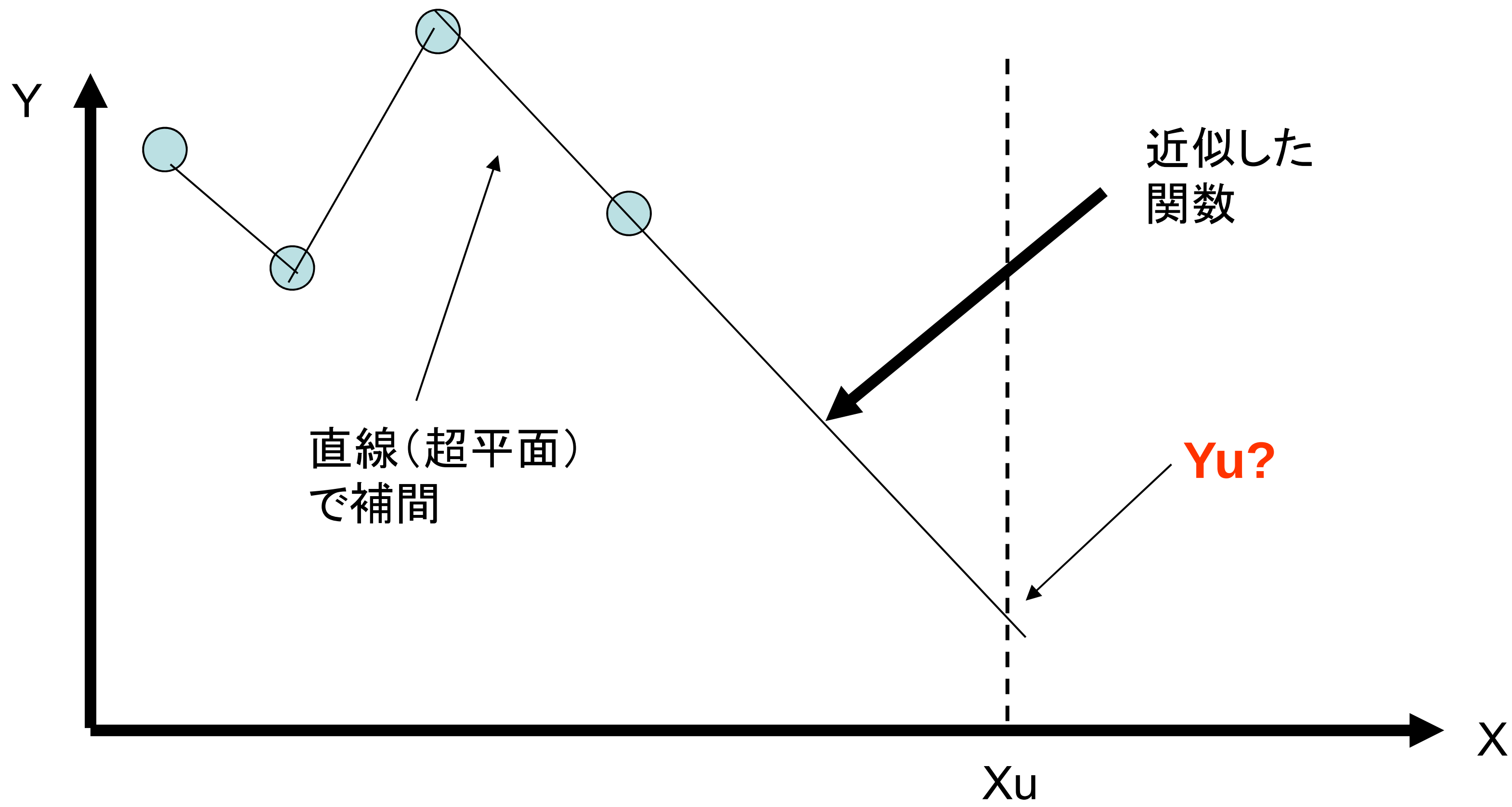
QHULLによる計算量
（24次元空間）





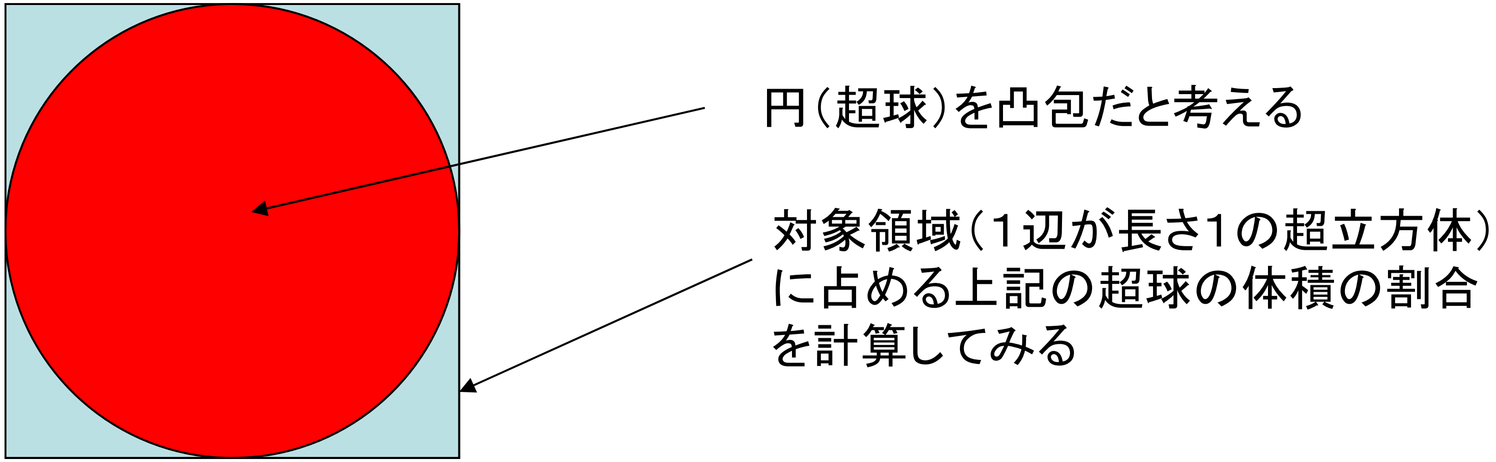
次元が増えると
シンプレックス（単体：高次元の３角形）
の数が爆発的に増加し、
計算不能になる

【ドロネー分割法の問題点２】



【問題点】
パラメータ空間が高次元だと、クエリーがデータの凸包に入らず、補外になるため近似精度が全く出ない

高次元パラメータ空間においてクエリーがデータ集合の凸包に入らない理由



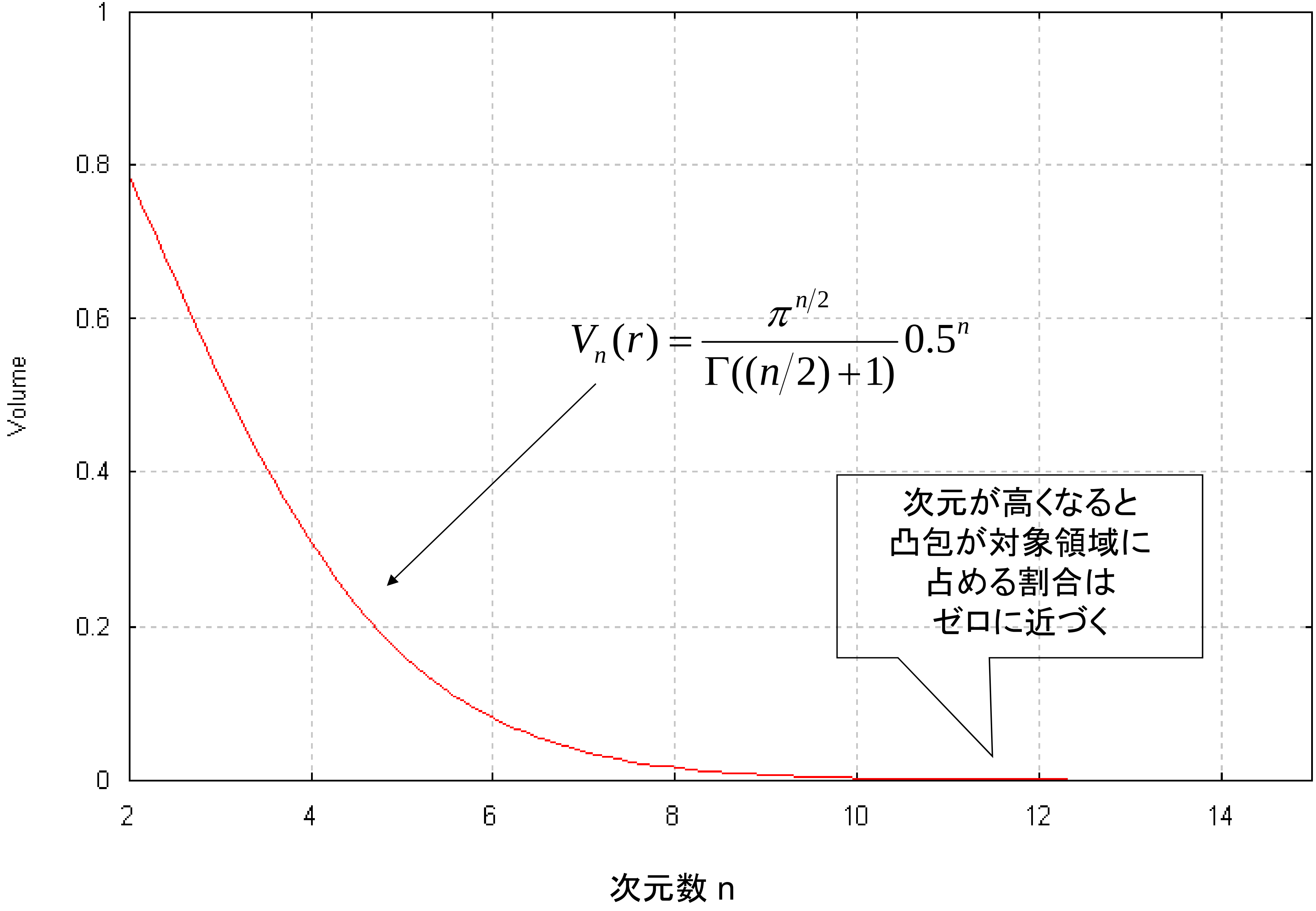
n 次元空間における半径 r の球の体積 $V_n(r) = \dfrac{\pi^{n/2}}{\Gamma((n/2)+1)} r^n$

・次元数 n が偶数 n=2m の場合

$$V_n(r) = V_{2m}(r) = \frac{2^m \pi^m r^{2m}}{2 \cdot 4 \cdot 6 \cdots (2m)} = \frac{\pi^m r^{2m}}{m!}$$

・次元数 n が奇数 n=2m+1 の場合

$$V_n(r) = V_{2m+1}(r) = \frac{2^{m+1} \pi^m r^{2m+1}}{1 \cdot 3 \cdot 5 \cdots (2m+1)}$$



1
0.8
0.6
0.4
0.2
0
Volume
2
4
6
8
10
12
14
$V_n(r) = \frac{\pi^{n/2}}{\Gamma((n/2)+1)} 0.5^n$
次元が高くなると
凸包が対象領域に
占める割合は
ゼロに近づく
次元数 n

まとめ

- **多項式曲線・多項式曲面**

  パラメータ変数を用いた表現
  通過点や端点での傾きを指示

- スプライン曲線

  多項式曲線を多数用いる
  接続点での傾きを等しくする

- ボロノイ分割

  領域に分割：**文字やパターン識別**に有効
  なめらかな補間には不向き

- ドロネー分割

  ボロノイの双対・３角形に分割して補間
  離散的な高さデータから**地形図を作成するような補間計算**に有効
  高次元空間に対しては不向き

【演習問題】　2013.10.13　学籍番号　　　　　　氏名

２点 $\mathbf{P}_1 = [0,0,0]$ および $\mathbf{P}_2 = [1,1,0]$ を通り、
点 $\mathbf{P}_1$ での接線方向ベクトルが $\dot{\mathbf{P}}_1 = [1,0,0]$
点 $\mathbf{P}_2$ での接線方向ベクトルが $\dot{\mathbf{P}}_2 = [1,0,0]$
で与えられる3次多項式曲線 $\mathbf{P}(t) = [x(t)\ y(t)\ z(t)] = \mathbf{a}_0 + \mathbf{a}_1 t + \mathbf{a}_2 t^2 + \mathbf{a}_3 t^3$
の係数ベクトルを全て求めよ。

【演習問題】　　　　　　　　　　　学籍番号　　　　　　　　氏名

２点　$\mathbf{P}_1 = [0,0,0]$　および　$\mathbf{P}_2 = [1,1,0]$　を通り、
点　$\mathbf{P}_1$　での接線方向ベクトルが　$\dot{\mathbf{P}}_1 = [1,0,0]$
点　$\mathbf{P}_2$　での接線方向ベクトルが　$\dot{\mathbf{P}}_2 = [1,0,0]$
で与えられる3次多項式曲線　$\mathbf{P}(t) = [x(t)\ y(t)\ z(t)] = \mathbf{a}_0 + \mathbf{a}_1 t + \mathbf{a}_2 t^2 + \mathbf{a}_3 t^3$
の係数ベクトルを全て求めよ。

$$\begin{cases} x = t \\ y = 3t^2 - 2t^3 \\ z = 0 \end{cases}$$

よって

$$\mathbf{a}_0 = [0 \quad 0 \quad 0]$$
$$\mathbf{a}_1 = [1 \quad 0 \quad 0]$$
$$\mathbf{a}_2 = [0 \quad 3 \quad 0]$$
$$\mathbf{a}_3 = [0 \quad -2 \quad 0]$$

$$\Rightarrow \quad y = 3x^2 - 2x^3$$
$$z = 0$$