理解するための ＧＰＳ測位計算プログラム入門

(その3) 測位計算のはなし

独立行政法人　電子航法研究所　福島　荘之介

1. はじめに

(**その1**)ではWGS84座標系と ECEF 座標, 測地座標を説明し, 地平座標に変換した。(**その2**)では実際にGPS衛星から放送された航法メッセージをデコードし, 軌道パラメータを抽出して, 衛星の3次元位置(ECEF座標)を求めた。さて, これで準備はそろった。今回はいよいよ測位計算を説明し, プログラム例を紹介する。前回同様, このプログラム例では, ノバテル社のGPS受信機(RT20)を利用し, 実際の受信データから, 独自の測位計算を実行する。更には受信機が測位計算した結果と独自の計算結果の比較を試みる。この結果, 驚くなかれ！重み付きの最小2乗法を用いたとき, 独自の計算値とノバテル受信機の個々の測位値は1cm 以下の差で一致する。

2. ＧＰＳの測位原理を理解する

GPSの測位原理というと, まず**図3．1**のような概念で,「衛星の位置を既知として, 衛星からの電波の伝搬時間から距離を測定し, 衛星から等距離である円弧の交点として位置を求める」と説明されることが多い。確かにこの説明は測位原理を簡単に示している。しかし, 次に「この図を3次元で考えると, 3個の衛星から等距離の球面が一点に定まり, 実際には受信機に時計の誤差があるため, 衛星を4個用いて, この交点が一点に交わるように時計の誤差を修正する」と説明が続くと, 4衛星の必要理由が今ひとつしっくりこないと感じる読者がいるかもしれない。そこで, (**その1**) (**その2**)を読んだ方には, 次のような説明が明快で, しかも厳密である。

図3．2は, (**その1**)で説明した ECEF 直交座標

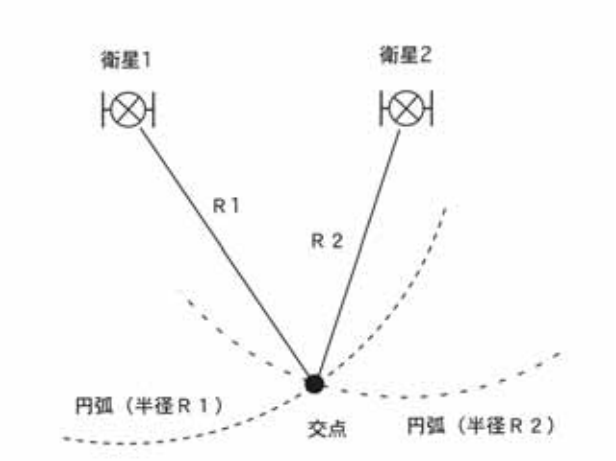

図3．1：測位原理(概念的説明)

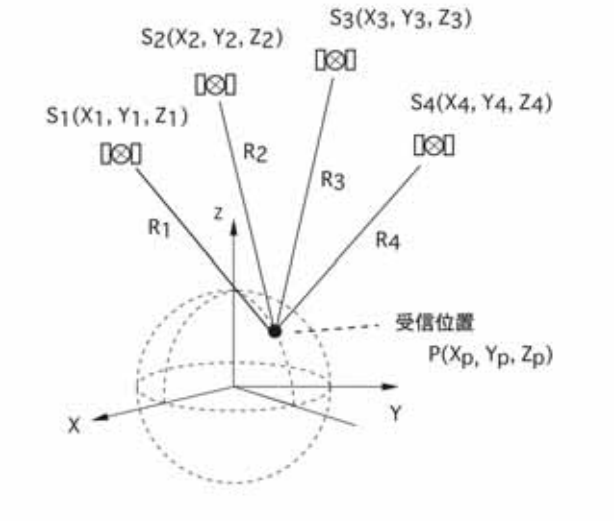

図3．2：測位原理(観測方程式)

である。この座標上で1番目の衛星の位置をベクトルS_1とし, その要素を(x_1, y_1, z_1)とする。同様に2番目の衛星の位置をS_2, 3番衛星, 4番衛星の位

置をS_3, S_4とする。次に, 求めるべき受信点の位置を$P(x_p, y_p, z_p)$とする。また, 伝搬時間の差から測定した時計の誤差を含む仮の距離(擬似距離という)を各衛星ごとにR_1, R_2, R_3, R_4とする。このとき, i番衛星(i=1,2,3,4)と受信点Pの真の距離

$$\sqrt{(x_i - x_p)^2 + (y_i - y_p)^2 + (z_i - z_p)^2}$$

は, 擬似距離R_iとの間に,

$$R_i = \sqrt{(x_i - x_p)^2 + (y_i - y_p)^2 + (z_i - z_p)^2} + c\,\Delta t + b_i \quad (1)$$

の関係がある。ここでcは光速(定数), Δtは受信機の時計オフセット, b_iは衛星のクロックオフセット表す。この式に含まれる変数のうち, 既に分かっているのは, 衛星位置x_i, y_i, z_iとb_i受信機で測定されるR_iであり, 未知である変数は, 受信点の位置であるx_p, y_p, z_pと時計オフセットのΔtである。従って, 未知数が4つなので, 4つの方程式を立てれば解が定まる。以上が測位に衛星を最低4個必要とする理由である。

3. 連立方程式を解くには

式(1)で i=1,2,3,4 とした連立方程式を解けば受信位置 P が求まるはずである。しかし, この非線形の連立方程式は簡単には解けない。そこで, 反復による逐次計算法(ニュートン法)がよく使われる(文献[1.1, 1.2]に詳しい説明がある)。

この方法では, 求めるべき未知数を, 初期値($_0$)と修正値(Δ)の和,

$$x_p = x_0 + \Delta x$$
$$y_p = y_0 + \Delta y$$
$$z_p = z_0 + \Delta z$$

と仮定する。次に式(1)を修正値(Δ)で線形化した,

$$\Delta R_i = \frac{\partial R}{\partial x}\Delta x + \frac{\partial R}{\partial y}\Delta y + \frac{\partial R}{\partial z}\Delta z + \Delta S \quad (2)$$

という近似式を考える。ここで, ΔR_iは擬似距離と初期値から計算される推定距離の差分で

$$\Delta R_i = R_i - \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} \quad (3)$$

となる。ΔSは$c\Delta t$を意味する。また, 偏微分項は

$$\frac{\partial R}{\partial x} = \frac{-(x_i - x_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}}$$
$$\frac{\partial R}{\partial y} = \frac{-(y_i - y_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}}$$
$$\frac{\partial R}{\partial z} = \frac{-(z_i - z_0)}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}}$$

となる。これを上からα_i, β_i, γ_iとおくと, 衛星数が4個の場合, 式(2)は,

$$\begin{pmatrix} \Delta R_1 \\ \Delta R_2 \\ \Delta R_3 \\ \Delta R_4 \end{pmatrix} = \begin{pmatrix} \alpha_1 & \beta_1 & \gamma_1 & 1 \\ \alpha_2 & \beta_2 & \gamma_2 & 1 \\ \alpha_3 & \beta_3 & \gamma_3 & 1 \\ \alpha_4 & \beta_4 & \gamma_4 & 1 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta s \end{pmatrix} \quad (4)$$

と行列の形で書くことができる。さらに,

$$A = \begin{pmatrix} \alpha_1 & \beta_1 & \gamma_1 & 1 \\ \alpha_2 & \beta_2 & \gamma_2 & 1 \\ \alpha_3 & \beta_3 & \gamma_3 & 1 \\ \alpha_4 & \beta_4 & \gamma_4 & 1 \end{pmatrix}$$

$$\Delta X = \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta s \end{pmatrix}, \qquad \Delta R = \begin{pmatrix} \Delta R_1 \\ \Delta R_2 \\ \Delta R_3 \\ \Delta R_4 \end{pmatrix}$$

とおくと, 式(4)は

$$\Delta R = A\,\Delta X \quad (5)$$

と表すことができて, ΔX(修正値)を

$$\Delta X = A^{-1}\,\Delta R \quad (6)$$

と求めることができる。

以上をアルゴリズムの形で書くと次のようになる。

STEP 1. (x_p, y_p, z_p)に適当な初期値(x_0, y_0, z_0)を設定する
STEP 2. 式(3)によりΔR_iを求める
STEP 3. 行列 A を求める
STEP 4. 式(5)のΔXを求める(衛星数が 4 個のときは式(6)でよい。4個以上の場合は次に

説明する最小2乗法を使う)

STEP 5. ΔX の要素 (Δx, Δy, Δz) が十分小さいかどうか判定する

5.1 十分小さくなければ (x0, y0, z0) に (Δx, Δy, Δz) を加え, STEP 2.に戻る。

5.2 十分小さければ終了。(x0, y0, z0) を解として出力する。

このアルゴリズムでは, 初期値に離れた値 (例えば, x0=-4000km, y0=3300km, z0=3700km) を与えても, 2～3回の反復で (Δx, Δy, Δz) は十分小さい値 (例えば 1mm以下) に収束する。

4. 最小2乗法とは

今までは, 衛星数が4個の場合 (i=1,2,3,4) について説明してきた。ところが, 実際に受信される衛星数は, おおむね 8～10 個である。初期の受信機では, もっとも衛星配置がよい (DOPが最小になる) 4個の衛星を選んで計算していたようだ。しかし, 現在の受信機では, 受信した全ての衛星 (n個とする: n>4) を使ってn次の連立方程式を解く。未知数は4つなので, これは「過剰決定の状態」と呼ばれる。実際, 式(1)の右辺には誤差 ε_i が含まれており, 式(5)も

$$\Delta R = A\,\Delta X + \varepsilon \qquad (7)$$

である。ただし, i=1,2,3,…,n。ε は,

$$\varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

とする。そこで, この誤差 ε_i の2乗和を最小にするように解を決定する。これを**最小2乗法**と呼ぶ。

次に最小2乗解の導出を説明する。まず, **観測方程式**とも呼ばれる式(7)の記号を簡単に

$$\varepsilon = R - A\,X \qquad (8)$$

と書き直す。すると誤差 ε の2乗和 f は,

$$\begin{aligned} f &= \varepsilon_1^2 + \varepsilon_2^2 + \cdots + \varepsilon_n^2 \\ &= \varepsilon^T \varepsilon \\ &= (R - AX)^T (R - AX) \\ &= R^T R - 2R^T AX + X^T (A^T A) X \end{aligned}$$

で表せる。ここで T は転置行列を示す。f を最小にするために変数 X で偏微分して極値を求めると,

$$\begin{aligned} \partial f / \partial X &= -2R^T A + 2X^T (A^T A) = 0 \\ X^T (A^T A) &= R^T A \end{aligned}$$

となり, 両辺の転置をとり, $(AB)^T = B^T A^T$ という性質と $(A^T A)$ が対称行列であることから

$$(A^T A) X = A^T R \qquad (9)$$

となる。これを**正規方程式**と呼び, X について解けば,

$$X = (A^T A)^{-1} A^T R \qquad (10)$$

と最小2乗解が得られる。

最小2乗法のイメージをつかんでいただくために, **付録A.** に簡単な数値例を紹介しておく。

5. 重み付き最小2乗法

今までは測定値の誤差が同じ大きさと考えて最小2乗解を求めた。しかし, 誤差があらかじめ異なることがわかっている場合もある (例えば, 精度の異なる測定器の計測値を扱う場合)。実際, GPS の擬似距離は衛星仰角の高いものほど誤差が小さく, 水平線に近い仰角の低い衛星ほど, 電離層, 対流圏, マルチパスなどの影響を受けて誤差が大きいことが知られている。このような場合, 最小2乗法に重みを導入する。重みは, 誤差分散 σ_i^2 に反比例するようにつける。

この場合, 観測値を独立と考え, 重み行列を

$$W = \begin{pmatrix} 1/\sigma_1^2 & 0 & 0 & 0 \\ 0 & 1/\sigma_2^2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 1/\sigma_n^2 \end{pmatrix}$$

とすると, 式(8)の R の重みが W であることになり,

同様に，最小2乗解

$$X = (A^T WA)^{-1} A^T WR \qquad (11)$$

を導出できる。

6. 測位計算のプログラム

ここまでの説明で，受信された衛星数個の連立方程式を立て，これを反復計算で解けば，測位値が求まることがわかった。また，反復計算で用いる修正値（ΔX）を求めるために，4衛星を選ぶ場合は，式(5)のような逆行列を使えるが，実際にはn衛星を使い式(10)の最小2乗解，または式(11)の重み付き最小2乗解を使うことを述べた。

紹介するプログラムは，n衛星の場合の測位計算である。具体的には，（**その2**）で使ったノバテル受信機の受信データから，衛星の ECEF 位置を表す **SVDA**，擬似距離を表す **RGEA** を読み込んで反復による測位計算を行う。また，受信機の計算する測位値 **POSA** を読み込んで，測位結果を比較する。プログラムの動作と関数の説明を**表3．1**に示す。main()関数を含むのは，positioning.c であり，まず入力データファイルをオープンして，**SVDA**，**RGEA**，**POSA** を1行づつ読み込む。ここまでは，（**その2**）の入力部分とほぼ同じである。次の copy_position()は重要な関数であり，今まで読み込んだ **SVDA** や **RGEA** の構造体から測位に使用する衛星の情報だけを data 型の構造体にコピーする。ノバテル受信機のマニュアルでは，「測位に使用している衛星は **SVDA** にエントリーされたものである」ことが明示されている。実際，**RGEA** には地平線から昇った直後の衛星など測位に使用できない衛星も含まれている。また，この関数内では，calc_angle_one()で PRN を与えて衛星の仰角を計算する。また，任意に与えたマスク角（mask_angle）によって，低仰角衛星を測位に使わないようにする「おまけ」機能もついている（ここで使用する入力データでは受信機のマスク角を5度に設定しているため変わりないが，c_pos.c の先頭で設定している mask_angle=5.0 を変更すればマスク角を任意に設定できる）。また，この関数では各擬似距離に **SVDA** に含まれている補正値（衛星のクロックバイアス，電離層誤差，対流圏誤差）を加えている。ノバテル受信機では，電離層誤差を放送モデルで，対流圏誤差を Hopfield モデルで計算している。以上は測位方程式をたてていることに相当する。

次にこの測位方程式を解くposdet()を実行する。関数では，**3．**で説明した式(5)のベクトルΔRを作成するr_vect()，行列Aを作成するpatrl()を実行し，最小2乗解を求める。このとき，posdet()の第2引数が最小2乗法に重みを付けるかどうかを意味する。重みなしの場合，式(10)の最小2乗解であるΔXを求め，次にΔXで初期値を更新して，収束するまで反復を繰り返せばよい。しかし，実際に正規方程式である式(10)を直接解く方法は，解が安定に求まらず精度が劣化する場合があることが知られている。この問題を回避する手段はいくつかあるようだが，その1つにハウスホルダー法と呼ばれる数値計算のアルゴリズム[3.1]が知られている。関数lls()は，文献[3.1]にあるFORTRANソースをCに変換して手直ししたものである。この関数は正規方程式を使わず，行列Aとベクトルを与えて，最小2乗解ΔXを返す。次に，重み付きの場合は，**5．**で説明した重み行列Wを作ってからlls()を使う。このとき，重み行列Wの対角項にある重みσ_iには，**SVDA**に含まれるrng_std（第14フィールド）を使うことにする。rng_stdはマニュアルにRange weight standard deviation(meter)と説明があるだけで，他に情報はない。しかし，この値はたぶん重みを意味すると仮説をたてる。次に同様に式(11)の正規方程式を直接解かずlls()を使う。このためには式(5)の観測方程式を，

$$\Delta R \rightarrow W^{1/2} \Delta R$$
$$A \rightarrow W^{1/2} A$$

と置き換えておく必要がある[3.2]。

7. ノバテル受信機の計算位置と比較する

以上の計算で得られた結果を，あらかじめ読み込んであった **POSA** と比較する（result_disp()）。計

算結果と **POSA** を ENU 座標に変換して差し引くと，その差は数 mm 程度となった。そこで全データ（約 1 日分）について，各要素の差の最大を求めてみると，(e, n, u) = (0.006, 0.004, 0.009) となり，1cm 以下であった。

また受信機の真の位置は既知のため，ENU 座標で測位誤差を求めてみる（この場合，単独測位）。すると，重みなし･重み付きの場合でそれぞれ**表3．2**のような結果となった（**POSA** の結果は重み付きと同じ）。この値は，平均的な単独測位の誤差とみなせる。従って，「重みなしの測位を行っても，十分精度の高い解が得られているが，重み付きの方が誤差を低減できる」ことがわかる。

8．おわりに（さらなる拡張に向けて）

今回はプログラムリストが大変長いものになるので，紙上には掲載できない。前回同様，WWWページ，

http://www.enri.go.jp/~fks442/K_MUSEN/

から取得してコンパイルして頂きたい。リストの内容が説明できないが，**（その1）（その2）**がコンパイルできたなら，たぶん**表3．1**の説明だけでほとんどの内容を理解できると思う。

さて，全3回の説明で，反復計算に重み付き最小2乗法を使い，ノバテル受信機と 1cm 以下の差で一致する測位結果プログラムを紹介した。「その次は？」と言えば，単独測位の精度を向上させるＤＧＰＳ（ディファレンシャルＧＰＳ）への拡張である。ディファレンシャル技術は，基準側とユーザ側の誤差の相関性を利用した古くからある基本技術であり，ＭＳＡＳやＧＢＡＳの基本原理でもある。こう書くと少し難しそうだが，実は意外と簡単！「要するに測位方程式をたてるとき，他の補正値と同様，擬似距離にディファレンシャルの補正値を足せばよい」のである。ディファレンシャル補正値を手に入れ（または既知の位置に受信機を置いて自分でつくる），プログラムを少し改良すれば，さらに一桁高い数mの誤差で測位することができる（ただし，精度の高いディファレンシャル補正値を作成するのは簡単ではない）。この他，今回のプログラムでは省略したが，衛星の配置による精度の指標としてよく用いられるＤＯＰを計算することも簡単である。さらに改造を進めれば，ＭＳＡＳやＧＢＡＳで計算されるインテグリティのパラメータ（プロテクションレベル）を計算することも可能となる。また，紹介したプログラムは数値（テキスト）を入力し，数値を出力する機能のみであるが，これをコアとしてＧＵＩ（グラフィック･ユーザ･インターフェース）を作れば，見た目にも楽しいものになるかもしれない。

これで，3回にわたり連載で進めてきた測位計算の説明を終わります。開始した当初はとても3回で済むとも思えず，多少後悔していたのですが，なんとか無事に終えられてほっと一息といったところです。はじめに書きましたように，今までの計算法の説明及び紹介したＣ言語のプログラムは全て「ＧＰＳの測位計算を理解すること」が目的です。この連載を読んで，一箇所でも「今まで何となくわかっていたが，明快に理解できた！」，「目からうろこが落ちた！」と思っていただけましたら，著者冥利につきます。内容についての質問，プログラムの拡張に関するお便りなどありましたら，前回のアドレスにメール頂きたく思います。また，追加の情報などはＷＷＷページ上に置きます。

9．謝辞

本稿で紹介したプログラムのうち lls.c は，文献[3.1]に掲載された FORTRAN ソース(Pro.8.1.1)を著者が C 言語に変換したものです。このプログラムの利用をご承諾頂いたオーム社に深く感謝致します。

また，上島一彦さん（元電子研開発部長）からは，ご質問と激励のお便りを頂きました。上島さんのご質問はいずれも的をえたもので，こちらも勉強になりました（WWWページに掲載）。また，私の下手なソースリストを丁寧に読んでいただき，Ｃ言語の表記手法についてご指導頂き，大変参考になりました。

最後に，本稿を書き始める動機を与えていただいた無線技術者の皆様，（その1）（その2）の内容についてご意見を頂いた，航空保安大学校岩沼

研修センター（衛星担当教官）の津江茂行さん，前担当教官の山内悟さん，神戸航空衛星センター（ＭＳＡＳ担当）の土肥賢一さん，常陸太田航空衛星センター（ＭＳＡＳ担当）の春木収さんに感謝します。

付録Ａ．最小２乗法の概念

４．では行列を使って最小２乗解を導いたが，わかりやすい数値例[3.3]を紹介する。

数値例：ある空港の滑走路の長さを測るため，両末端間の距離を測ったら 3000.8m であった。また，両末端のほぼ中間の地点に目印をおき，両端からその目印までの距離 x，y を測ったら，1500.1m と 1500.2m であった。目印は両末端を結ぶ直線上にあるとすれば，もっとも確からしい長さはいくらであろうか？誤差を考慮しなければ，

x=1500.1

y=1500.2

x+y=3000.8

となって，勿論答えは求まらない。そこで測定値に誤差があるとして，

$x=1500.1+\varepsilon_1$

$y=1500.2+\varepsilon_2$

$x+y=3000.8+\varepsilon_3$

と考え，４．のように誤差の２乗和を最小にする。すると，

$f=\varepsilon_1^2+\varepsilon_2^2+\varepsilon_3^2$

$=(x-1500.1)^2+(y-1500.2)^2+(x+y-3000.8)^2$

$$\partial f/\partial x = 2(x-1500.1)+2(x+y-3000.8)=0$$

$$\partial f/\partial y = 2(y-1500.2)+2(y+x-3000.8)=0$$

となり，

2x+y=4500.9

x+2y=4501.0

から，

x=1500.27，y=1500.37

と解を得る。

参考文献

[3.1] 大野豊，磯田和男監修，「新版　数値計算ハンドブック」，pp.748-752，オーム社，1990.1.

[3.2] 中川徹，小柳義夫，「最小二乗法による実験データ解析」，東京大学出版，1989.

[3.3] 中根勝見，「ＧＰＳ時代の最小２乗法　測量データの３次元処理」，東洋書店，1994

表３．１：プログラムの説明

１．**プログラム名**：positioning.c:
目的：衛星位置（SVDA），擬似距離（RGEA）から測位計算を実施し，その結果を受信機位置（POSA）と比較する
動作：
main()：メイン
　init()：初期設定
　file_open()：入力データファイルのオープン
　read_file()：
　　init_ans()：測位解の初期値を与える
　　・同時刻の SVDA，RGEA，POSA を１行だけ読む
　　get_svd_one()：SVDA を構造体へ代入
　　get_rge_one()：RGEA を構造体へ代入
　　copy_position()：測位方程式をたてる
　　calc_angle_one()：c_pos.c の関数
　　posdet()：測位方程式を解く（posdet.c を参照）
　　result_disp()：測位結果の表示と比較

２．**プログラム名**：posdet.c
目的：測位方程式を解く
関数：
posdet()：
　rvect()：ベクトルΔＲを作成
　patrl()：偏微分（α，β，γ）を計算し，行列 A を作成する
　make_weight()：重み行列 W を作成
　lls()：最小２乗解を計算
　・解（Δx，Δy，Δz）が収束するまで，rvect()に戻って反復
　・初期値から修正値である解（ΔX）を積算して，位置（P）を求める

３．**プログラム名**：lls.c
目的：ハウスホルダー変換を使った最小２乗法
関数：
　lls()：行列 A とベクトルΔＲを与え，最小２乗解

ΔXを求める[3.2]

4．プログラム名：.c_pos.c
目的：posdet()で使う関数群
関数：
copy_position()：測位計算に使用する衛星（SVDAにエントリーされている衛星）の位置，擬似距離データを揃えて，測位方程式をたてる
calc_angle_one()：PRNを与えてAZ，EL角を求める（copy_positionでマスク角を計算ため）
anglecalc()：calc_angle_one()から呼ばれ，地平線座標でAZ，EL角を計算
init_ans()：測位値の初期値を与える

表3．2：計算結果（単独）の測位誤差

単位（m）

	重みなし	重み付き
E－平均	0．115	0．187
E－σ	1．869	1．713
E－RMS	1．872	1．723
N－平均	3．335	3．012
N－σ	3．607	3．275
N－RMS	4．912	4．449
U－平均	4．007	3．601
U－σ	5．953	5．260
U－RMS	7．161	6．375