

一元化処理業務の測地系移行について

中村浩二*3・上野 寛*1・大野浩史*2・下山利浩*3・高濱 聡*3
西 政樹*3・原田智史*4・福満修一郎*3・森脇 健*3・吉田知央*3

The Change of geodetic system in the Integrated Data Analysis

Koji NAKAMURA, Hiroshi UENO, Hiroshi OHNO, Toshihiro SHIMOYAMA, Satoshi TAKAHAMA
Masaki NISHI, Satoshi HARADA, Syuichiro FUKUMITSU, Ken MORIWAKI, Tomohisa YOSHIDA

1. 世界測地系移行の概要

地球上の位置を緯度と経度で表わすための基準を測地基準系(測地系)といい、地球の形に最も近い回転楕円体で定義されている。緯度と経度は、この回転楕円体(地球楕円体)の上で表示されている。

日本では、明治時代に5万分の1地形図を作るために決定した回転楕円体(いわゆるベッセル楕円体)を今まで位置の基準としてきた。今まで使用されてきたこの測地基準系を日本測地系(Tokyo Datum)という。

一方で、近年、衛星測地技術の進歩によりGPS(全地球測位システム)による位置情報とGIS(地理情報システム)の利用が一般に普及してきた。GPSが前提とする測地基準系は、楕円体の中心が地球の重心と一致するように設定されている世界測地系である。明治時代に当時の測量技術にもとづいて設定された日本測地系の楕円体の中心は、この世界測地系の楕円体の中心と一致していない。また、日本測地系には、明治時代の測量機器や測量技術による制約と過去100年間の日本列島の地殻変動の影響等で基準点網そのものに無視できないひずみが内在していることが従来から指摘されていた。

そこで、日本の測地基準系を管理する国土地理院は、近い将来に予測されているGPSやGISの本格的普及を考慮し、同時に日本測地系のもつ基準点網のひずみを解消するために、日本の測地基準系を世界測地系に移

行することにした。測地基準系の移行のため、国土地理院では全国に電子基準点を整備し、世界測地系に基づく高精度な新しい測地基準点成果である測地成果2000(Geodetic Coordinates 2000)を作成した。また、新しい測地基準系の名称を日本測地系2000(Japanese Geodetic Datum 2000)とした。ただし、以下では単に世界測地系と記す。

世界測地系を適用するため、平成13年6月12日に測量法の一部が改正され、平成13年6月20日に公布された。この法律改正は、平成14年4月1日から施行された。また、測量法施行令の一部を改正する政令により、我が国の経度・緯度を世界測地系に変更するのに必要な数値が定められた。

気象庁の地震火山業務は、従来から日本測地系にもとづいて、観測点の位置や震源の位置などを表してきた。しかし、改正測量法の施行により日本の測地基準系が世界測地系に移行したことから、地震火山業務に使用する測地基準系を世界測地系にするべく検討と準備作業を行った。そして、平成17年2月1日に測地基準系の移行を行った。

以下に、具体的な測地基準系移行作業のうち、地震観測データの管理上、記録にとどめておくべきことを簡単に記述する。

*1 気象庁地震火山部地震津波監視課

*2 気象庁地球環境・海洋部海務課

*3 気象庁地震火山部地震予知情報課

*4 気象庁地震火山部管理課

2. 測地成果2000

国土地理院の測地成果2000は、世界測地系に基づく我が国の測地基準点（電子基準点・三角点等）成果で、従来の日本測地系に基づく測地基準点成果と区別するための呼称である。測地成果2000での経度・緯度は、世界測地系であるITRF94座標系（International Terrestrial Reference Frame：国際地球基準座標系）とGRS80（Geodetic Reference System 1980：測地基準系1980）の楕円体を使用して表す。ただし、標高については、現在と同様に東京湾平均海面を基準に表したもので変更は無い。

ITRF94座標系はInternational Terrestrial Reference Frame（国際地球基準座標系）1994の略語で、IERS（International Earth Rotation Service：国際地球回転観測事業）が構築した3次元直交座標系である。この座標系では、地球の重心に原点を置き、X軸をグリニッジ子午線と赤道との交点方向に、Y軸を東経90度方向に、Z軸を北極方向にとって、空間上の位置をX、Y、Zの数字の組で表現する。またGRS80はGeodetic Reference System（測地基準系）1980の略語で、IAG（International Association of Geodesy：国際測地学協会）及びIUGG（International Union of Geodesy and Geophysics：国際測地学及び地球物理学連合）が1979年に採択した、地球の形状、重力定数、角速度等地球の物理学的な定数及び計算式である。GRS80では、楕円体の形状や軸の方向及び地球重心を楕円体の原点とすることも定められている。この楕円体をGRS80楕円体という。

ここでは、参考のために主に地震火山業務に関連している地球の形状に関する赤道半径と扁平率の新旧比較を表1に示した。

3. 一元化処理業務の測地系移行

3. 1 測地系移行概要

気象庁の地震火山業務のうち、主な処理システムに

は、EPOS/ETOSの地震・津波の緊急処理系のシステム、一元化処理システム、そして、火山監視業務を行うVOISシステムの3系統がある。

これらのうち、EPOS/ETOSの緊急処理系のシステムについては、出される情報（地震情報、津波予報など）の震源精度が0.1度単位に丸められているため、日本測地系と世界測地系の違いは情報の品質を左右するものではない。そのため、測地系移行の費用対効果等も考慮して、今回の世界測地系移行は、一元化システムとVOISシステムのみに限ることになった。

また、EPOS/ETOSシステムの震源情報と直結している震度情報、面的震度情報等も当面は日本測地系をベースにしたものを維持することになった。これは、面的震度情報などがもとにする国土数値データ（メッシュ毎のデータ）が当面、世界測地系に移行しないことを考慮した上での判断であった。

3. 2 観測点パラメータの移行

一元化業務を世界測地系に移行するためには、まず観測点の緯度、経度を日本測地系準拠のものから世界測地系準拠のものに変えなくてはならない。移行日である平成17年2月1日時点に、気象庁が一元化業務で処理している観測点の数は、1406点にのぼる。その協力機関の内訳は、国土地理院、独立行政法人防災科学技術研究所、独立行政法人海洋研究開発機構、独立行政法人産業技術総合研究、北海道大学、弘前大学、東北大学、東京大学、名古屋大学、京都大学、高知大学、九州大学、鹿児島大学、東京都、青森県、静岡県、神奈川県温泉地学研究所、横浜市である。

これらの多数の観測点の緯度、経度の数値の世界測地系への変換自体は、国土地理院から公表されているプログラム（TKY2JGD）を利用すれば容易に行うことができる。しかしながら、これらの観測点の緯度、経度の数値はそれぞれの所属機関で管理されているものである。そこで、それぞれの観測点の所属機関に、

表1 新旧の測地系の楕円体パラメータの比較

	日本測地系	世界測地系 (日本測地系2000)
赤道半径 (m)	6377397.155	6378137
扁平率の逆数	299.152813	298.257222101

表2 震源計算に使用するパラメータの比較

	日本測地系（一元化処理で従来利用していた値）	世界測地系（日本測地系2000）
平均半径 (m)	6370291	6371009
離心率の二乗	6.674372×10^{-3}	6.694380×10^{-3}

世界測地系移行のための観測点の緯度、経度の問い合わせを行い、それぞれの機関から世界測地系での観測点の緯度、経度を入手した。

なお、観測点の標高については世界測地系移行による変化はないので、考慮していない。また、緯度、経度の世界測地系への変換は、陸地については国土地理院のプログラム、海域については海上保安庁のプログラムの使用が推奨されている。地震観測点のうち、大部分を占める陸上観測点については国土地理院のプログラムを使用した。また、海域の観測点（各機関のケーブル式海底地震計）については、設置位置の決定精度等も考慮して、陸地と同様に国土地理院のプログラムによる変換（3パラメータ変換）の結果を使用した。

3.3 プログラム等の移行

一元化処理における震源計算では、走時表を使用している。気象庁で使用している走時表JMA2001は、地球を回転楕円体として扱い、その回転楕円体内に速度構造を仮定して震央距離と震源から走時を計算している。

世界測地系は、従来の日本測地系と中心の位置が異なるだけでなく、仮定する回転楕円体の形状も異なっている。そのため、走時表の再計算が必要になった。具体的に変わるパラメータは、回転楕円体の赤道半径（長径）と扁平率であるが、震源計算に使用しているのは地球の平均半径と離心率である。これらに関する新旧の比較を表2に示す。

表2で分かるとおり、数値の変更は微小であるため、走時表も大きな変更にはならなかった。ただし、今回の走時表の再計算にあわせて、走時表の桁数を増やした（小数点以下2位までから少数点以下3位まで）。

また、同様な理由で、震源計算プログラムなどで使用する震央距離計算も、回転楕円体のパラメータ変更に影響を受ける。世界測地系移行後は、表2の数値に

もとづいた震央距離計算プログラムが使用されている。補遺に標準的な震央距離計算サブルーチン（C言語）を示した。気象庁の地震火山業務では、このサブルーチンを使用することが推奨されている。

3.4 震源データの移行

震源データは、切り替え日以降については、観測点の緯度、経度と震源計算プログラムが全て世界測地系準拠のものになっているので、その結果は世界測地系準拠のものになる。

したがって、震源データについては切り替え日以前のデータの扱いの検討が重要である。移行方法としては2通り考えられた。ひとつは移行日以降と同じ環境で、全ての震源データの再計算を行う方法。もうひとつは、観測点パラメータと同じように、単純に国土地理院の変換プログラムで震源の緯度、経度を変換する方法である。もちろん前者の方法と後者の方法で多くの場合に本質的な違いはない。しかしながら、以下の理由から厳密に数値は一致しないし、一部の事例では大きな違いが生まれる可能性がある。理由のひとつは、日本測地系と世界測地系の楕円体は位置も形状も異なるものであること。また、もう一つの理由は、従来日本測地系が内在していた測地系のゆがみの修正も行われたため、測地系による緯度、経度の差が場所によって異なることである。

いずれの方法で対応するかを検討するために、試験的に二つの方法で震源データの世界測地系対応を行った場合の差を調べた。

使用したデータは、2003年1月分の気象庁震源（K, S）合計9645個の検測値で、震源要素（緯度経度）を地理院のプログラムで変換した結果と、新測地系に基づき観測点テーブル、走時表を改訂し再計算した結果である。比較したところ、ほとんどの地震（92%以上）については誤差の範囲で一致していることが分かった。一方で、旧測地系の歪が顕著である沖縄、小笠原、

北海道の観測点を含むと思われる地震については、両者の差がやや大きくなった。緯度、経度で0.01度(0.6分)のずれがある地震は64個あった。そのうちいくつかの地震については、深さが大きく変化したり、解が求まらなかった。図1にそれらの例を示す。

この調査にもとづけば、移行方法によって一部の結果の間に無視できない差が出る可能性があることが分かった。図1などから分かる通り、測地系の違いによって震源の計算結果に無視できない大きな差が出る地震は、海域などの観測網の端で発生した地震で、もともと決定精度が低く、計算結果には不安定な要素を内在していると考えられる。したがって、仮に世界測地系で再計算したとしても、その結果の震源がより正しい位置に決まっている保証はない。むしろ、そのような不安定な事例は、過去に観測点や相の選択など特別な配慮をしてしかるべき場所に決められたものが多いと考えるべきで、このような事例は世界測地系で単純に再計算するよりは、震源の位置のみを日本測地系から世界測地系に変えた方がデータの品質低下を防ぐことができる。

そこで、世界測地系への移行方法としては、移行日

以前のものについては、国土地理院の変換プログラムで震源の緯度、経度を変換する方法を採用することとした。

なお、過去の震源のうち決定精度が低いものについては、単純に世界測地系へ数値を変換する意味合いは低く、かえってその震源データの品質に誤った情報を付加してしまう可能性もある。そのため、震源の緯度、経度を表す数値が1分の単位までの震源については、数値の変換は行わなかった。

また、P波初動極性を用いた発震機構解については、測地系移行によって震源と観測点の位置関係に変化はないと考えられるので、これによる再計算作業は行わなかった。

3. 5 そのほか

火山監視センターのシステム (VOIS) においても、平成17年2月1日に観測点の緯度、経度は世界測地系へ移行した。

EPOSで監視している埋込式体積歪計等の地殻変動の観測点についても、その観測点の緯度、経度は世界測地系へ移行した。

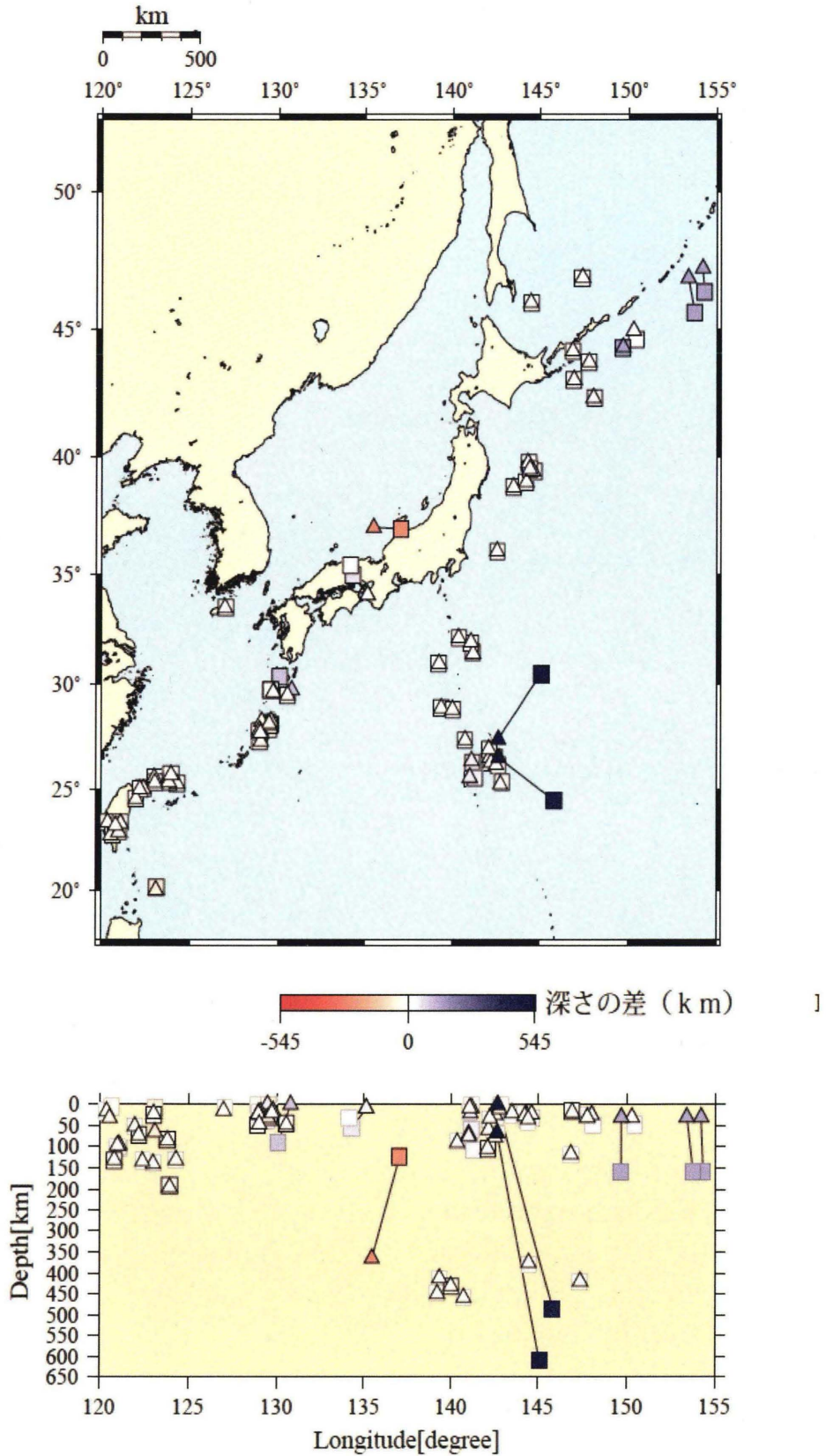


図1 測地系の以降方法によって0.01度以上の違いがある地震 (2003年1月)
 △: 震源を世界測地系に単純変換 □: 世界測地系で再計算した震源

補遺：震央距離計算プログラム

```

delta.c
/* 距離計算サブルーチン */
#include <stdio.h>
#include <math.h>
#include "hypdet.h"

/* 震央距離計算 */
/* 震央(ae,be,ce)と観測点(as,bs,cs)の距離(rad) */
double delt3_(
double *ae, /* 震央 */
double *be, /* 震央 */
double *ce, /* 震央 */
double *as, /* 観測点 */
double *bs, /* 観測点 */
double *cs) /* 観測点 */
{
    double delta;
    double cle, sle, cfe, dda;
    double ae2, be2, ce2, as2, bs2, cs2;
    double sl;

    ae2 = *ae / sqrt((*ae)*(*ae) + (*be)*(*be) + (*ce)*(*ce));
    be2 = *be / sqrt((*ae)*(*ae) + (*be)*(*be) + (*ce)*(*ce));
    ce2 = *ce / sqrt((*ae)*(*ae) + (*be)*(*be) + (*ce)*(*ce));

    as2 = *as / sqrt((*as)*(*as) + (*bs)*(*bs) + (*cs)*(*cs));
    bs2 = *bs / sqrt((*as)*(*as) + (*bs)*(*bs) + (*cs)*(*cs));
    cs2 = *cs / sqrt((*as)*(*as) + (*bs)*(*bs) + (*cs)*(*cs));

    cle = ae2 - as2;
    sle = be2 - bs2;
    cfe = ce2 - cs2;

    sl = cle*cle + sle*sle + cfe*cfe;

    dda = sl / 4.;
    dda = sqrt(dda / (1. - dda));
    delta = atan(dda) * 2.;

    return delta;
}

/* 震源距離計算 */
/* 震源(ae,be,ce,hdep)と観測点(as,bs,cs)の距離(rad) */
double dist_(
double *ae, /* 震源 */
double *be, /* 震源 */
double *ce, /* 震源 */
double *hdep, /* 震源の深さ(km) */
double *as, /* 観測点 */
double *bs, /* 観測点 */
double *cs) /* 観測点 */
{
    double dist;
    double ae0, be0, ce0;

    ae0 = *ae * (RE - *hdep) / RE;

```

一元化処理業務の測地系移行について

```

be0 = *be * (RE - *hdep) / RE;
ce0 = *ce * (RE - *hdep) / RE;

dist = sqrt((ae0 - *as) * (ae0 - *as) + (be0 - *bs) * (be0 - *bs)
+ (ce0 - *cs) * (ce0 - *cs));

return dist;
}

/* 方位角計算 */
/* 震央(ae,be,ce)から見た観測点(as,bs,cs)の方位角(deg) */
double azimpl_(
double *ae, /* 震央 */
double *be, /* 震央 */
double *ce, /* 震央 */
double *as, /* 観測点 */
double *bs, /* 観測点 */
double *cs) /* 観測点 */
{
double rddg;
double d, qa, qb;
double azimuth;

rddg = M_PI/180.;

d = (*ae) * (*as) + (*be) * (*bs) + (*ce) * (*cs);
qa = sqrt(1. - d * d);
qb = 1. - (*ce) * (*ce);
qb = sqrt(qb) * qa + (*cs) - (*ce) * d;

if(fabs(qb) < 1e-7) return 180.;

qa = ((*ae) * (*bs) - (*as) * (*be)) / qb;
azimuth = atan(qa) * 2. / rddg;
if(qa < 0.) azimuth += 360.;

return azimuth;
}

/* 震央距離計算 */
/* 震央(elat,elon)と観測点(slat,slon)の距離(km) */
double epdist(
double elat, /* 震央 */
double elon, /* 震央 */
double slat, /* 観測点 */
double slon) /* 観測点 */
{
double delta, ae, be, ce, as, bs, cs;

dtoabc_(&elat, &elon, &ae, &be, &ce);
dtoabc_(&slat, &slon, &as, &bs, &cs);
delta = delt3_(&ae, &be, &ce, &as, &bs, &cs);

return (delta * RE);
}

/* 震源距離計算 */
/* 震源(elat,elon,hdep)と観測点(slat,slon)の距離(km) */
double hypdist(
double elat, /* 震源 */

```

```

double elon, /* 震源 */
double hdep, /* 震源 */
double slat, /* 観測点 */
double slon) /* 観測点 */
{
    double dist, ae, be, ce, as, bs, cs;

    dtoabc_(&elat, &elon, &ae, &be, &ce);
    dtoabc_(&slat, &slon, &as, &bs, &cs);
    dist = dist_(&ae, &be, &ce, &hdep, &as, &bs, &cs);

    return (dist * RE);
}

/* 地理→地心緯度変換 */
double latgtoc_(
double *latg) /* 地理緯度(rad) */
{
    return (atan((1. - EC) * tan(*latg)));
}

/* 地心→地理緯度変換 */
double latctog_(
double *latc) /* 地心緯度(rad) */
{
    return (atan(tan(*latc)/(1. - EC)));
}

/* 緯度(deg),経度(deg)から方向余弦(a, b, c)を算出する */
int dtoabc_(
double *lat, /* 緯度(deg) */
double *lon, /* 経度(deg) */
double *a, /* a */
double *b, /* b */
double *c) /* c */
{
    double epa, epfa, epao;

    *a = *b = *c = 0.;

    if(*lat < -90. || *lat > 90. || *lon < -180. || *lon > 180.)
        return -1;

    epa = *lat * M_PI / 180.;
    epfa = *lon * M_PI / 180.;
    epa = latgtoc_(&epa);
    epao = cos(epa);

    *a = epao * cos(epfa);
    *b = epao * sin(epfa);
    *c = sin(epa);

    return 0;
}

```



```
hypodet.h
/* 常数,関数定義ファイル */

#ifndef M_PI
#define M_PI (3.14159265358979323846d0) /*  $\pi$  */
#endif

#ifndef BESSEL /* 旧測地系 */
#define RE (6370.291e0) /* 地球の半径 (km) */
#define EC (6.674372e-3) /* 離心率 $e^2$  */
#else /* 新測地系 */
#define RE (6371.009e0) /* 地球の半径 (km) */
#define EC (6.694380e-3) /* 離心率 $e^2$  */
#endif

/* 震央距離計算 */
double delt3(double*, double*, double*, double*, double*, double*);
double epdist(double, double, double, double);

/* 震源距離計算 */
double dist_(double*, double*, double*, double*, double*, double*, double*);
double hypdist(double, double, double, double, double);

/* 方位角計算 */
double azimpl_(double*, double*, double*, double*, double*, double*);

/* 緯度,経度 -> 方向余弦 (a, b, c) 変換 */
int dtoabc_(double*, double*, double*, double*, double*);

/* 地理→地心緯度変換 */
double latgtoc_(double*);

/* 地心→地理緯度変換 */
double latctog_(double*);
```