京都大学防災研究所年報　第 59 号 B　平成 28 年 6 月
Annuals of Disas. Prev. Res. Inst., Kyoto Univ., No. 59 B, 2016

フーリエ・ルジャンドル法の誤差特性

Error Characteristic of the Fourier–Legendre Method

榎本剛

Takeshi ENOMOTO

Synopsis

Error characteristics of the Fourier expansion of the Legendre polynomials are examined in the computation of the Gaussian latitudes and weights and in the Fourier–Legendre scaling factor for associated Legendre functions. The Fourier–Newton method yields values accurate enough for practical use up to 10000 nodes. Because the relative error grows in $\mathrm{O}(n)$ and the computational complexity in $\mathrm{O}(n^2)$, alternative methods may be required for larger calculation. There is a room for improvement in the computation of the Fourier–Legendre coefficients. The Fourier–Legendre scaling factor is not necessary in the computation of the Gaussian latitudes and weights, but for associated Legendre functions. The accuracy can be improved significantly by use of alternative equations or the asymptotic series of the Gamma function.

キーワード: ガウス緯度，ルジャンドル陪函数，フーリエ展開
Keywords: Gaussian latitudes, associated Legendre functions, Fourier expansion

1.　はじめに

球面調和函数展開は，球面上の変数を展開するために地球物理学を始め様々な分野で用いられている．これを水平離散化に用いたスペクトル変換法（Orszag, 1970）は，多くの現業数値予報モデルや気候モデルに採用されている．

近年の計算機性能の向上により現業数値予報モデルでは1000程度の切断波数が用いられ，8〜20 kmの水平解像度を実現している．切断波数3999や7999を試みたという報告（Wedi, 2013）があり，10000程度の高階高次の展開が視野に入りつつある．

ところが，球面調和函数を構成するルジャンドル陪函数は3点漸化式で計算すると，極の近くで低次の値がアンダーフローを起こすため高階高次の値を倍精度の範囲で正確に求めることができない．

Enomoto (2015)は，フーリエ展開を用いてルジャンドル多項式（0階のルジャンドル陪函数）を計算（フーリエ・ニュートン法; Swarztrauber, 2002）し，より高階高次の項は4点漸化式（Swarztrauber, 1993）を用いて求めればアンダーフローを回避し精度よく計算できることを示した．また，フーリエ・ルジャンドル法に現れる伸縮因子（フーリエ・ルジャンドル伸縮因子）をSwarztrauber (2002)の示した表式で計算すると大きな分点数で問題が生じることを改めて指摘し，問題を生じない表式を提案した．

本稿の目的は，フーリエ・ルジャンドル法を紹介し，その精度を検証することである．第2節では，Swarztrauber (1993, 2002)のフーリエ・ニュートン法と4点漸化式を紹介する．第3節では，フーリエ・ニュートン法によるガウス緯度と重みの計算（Swarztrauber, 2002）を，第4節では，フーリエ・ルジャンドル伸縮因子の精度について検証する．

2.　フーリエ・ニュートン法と4点漸化式

球面上の変数$f(\lambda,\theta)$は球面調和函数$Y_n^m(\lambda,\theta)$を用いて次のように展開できる．

$$f(\lambda,\theta)=\sum_{m,n} f_n^m Y_n^m(\lambda,\theta) \tag{1}$$

ここでλ, θはそれぞれ経度と余緯度（$\pi/2$ −緯度，北極で0，南極でπ）である．m, nはそれぞれ球面調和函数の階数と次数を表し，東西波数と全波数（東西波数+南北波数）と解される．球面調和函数$Y_n^m(\lambda,\theta)$は正規化されたルジャンドル陪函数$\tilde{P}_n^m(\cos\theta)$を用いて

$$Y_n^m(\lambda,\theta) = \tilde{P}_n^m(\cos\theta)\exp(im\lambda) \tag{2}$$

と表される．

正規化されていないルジャンドル陪函数は，Rodriguesの公式で定義される．

$$P_n^m(x) = \frac{(-1)^n}{2^n n!}(1-x^2)^{\frac{m}{2}}\frac{\mathrm{d}^{n+m}}{\mathrm{d}x^{n+m}}(1-x^2)^n \tag{3}$$

ここで$x=\cos\theta$である．正規化因子1すなわち

$$\int_{-1}^{1}\left|\tilde{P}_n^m(x)\right|^2\mathrm{d}x = 1 \tag{4}$$

であるとき，$\tilde{P}_n^m(\cos\theta)$と$P_n^m(\cos\theta)$との間には

$$\tilde{P}_n^m(\cos\theta) = \sqrt{\frac{2n+1}{2}\frac{(n-m)!}{(n+m)!}}P_n^m(\cos\theta) \tag{5}$$

という関係が成り立つ．

ルジャンドル陪函数は3点漸化式を用いて求められることが多いが，Swarztrauber (1993)は4点漸化式を用いて計算することを提案した．その正規化された表式は次の通りである．

$$\begin{aligned}\tilde{P}_n^m&(\cos\theta)\\ &= \sqrt{\frac{(2n+1)(n+m-3)(n+m-2)}{(2n-3)(n+m-1)(n+m)}}\tilde{P}_{n-2}^{m-2}(\cos\theta)\\ &- \sqrt{\frac{(n-m+1)(n-m+2)}{(n+m-1)(n+m)}}\tilde{P}_n^{m-2}(\cos\theta)\\ &+ \sqrt{\frac{(2n+1)(n-m-1)(n-m)}{(2n-3)(n+m-1)(n+m)}}\tilde{P}_{n-2}^{m}(\cos\theta)\end{aligned} \tag{6}$$

3点漸化式は同じ階数mに対して次数nが大きくなる方向に漸化式を進めるが，4点漸化式(6)では$m-2$での値を用いていることが特徴である．mが小さい方に辿っていくと，偶数のものは$\tilde{P}_n(\cos\theta)$から奇数のものは$\tilde{P}_n^1(\cos\theta)$から求められることになる．

ルジャンドル多項式$\tilde{P}_n(\cos\theta)$は

$$\begin{aligned}\tilde{P}_n&(\cos\theta)\\ &= \sqrt{\frac{2n+1}{2}}2\frac{1\cdot3\cdot5\cdots(2n-1)}{2\cdot4\cdot6\cdot6\cdots2n}\\ &\times\left[\cos n\theta + \frac{1}{1}\frac{n}{(2n-1)}\cos(n-2)\theta\right.\\ &+\frac{1\cdot3}{1\cdot2}\frac{n(n-1)}{(2n-1)(2n-3)}\cos(n-4)\theta\\ &\left.+\frac{1\cdot3\cdot5}{1\cdot2\cdot3}\frac{n(n-1)(n-2)}{(2n-1)(2n-3)(2n-5)}\cos(n-6)\theta+\cdots\right]\end{aligned} \tag{7}$$

とフーリエ展開される（Hobson ,1931）．$\tilde{P}_n^1(\cos\theta)$は

$$P_n^1(\cos\theta) = -\frac{\mathrm{d}P_n(\cos\theta)}{\mathrm{d}\theta} \tag{8}$$

と$\tilde{P}_n(\cos\theta)$と$\tilde{P}_n^1(\cos\theta)$の正規化因子の$1/\sqrt{n(n+1)}$から求められる（Belousov, 1962）．すなわち，4点漸化式(6)を用いると，ルジャンドル陪函数は全て$\tilde{P}_n(\cos\theta)$から計算されることになる。

n次のルジャンドル多項式$P_n(\cos\theta)$に対する$\cos k\theta$の係数（フーリエ・ルジャンドル係数）を$a_{n,k}$と表すことにする．$\cos n\theta$に対するフーリエ・ルジャンドル伸縮因子は次のように書ける．

$$a_{n,n} = \sqrt{\frac{2n+1}{2}}\frac{(2n-1)!!}{2^{n-1}n!} \tag{9}$$

$a_{n,n}/a_{n-1,n-1}$から

$$a_{n,n} = \sqrt{1-\frac{1}{4n^2}}a_{n-1,n-1} \tag{10}$$

が得られる．ガンマ函数を用いると

$$a_{n,n} = \sqrt{\frac{2n+1}{2}}\frac{\Gamma(2n+1)}{2^{2n-1}\Gamma^2(n+1)} \tag{11}$$

と表すこともできる．nが大きいところではStirlingの公式

$$\Gamma(n+1) = n! \approx \sqrt{2\pi n}n^n e^{-n} \tag{12}$$

を用いて

$$\begin{aligned}a_{n,n} &\approx \sqrt{\frac{2n+1}{2}}\frac{\sqrt{4\pi n}(2n)^2e^{-2n}}{2^{2n-1}2\pi n^{2n}e^{-2n}}\\ &= \sqrt{\frac{2(2n+1)}{\pi n}} = \sqrt{\frac{4}{\pi}+\frac{2}{n}}\end{aligned} \tag{13}$$

となり，$n\to\infty$のとき

$$a_{\infty,\infty} = \sqrt{\frac{4}{\pi}} \tag{14}$$

に漸近する．

$a_{n,n}$が決まれば，フーリエ・ルジャンドル係数は

$$\begin{aligned}&l(2n-l+1)a_{n,n-l}\\ &= (l-1)(2n-l+2)a_{n,n-l+2}\end{aligned} \tag{15}$$

から計算される。ここで$l=n-k$である。

3. ガウス緯度と重み

Swarztrauber (2002)は，1000点を超えてもガウス緯度を正確に計算するため，極付近に集積する$\cos\theta$に変えてルジャンドル多項式をθの函数と見てニュートン法を適用し，ルジャンドル多項式のフーリエ展開を用いるフーリエ・ニュートン法を提案した．フ

ーリエ展開の伸縮因子$a_{n,n}$は，ルジャンドル多項式の0点をニュートン法で求める際は不要なのでガウス緯度の計算には無関係である．ガウス重みは重みの和が2となるように規格化すればよい．SPHEREPACK (Adams and Swarztrauber, 1999)では，この方法を採用している．

ガウス緯度のSwarztrauber (2002)は単精度で計算されたガウス緯度と重みの精度評価をしているが，倍精度でのふるまいは明らかではない．そこで本研究では，倍精度での誤差の分点数依存性を調べた．

GawkはGNU MPFRを用いた拡張により，任意精度計算が可能である．そこで，SPHEREPACKのgaqd.fをAWKに移植した（付録）．分点数$n=92$と384の場合について有効桁数100（PREC=100）での計算を行い，Yakimiw (1996)と比較しよく一致することを確認した．次にSwarztrauber (2002)に従い，$n=256$から分点数を倍増させながら$n=16384$まで，でガウス緯度と重みを計算した．4倍精度（PREC＝QUAD）の値を真値として，倍精度での値の誤差を評価した．

ここで誤差の尺度を定義しておく．要素がb_iであるベクトルの要素の誤差が$\epsilon(b_i)$で表されるとする．相対最大誤差は，絶対誤差$|\epsilon(b_i)|$を単位ベクトルに規格化した量を表す．

$$\mathrm{rm}(b_i) = \frac{\max_i |\epsilon(b_i)|}{\max_i |b_i|} \tag{16}$$

これに対し，最大相対誤差は有効桁数の最小値の尺度である．

$$\mathrm{mr}(b_i) = \max_i \frac{|\epsilon(b_i)|}{|b_i|} \tag{17}$$

gaqdの誤差の分点数依存性をFig. 1に示す．ガウス緯度及びその余弦（それぞれ黒と青の破線，黒の破線は青の破線に隠されている）の相対最大誤差は，解像度に依存していない．ガウス重みは$n=256$でやや大きく，$n=2048$から緩やかに増大しているが，その大きさは桁が変わるほどではない．Swarztrauber (2002)が示した単精度の場合では2桁程度増大している．最大相対誤差（実線）は，どれも分点数に比例して線型に増大している．重みの最大相対誤差も分点数に依存しないHale and Townsend (2013)には劣るものの，分点数の2乗に比例する他の一般的な手法よりも精度がよい．ガウス緯度，その余弦，ガウス重みの順に誤差が大きいことはSwarztrauber (2002)が示した単精度の結果と整合的である．

ガウス緯度と重みの誤差の原因を調べるため，SPHEREPACKの伸縮因子を含まないフーリエ・ルジャンドル係数を計算するサブルーチンcpdpをGawkに移植したものを用いて誤差特性を調べた（Fig. 2）．破線は相対最大誤差で，実線は最大相対誤差を表す．

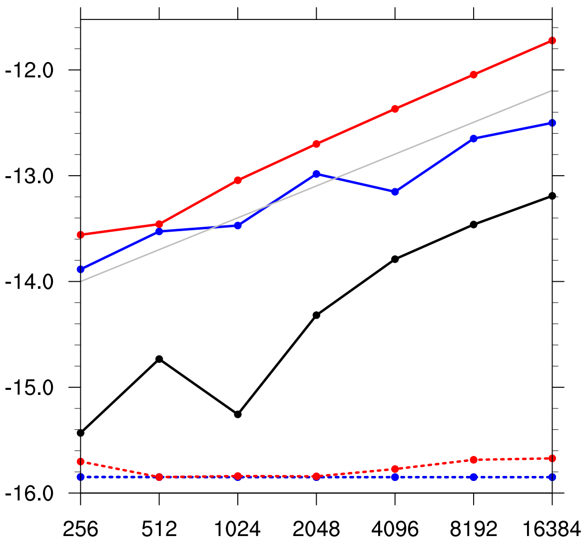

Fig. 1 The relative maximum (broken) and maximum relative (solid) error (ordinate, in common logarithm) for Gaussian colatitude θ (black), $\cos\theta$ (blue) and weights (red) computed with a Gawk version of gaqd in SPHERPACK against the number of nodes n (abscissa). The grey line represents the slope for n.

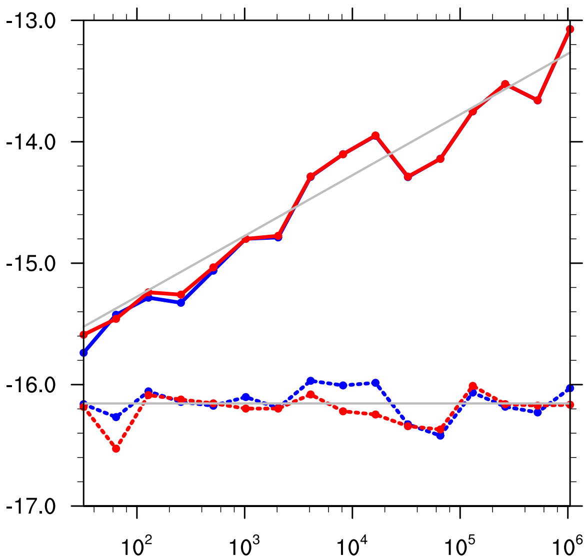

Fig. 2 As in Fig. 1 but for Fourier–Legendre coefficients $a_{n,k}$ without the scaling factor for the Legendre polynomials (blue) and their derivatives (red). The grey lines represent the slope for n and for the constant.

ルジャンドル多項式（青）とその微分（赤）とも，相対最大誤差は分点数に依存せず，最大相対誤差は分点数とともに線型に増大している．この結果からガウス緯度と重みの誤差は，フーリエ・ルジャンドル係数に起因していることが示唆される．

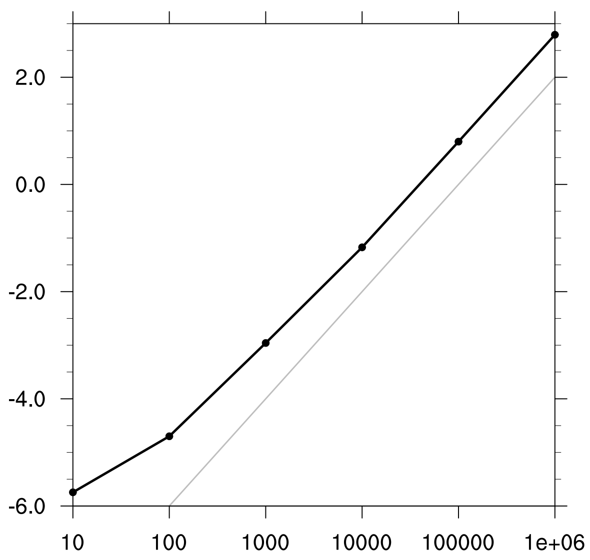

Fig. 3 As in Fig. 1 but for the wall-clock time in common logarithm using gaqd of SPHEREPACK (Fortran). The grey line represents slope for n^2.

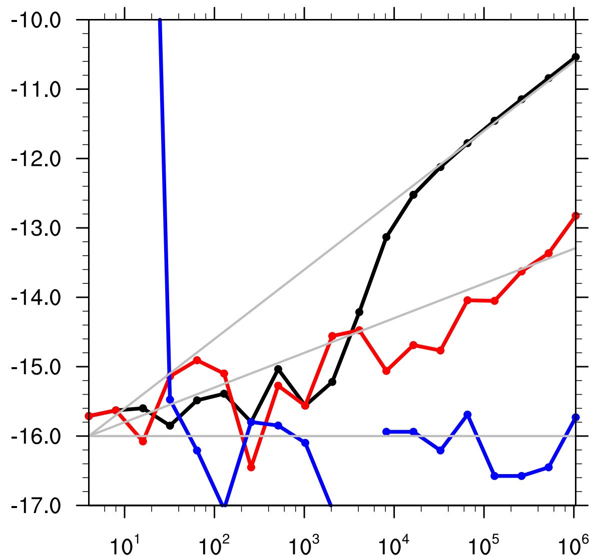

Fig. 4 As in Fig. 1 but for the Fourier–Legendre scaling factor $a_{n,n}$ for (10) (black), (19) (red) and (20) (blue). The grey lines represent the slope for n, for $\sqrt{n}$ and for the constant.

フーリエ・ルジャンドル法の計算複雑性について確認した．SPHEREPACKのgaqdを10から10^6まで桁数を変えながら実行した．最適化オプションは-O2である． gaqdを呼び出しその前後でFortran 95のcpu_time()の差により実行時間を計測した．10回の平均値をFig. 3に示す．計算複雑性はn^2であることは，灰色のn^2に傾きが同じであることから分かる．Swarztrauber (2002)では，$n = 1048576$の単精度計算に9.7時間を要したと記されている．本研究ではiMac Late 2015 (4 GHz Intel Core i7)，倍精度で$n = 1000000$が621秒であった．

4. フーリエ・ルジャンドル伸縮因子

ルジャンドル陪函数の計算には，フーリエ展開の伸縮因子$a_{n,n}$を計算する必要がある．フーリエ展開の係数$a_{n,k}$の誤差特性は，Enomoto (2015)のFig. 2に示されている．その結果を簡単にまとめておく．Swarztrauber (2002)の方法では，nが小さいところでは$\sqrt{n}$に比例して増えるが，$n = 2^{11} = 2048$から$n = 2^{14} = 16384$まで誤差がn^3に比例して急速に増大する．それより大きな分点数では，nに比例して増大する．

誤差増大の原因は，nが大きくなると根号の中で1に対して$1/4n^2$が小さくなり

$$a_{n,n} = a_{n-1,n-1} \tag{18}$$

に縮退してしまうためである．これを回避するためにEnomoto (2015)は

$$a_{n,n} = \frac{\sqrt{(2n-1)(2n+1)}}{2n} a_{n-1,n-1} \tag{19}$$

を提案した．Swarztrauber (2002)はガンマ函数の漸近展開を用いる方法を提案した．Enomoto (2015)では，$n < 128$で既存の式それよりも大きなところで$1/n^7$までのガンマ函数の漸近展開を用いている．

$$\Gamma(n+1) = \sqrt{2\pi n} n^n e^{-n} \left(1 + \frac{1}{12n} + \frac{1}{288n^2} - \frac{139}{51840n^3} - \frac{571}{2488320n^4} + \frac{163879}{209018880n^5} + \frac{5246819}{75246796800n^6} - \frac{534703531}{902961561600n^7} + \cdots\right) \tag{20}$$

フーリエ・ルジャンドル伸縮因子$a_{n,n}$の誤差特性をFig. 4に示す．黒線で示したSwarztrauber (2002)の表式(10)では，上述の誤差の急増が見られ，nの大きなところでは線型に誤差が増大しており，フーリエ係数の誤差に反映していると考えられる．これに対し，Enomoto (2015)の表式(19)では誤差は$\sqrt{n}$に比例しているが，nの小さなところでSwarztrauber (2002)の式よりも精度の悪いところが認められる．これに対し，ガンマ函数の漸近展開(20)を用いると，nが32以上で急速に誤差が減少し，分点数に依存せずに伸縮因子$a_{n,n}$が精度よく求められていることが分かる．

5. まとめ

フーリエ・ニュートン法によるガウス緯度と重みの計算法及びルジャンドル陪函数を求めるときに必

要となるフーリエ・ルジャンドル伸縮因子の精度について検証した．フーリエ・ニュートン法は分点数10000程度まで実用上十分な精度でガウス緯度と重みを計算することができる．誤差は分点数に比例して増大し，計算複雑性は分点数の2乗に比例するので，さらに分点数を大きく増やす場合は別の方法を検討する必要がある．フーリエ展開は特に問題にならないと考えられるが，フーリエ係数の計算方法は改善の余地がある．フーリエ・ルジャンドル伸縮因子は表式を工夫することにより精度を向上させることができる．分点数が100を超えたところではガンマ函数の漸近展開が分点数に依存しないので有利である．

謝　辞

本研究はJSPS科研費15K13417の助成を受けたものです．

参考文献

Adams, John C. and P. N. Swarztrauber, (1999): SPHEREPACK 3.0: a model development facility. Mon. Wea. Rev., Vol. 127, pp. 1872–1878.

Belousov, S. L., (1962): Tables of normalized associated Legendre polynomials (D E. Brown, trans). Mathematical Table Series Vol. 18, Pergamon Press, New York., pp. 384.

Enomoto, T. (2015): Comparison of computational methods of associated Legendre functions. SOLA, Vol. 11, pp. 144–149.

Hale, N. And A. Townsend, (2013): Fast and accurate computation of Gauss–Legendre and Gauss–Jacobi quadrature nodes and weigths. SIAM J. Sci. Comput., Vol. 35, pp. A652–A674.

Hobson, E. W., (1931): The theory of spherical and ellipsoidal harmonics. Cambridge University Press, Cambridge, UK., pp. 500.

Orszag, S. A. （1970） : Transform method for the calculation of vector-coupled sums: application of the vorticity equation. J. Atmos. Sci., Vol. 27, pp. 890–895.

Swarztrauber, P. N., (1993): The vector harmonic transform method for solving partial differential equations in spherical geometry. Mon. Wea. Rev., Vol. 121, pp. 3415–3437.

Swarztrauber, P. N., (2002): On computing the points and weights for Gauss–Legendre quadrature. SIAM J. Sci. Comput., Vol. 24, pp. 945–954.

Wedi, N. P., M. Hamrud, G. Mozdzynski, (2013): A fast spherical harmonics transform for global NWP and climate models. Mon. Wea. Rev., Vol. 141, pp. 3450–3461.

Yakimiw, E. (1996): Accurate computation of weights in classical Gauss–Cristoffel quadrature rules. J. Comput. Phys., Vol. 192, pp. 406–430.

付　録

SPHEREPACKのgaqdをGawkに移植したコード。フーリエ・ルジャンドル係数の計算を含むコードはhttps://git.io/vocUd参照。

```
function abs(x)
{
  return (x > 0) ? x : -x
}

function cpdp(n, cp, dcp,       t1, t2, t3, t4, ncp, i, j)
# computes the Fourier coefficients of the Legendre
polynomial
{
# p_n^0 and its derivative.
# n: degree. must be even
# cp: coefficients. 0..ncp = (n+1)/2
# dcp: coefficients. 1..ncp
  t1 = -1
  t2 = n + 1
  t3 = 0
  t4 = n + n + 1
  ncp = int((n+1)/2)
  cp[ncp] = 1
  for (i = 0; i < ncp; i++) {
    j = ncp - i
    t1++; t1++
    t2--
    t3++
    t4--; t4--
    cp[j-1] = (t1*t2)/(t3*t4) * cp[j]
  }
  for (i = 1; i <= ncp; i++) {
    dcp[i] = (i + i) * cp[i]
  }
}

function tpdp(n, theta, cp, dcp, pb, dpb,         cdt, sdt,
kdo, cth, sth, chh, i)
#   computes pn(theta) and its derivative dpb(theta)
{
  cdt = cos(theta + theta)
```

```
    sdt = sin(theta + theta)

    kdo = int(n/2)
    pb[0] = 0.5 * cp[0]
    dpb[0] = 0
    cth = cdt
    sth = sdt
    for (i = 1; i <= kdo; i++) {
        pb[0] += cp[i] * cth
        dpb[0] += -dcp[i] * sth
        chh = cdt * cth - sdt * sth
        sth = sdt * cth + cdt * sth
        cth = chh
    }
}

function dzeps(x,        a, b, c, eps)
# estimate unit roundoff in quatities of size x
{
    a = 4 / 3
    b = a - 1
    c = b + b + b
    eps = abs(c - 1)
    return eps * abs(x)
}

function gaqd(nlat, theta, wts,        eps, pi, pis2, nhalf,
zero, i, nix, zlast, zhold, zprev, ddpb, sum)
{
    eps = dzeps(1)
    eps = sqrt(eps)
    eps = eps * sqrt(eps)

    pi = atan2(0, -1)
    pis2 = 0.5 * pi
    nhalf = int(nlat/2)

    cpdp(nlat, cp, dcp)

# estimate first point next to pi/2
    zero = pis2 - 0.5 * (pi / nlat)
    for (i = 0; i < nhalf; i++) {
        nix = nhalf - i – 1

# Newton iteration
        zlast = zero
        tpdp(nlat, zero, cp, dcp, pb, dpb)
        zero += -pb[0]/dpb[0]
        while (abs(zero - zlast) > eps * abs(zero)) {
            zlast = zero
            tpdp(nlat, zero, cp, dcp, pb, dpb)
            zero += -pb[0]/dpb[0]
        }
        theta[nix] = zero
        zhold = zero

# Yakimiw's formula permits using old pb and dpb
        ddpb = dpb[0] + pb[0] * cos(zlast)/sin(zlast)
        wts[nix] = (nlat + nlat + 1) / (ddpb * ddpb)
        if (i == 0) {
            zero = 3 * zero - pi
        } else {
            zero = zero + zero - zprev
        }
       zprev = zhold
  }

# Extend points and weights via symmetries
    for (i = 0; i < nhalf; i++) {
        wts[nlat - i] = wts[i]
        theta[nlat -i] = pi - theta[i]
    }
    sum = 0
    for (i = 0; i < nlat - 1; i++) {
        sum += wts[i]
    }
    for (i = 0; i < nlat - 1; i++) {
        wts[i] *= 2 / sum
    }
}
```

（論文受理日：2016年6月13日）