Computing 32-Place Tables of Zeroes and Weights for Gauss-Legendre Quadrature

Pablo Manalastas

Retired faculty, Department of Information Systems & Computer Science
Ateneo de Manila University, Quezon City, Philippines
pmanalastas@gmail.com

Abstract. We numerically compute tables of zeroes and weights of Legendre polynomials $P_n(x)$ correct to thirty-two decimal places for $n = 2, 3, 4, \cdots, 10, 12, 16, 20, 24, 32, 40, 48, 64, 80,$ and 96. These zeroes and weights are useful in Gaussian quadrature of arbitrary continuous functions over finite intervals. We compute the zeroes of $P_n(x)$ by applying Newton's method to the equation $P_n(x) = 0$. For Newton's method, $P_n(x)$ is computed using a nonrecursive C function using a `for`-loop. Then we compute the weights w_i associated with each zero x_i of $P_n(x)$. All computations were done using `gcc` on an Intel x86_64 Linux laptop, using the `quadmath` library, in which the `__float128` data type provides 35-digit precision equivalent to true `long double`.

Keywords: Legendre polynomials, Gaussian quadrature, Newton's method

1 Legendre polynomials

The Legendre polynomial $P_n(x)$ of degree n is recursively defined as follows, where n is a positive integer.

$$P_n(x) = \begin{cases} 1 & n = 0 \\ x & n = 1 \\ ((2n-1)xP_{n-1}(x) - (n-1)P_{n-2}(x))/n & n > 1 \end{cases} \tag{1}$$

Abramowitz[1] and many others give the following Bonnett's recursion formula for Legendre polynomials.

$$\begin{aligned} P_0(x) &= 1 \\ P_1(x) &= x \\ (n+1)P_{n+1}(x) &= (2n+1)xP_n(x) - nP_{n-1}(x) \end{aligned}$$

This is equivalent to Equation (1) above. However, we have decided to use (1) in this paper because that equation is easier to rewrite as a C program (using a `for`-loop) than Bonnett's equation.

J. Caro et al. (eds.), Proceedings of the Workshop on Computation: Theory and Practice (WCTP 2024) , Atlantis Highlights in Computer Sciences 23,
https://doi.org/10.2991/978-94-6463-684-0_10

The first few Legendre polynomials are as follows.

$$
\begin{aligned}
P_0(x) &= 1 \\
P_1(x) &= x \\
P_2(x) &= (3x^2 - 1)/2 \\
P_3(x) &= (5x^3 - 3x)/2 \\
P_4(x) &= (35x^4 - 30x^2 + 3)/8 \\
P_5(x) &= (63x^5 - 70x^3 + 15x)/8 \\
P_6(x) &= (231x^6 - 315x^4 + 105x^2 - 5)/16 \\
P_7(x) &= (429x^7 - 693x^5 + 315x^3 - 35x)/16 \\
P_8(x) &= (6435x^8 - 12012x^6 + 6930x^4 - 1260x^2 + 35)/128 \\
P_9(x) &= 12155x^9 - 25740x^7 + 18018x^5 - 4620x^3 + 315x)/128 \\
P_{10}(x) &= (46189x^{10} - 109395x^8 + 90090x^6 - 30030x^4 + 3465x^2 - 63)/256
\end{aligned}
$$

Note that even degree Legendre polynomials have only even powers of x, so that when n is even, $P_n(-x) = P_n(x)$. However, odd degree Legendre polynomials have only odd powers of x, so that when n is odd, $P_n(-x) = -P_n(x)$. In both cases, if x_0 is a zero of the polynomial $P_n(x)$, then so is $-x_0$.

2 Gauss-Legendre Quadrature

Given an arbitrary function $f(x)$, Gauss-Legendre quadrature involves approximating the integral of $f(x)$ over the interval $[-1, 1]$ using a finite sum of n terms, namely,

$$\int_{-1}^{1} f(x)dx = \sum_{i=1}^{n} w_i f(x_i) + R_n \tag{2}$$

where $x_1, x_2, x_3, \cdots, x_n$ are the n zeroes of the Legendre polynomial of degree n, which are all in the interval $[-1, 1]$, and $w_1, w_2, w_3, \cdots, w_n$ are the weights associated with the x_i's, given by the formula,

$$w_i = \frac{2}{(1 - x_i^2)[P_n'(x_i)]^2} \tag{3}$$

and R_n is the error term or remainder, given by,

$$R_n = \frac{2^{2n+1}(n!)^4}{(2n+1)[(2n)!]^3} f^{(2n)}(\xi) \tag{4}$$

where $-1 < \xi < 1$. When integration is over an arbitrary closed interval $[a, b]$, the Legendre-Gauss quadrature formula is,

$$\int_a^b f(y)dy = \frac{b-a}{2}\sum_{i=1}^{n} w_i f(y_i) + R_n$$
$$y_i = \left(\frac{b-a}{2}\right)x_i + \left(\frac{b+a}{2}\right) \tag{5}$$

where x_i and w_i are the zeroes and weights of the degree n Legendre polynomial, and the remainder term is given by,

$$R_n = \frac{(b-a)^{2n+1}(n!)^4}{(2n+1)[(2n)!]^3} f^{(2n)}(\xi) \tag{6}$$

and $a < \xi < b$.

The above Equations (2), (3), (4), (5), (6) are from Abramowitz[2].

To give an idea of the size of the remainder term R_n, we computed the coefficient of $f^{(2n)}(\xi)$ and we come up with the following table of values of R_n for selected values of n.

n	R_n
3	$0.00006349206349206349 f^{(2n)}(\xi)$
4	$0.00000028794586617715 87 f^{(2n)}(\xi)$
5	$0.00000000080792891744 43 f^{(2n)}(\xi)$
6	$0.00000000000154086888 26 f^{(2n)}(\xi)$
7	$0.00000000000000212743 24 f^{(2n)}(\xi)$
8	$0.00000000000000000222 48 f^{(2n)}(\xi)$
9	$0.00000000000000000000 18 f^{(2n)}(\xi)$
10	$1.20251 \times 10^{-24} f^{(2n)}(\xi)$
12	$2.95829 \times 10^{-31} f^{(2n)}(\xi)$
16	$2.73804 \times 10^{-45} f^{(2n)}(\xi)$
20	$3.45947 \times 10^{-60} f^{(2n)}(\xi)$
24	$8.89959 \times 10^{-76} f^{(2n)}(\xi)$
32	$1.33190 \times 10^{-108} f^{(2n)}(\xi)$
40	$3.60862 \times 10^{-143} f^{(2n)}(\xi)$
48	$3.97797 \times 10^{-179} f^{(2n)}(\xi)$

If we can approximate the value of $f^{(2n)}(\xi)$, we see that Gauss-Legendre quadrature can give a pretty good approximation of the integral of $f(x)$ over the interval $[-1, 1]$.

3 Related Literature

In this paper, we shall use Newton's method to compute the zeroes of $P_n(x)$, but in Bogaert[4], series expansions are used to compute both zeroes and weights of $P_n(x)$.

Kovvali[8] states that orthogonal polynomials provide least squares approximations to continuous functions, and so are better than Newton-Cotes methods for approximate integration.

Keesling[7] states that if $f(x)$ is any polynomial of degree k up to $2n+1$, then the Gaussian Quadrature estimate of the integral of $f(x)$ is exact.

Fousse[5] gives an alternate definition of the Legendre polynomial as $P_n(x) = 2^{-n}Q_n(x^2)$ if n is even, and $P_n(x) = 2^{-n}xQ_n(x^2)$ if n is odd, and computes $P_n(x)$ through $Q_n(x)$, and subsequently gives a multi-precision method of Gaussian quadrature, using the zeroes of $P_n(x)$ computed using Newton's method.

Hale[6] gives an algorithm for computing Gauss-Legendre and Gauss-Jacobi quadrature nodes and weights based on Newton's method, but their method of computing Legendre polynomials is via the $P_n(\cos\theta)$ definition, which is different from the method used in this paper.

4 Newton's method

We shall compute the n zeroes of the Legendre polynomial $P_n(x)$ using Newton's method. Starting with an initial guess $x_0 \in [-1, 1]$, we get a new value x_1 that should be nearer to the actual zero, using Newton's formula,

$$x_1 = x_0 - \frac{P_n(x_0)}{P'_n(x_0)} \tag{7}$$

If $P_n(x_1) = 0$, or if $|P_n(x_1)| < \epsilon$ and $|x_1 - x_0| < \epsilon$, where ϵ is a preset measure of precision, say $\epsilon = 1.0 \times 10^{-33}$, we accept x_1 as a zero of $P_n(x)$, and we stop. Otherwise, we replace x_0 by x_1, and compute a new value of x_1 using Equation (7).

The flowchart of Newton's method for computing zeroes of a Legendre polynomial is given in Figure 1.

We have written a C program using `long double` data type on an Android ARM64 tablet, and an equivalent C program using the data type `__float128` on an Intel x86_64 laptop. Both programs support true `long double` precision of 35 decimal digits, which is sufficient for creating 32-place tables of zeroes and weights of the Legendre polynomials.

To compute $P_n(x)$, we derecursify Equation (1), and write it as a nonrecursive function using a `for`-loop, since for $n > 30$, recursive computation of $P_n(x)$ takes too much time. The `for`-loop works correctly, and runs in $O(n)$ time. The C code for the ARM64 is given in Figure 2.

To compute the derivative $P'_n(x)$, we use the following derivative formula from Hale[6].

$$(1 - x^2)P'_n(x) = -nxP_n(x) + nP_{n-1}(x) \tag{8}$$

It is easy to rewrite this derivative $P'_n(x)$ as a C function `P1()`, as shown in Figure 3.

Fig. 1. Flowchart of Newton's method

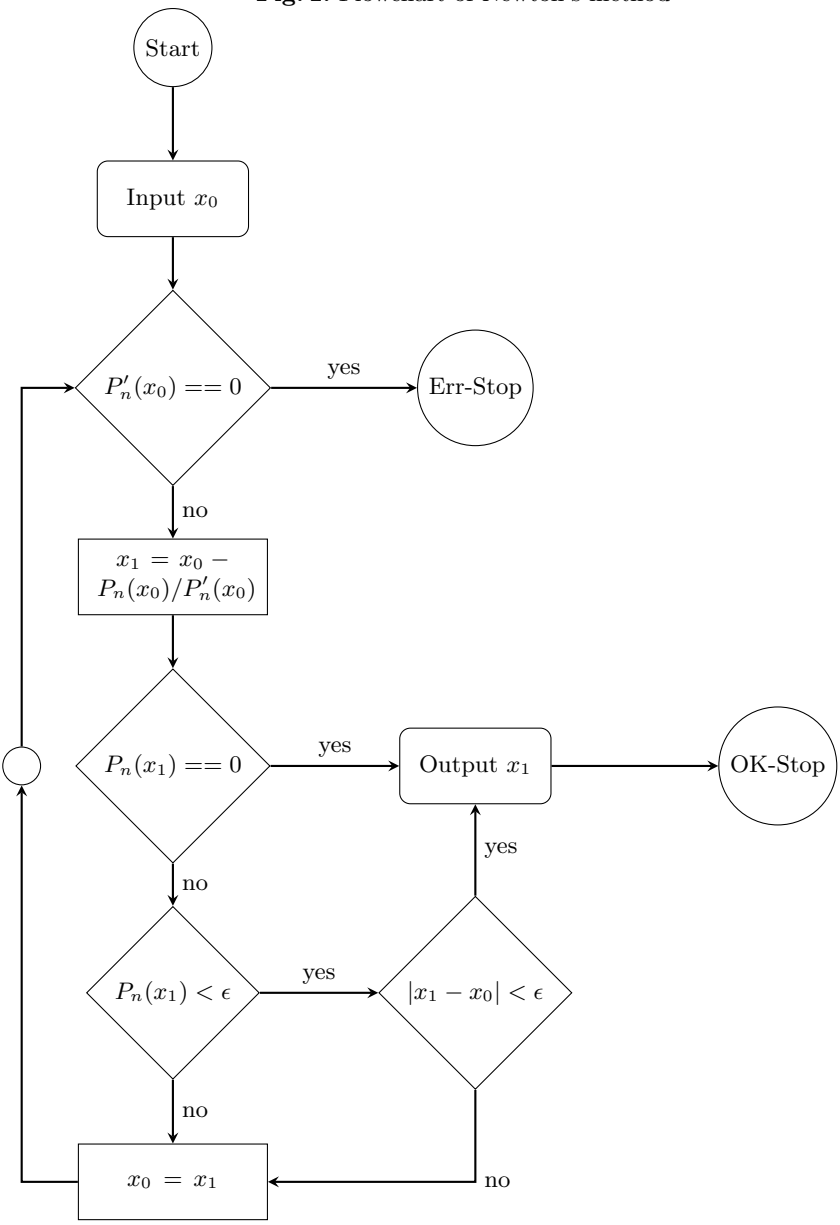

Fig. 2. $O(n)$ `for`-loop evaluation of Legendre polynomials

```
long double P(int n, long double x)
{
  long double p0, p1, pn; int k;
  if(n == 0) return 1.0L;
  if(n == 1) return x;
  if(n > 1) {
    p0 = 1.0L; p1 = x;
    for(k = 2; k <= n; ++k, p0=p1, p1=pn) {
      pn = ((2*k-1)*x*p1 - (k-1)*p0)/k;
    }
    return pn;
  }
  exit(1);
}
```

Fig. 3. $O(n)$ computation of $P_n'(x)$

```
long double P1(int n, long double x)
{
  return (-n*x*P(n,x) + n*P(n-1,x ))/(1-x*x);
}
```

To compute the weight w_i given in Equation (3) associated with each zero x_i, we just use the following C program fragment in Figure 4.

Fig. 4. $O(n)$ computation of weights w_i

```
dP = P1(n, xi);
wi = 2.0L/((1.0L-xi*xi)*dP*dP);
```

5 Computation of the 32-place tables

Abramowitz[3] gives 15-place tables of zeroes and weights of $P_n(x)$ for $n = 2, 3, 4, \cdots, 10, 12$, and 20-place tables for $n = 16, 20, 32, 40, 48, 64, 80, 96$. In this paper, we compute and present 32-place tables for all of these values of n.

Here, we compute only the positive zeros, since the negative zeroes have the same absolute values as the positive zeroes. We do not have to guess the starting values of x_0 for Newton's method, since we can get very accurate starting values from Abramowitz's tables[3]. For example, the pseudo-code for computing the 32-place zeroes of $P_{10}(x)$ is given in Figure 5.

Fig. 5. Newton's method of computing zeroes of $P_{10}(x)$

```
for( x0 in {0.148, 0.433, 0.679, 0.865, 0.973} ) {
  DoNewtonsMethod(startingValue=x0);
  Print32PlaceValueOf(x1);
}
```

The actual 32-place tables are given in Section 8 (Appendix). The first column of the tables are the zeroes, and the second column are the associated weights.

6 Example Gauss-Legendre quadrature

We want to approximate the following integral using Gauss-Legendre quadrature

$$A = \int_{-1}^{1} e^{-x^2} dx \approx 1.49364826562485405079891359230764 \tag{9}$$

We obtained the 32-place approximate value in Equation 9 by expanding the integrand into an infinite series, integrating term-by-term, and then evaluating at the limits to obtain the infinite series

$$A = 2(1 - \frac{1}{3 \cdot 1} + \frac{1}{5 \cdot 2!} - \frac{1}{7 \cdot 3!} + \frac{1}{9 \cdot 4!} - \frac{1}{11 \cdot 5!} + \cdots) \tag{10}$$

Computing 30 terms of this alternating series gives the righmost value in Equation 9. The 30th term is about 6.18030×10^{-35}, and so the 31st term must be much smaller than this, and therefore all 32 decimal places of A are correct.

The Gauss-Legendre quadrature of $f(x) = e^{-x^2}$ over the interval $[-1, 1]$ using $n = 16$ is given by the formula

$$B = \sum_{j=1}^{16} w_j f(x_j) + R_{16} \tag{11}$$

The 16 pairs of zeroes and weights are, from the table for $n = 16$ in Section 8, as follows

x_i	w_i
−0.98940093499164993259615417345033	0.027152459411754094851780572456 02
−0.94457502307323257607798841553461	0.062253523938647892862843836994 38
−0.86563120238783174388046789771239	0.095158511682492784809925107602 25
−0.75540440835500303389510119484744	0.124628971255533872052476282192 02
−0.61787624440264374844667176404879	0.149595988816576732081501730547 48
−0.45801677765722738634241944298358	0.169156519395002538189312079030 36
−0.28160355077925891323046050146050	0.182603415044923588866763667969 22
−0.09501250983763744018531933542496	0.189450610455068496285396723208 28
0.09501250983763744018531933542496	0.189450610455068496285396723208 28
0.28160355077925891323046050146050	0.182603415044923588866763667969 22
0.45801677765722738634241944298358	0.169156519395002538189312079030 36
0.61787624440264374844667176404879	0.149595988816576732081501730547 48
0.75540440835500303389510119484744	0.124628971255533872052476282192 02
0.86563120238783174388046789771239	0.095158511682492784809925107602 25
0.94457502307323257607798841553461	0.062253523938647892862843836994 38
0.98940093499164993259615417345033	0.027152459411754094851780572456 02

The remainder term R_{16} in Equation 11 is computed as follows

$$R_{16} = \frac{2^{2(16)+1}(16!)^4}{(2(16)+1)[(2(16))!]^3} f^{(2(16))}(\xi) \tag{12}$$

where $-1 < \xi < 1$. Using the `Sagemath` app, we computed $f^{32}(x)$ and determined its maximum over $[-1, 1]$ at the point $x = 0$. To maximize R_{16} we select $f^{32}(\xi)$ as follows

$$f^{32}(\xi) = \max_{-1<x<1} |f^{32}(x)| = f^{32}(0) = 1.25763 \times 10^{22} \tag{13}$$

And so $R_{16} = (2.73804 \times 10^{-45})(1.25762 \times 10^{22}) = 3.44341 \times 10^{-23}$. Thus Gauss-Legendre quadrature in Equation 11 should be correct to 22 decimal places. The actual computed value of B is $B = 1.49364826562485405079893487226371$. Now A and B agree up to the 22nd decimal place, as expected.

7 Conclusion

It is practical and efficient to compute the values of the Legendre polynomials $P_n(x)$ using the non-recursive `for`-loop in Figure 2. The computation of the derivatives $P'_n(x)$ in Figure 3 that is based on the computation of $P_n(x)$ will therefore also be efficient. Newton's method for computing roots of the Legendre polynomials, that uses the iteration

$$x_1 = x_0 - \frac{P_n(x_0)}{P'_n(x_0)}$$

is useful for efficiently creating 32-place tables of zeroes and weights for various values of n. The use of true 35-place `long double` data type in ARM64 tablets and of true 35-place `__float128` data type in Intel x86_64 laptop ensures 32-place accuracy of the computed tables. Comparison of the tables computed in this paper agree with Abramowitz's tables[3] up to the maximum 15 or 20 places available from Abramowitz. Our computed tables are provably correct to at least 32-places, since the ϵ value used as stopping condition for Newton's method is 1.0×10^{-33}.

References

1. Abramowitz, M., I.A. Stegun: Handbook of mathematical functions. Dover Publications, N.Y., (1965), National Bureau of Standards, (1964), Chap 22: Orthogonal polynomials, pp. 773-802.
2. Abramowitz, M., I.A. Stegun: Handbook of mathematical functions. Dover Publications, N.Y., (1965), National Bureau of Standards, (1964), Section 25.4.29-25.4.30: Gauss' formula, pp887-888.
3. Abramowitz, M., I.A. Stegun: Handbook of mathematical functions. Dover Publications, N.Y., (1965), National Bureau of Standards, (1964), Table 25.4: Abscissas and weight factors for Gaussian integration, pp. 916-919.
4. Bogaert, I.: Iteration-Free Computation of Gauss–Legendre Quadrature Nodes and Weights. SIAM Journal on Scientific Computing, Vol. 36, Iss. 3 (2014), pp. A1008-A1026.
5. Fousse, L.: Accurate Multiple-Precision Gauss-Legendre Quadrature. https://www.lirmm.fr/arith18/papers/fousse-GLQuad.pdf
6. Hale, N., Townsend, A.: Fast and Accurate Computation of Gauss-Legendre and Gauss-Jacobi Quadrature Nodes and Weights. SIAM Journal on Scientific Computing, Vol 35, Iss. 2 (2013), pp. A652-A674
7. Keesling, J.: Gaussian Quadrature. https://people.clas.ufl.edu/kees/files/GaussianQuadrature.pdf.
8. Kovvali, N.: Theory and applications of gaussian quadrature methods. (2011), Morgan & Claypool.
9. Wikipedia: Legendre polynomials. https://en.wikipedia.org/wiki/ Legendre_polynomials.

8 Appendix: 32-place tables of zeroes & weights for Gauss-Legendre quadrature

$n = 2$	
0.57735026918962576450914878050196	1.00000000000000000000000000000000
$n = 3$	
0.00000000000000000000000000000000	0.88888888888888888888888888888889
0.77459666924148337703585307995648	0.55555555555555555555555555555556
$n = 4$	
0.33998104358485626480266575910324	0.65214515486254614262693605077800
0.86113631159405257522394648889281	0.34785484513745385737306394922200
$n = 5$	
0.00000000000000000000000000000000	0.56888888888888888888888888888889
0.53846931010568309103631442070021	0.47862867049936646804129151483564
0.90617984593866399279762687829939	0.23692688505618908751426404071992
$n = 6$	
0.23861918608319690863050172168071	0.46791393457269104738987034398955
0.66120938646626451366139959501991	0.36076157304813860756983351383772
0.93246951420315202781230155449399	0.17132449237917034504029614217273
$n = 7$	
0.00000000000000000000000000000000	0.41795918367346938775510204081633
0.40584515137739716690660641207696	0.38183005050511894495036977548898
0.74153118559939443986386477328079	0.27970539148927666790146777142378
0.94910791234275852452618968404785	0.12948496616886969327061143267908
$n = 8$	
0.18343464249564980493947614236018	0.36268378337836198296515044927720
0.52553240991632898581773904918925	0.31370664587788728733796220198660
0.79666647741362673959155393647583	0.22238103445337447054435599442624
0.96028985649753623168356086856947	0.10122853629037625915253135430996
$n = 9$	
0.00000000000000000000000000000000	0.33023935500125976316452506928697
0.32425342340380892903853801464334	0.31234707704000284006863040658444
0.61337143270059039730870203934147	0.26061069640293546231874286941863
0.83603110732663579429942978806973	0.18064816069485740405847203124291
0.96816023950762608983557620290367	0.08127438836157441197189215811052
$n = 10$	
0.14887433898163121088482600112972	0.29552422471475287017389299465134
0.43339539412924719079926594316578	0.26926671930999635509122692156947
0.67940956829902440623432736511487	0.21908636251598204399553493422816
0.86506336668898451073209668842349	0.14945134915058059314577633965770
0.97390652851717172007796401208445	0.06667134430868813759356880989333
$n = 12$	
0.12523340851146891547244136946385	0.24914704581340278500056243604295
0.36783149899818019375269153664372	0.23349253653835480876084989892488
0.58731795428661744729670241894053	0.20316742672306592174906445580980
0.76990267419430468703689383321282	0.16007832854334622633465252954336
0.90411725637047485667846586611910	0.10693932599531843096025471819400
0.98156063424671925069054909014928	0.04717533638651182719461596148502

$n = 16$	
0.095012509837637440185319335424 96	0.189450610455068496285396723208 28
0.281603550779258913230460501460 50	0.182603415044923588866763667969 22
0.458016777657227386342419442983 58	0.169156519395002538189312079030 36
0.617876244402643748446671764048 79	0.149595988816576732081501730547 48
0.755404408355003033895101194847 44	0.124628971255533872052476282192 02
0.865631202387831743880467897712 39	0.095158511682492784809925107602 25
0.944575023073232576077988415534 61	0.062253523938647892862843836994 38
0.989400934991649932596154173450 33	0.027152459411754094851780572456 02
$n = 20$	
0.076526521133497333754640409398 84	0.152753387130725850698084331955 10
0.227785851141645078080496195368 57	0.149172986472603746787828737001 97
0.373706088715419560672548177024 93	0.142096109318382051329298325067 17
0.510867001950827098004364050955 25	0.131688638449176626898494499748 16
0.636053680726515025452836696226 29	0.118194531961518417312377377711 38
0.746331906460150792614305070355 64	0.101930119817240435036750135480 35
0.839116971822218823394529061701 52	0.083276741576704748724758143222 05
0.912234428251325905867752441203 30	0.062672048334109063569506535187 04
0.963971927277913791267666131197 28	0.040601429800386941331039952274 93
0.993128599185094924786122388471 32	0.017614007139152118311861962351 85
$n = 24$	
0.064056892862605626085043082624 75	0.127938195346752156974056165224 70
0.191118867473616309158639820757 07	0.125837456346828296121375382511 18
0.315042679696163374386793291319 81	0.121670472927803391204463153476 26
0.433793507626045138487084231913 35	0.115505668053725601353344483906 78
0.545421471388839535658375617218 37	0.107444270115965634782577342446 61
0.648093651936975569252495786910 75	0.097618652104113888269880664464 25
0.740124191578554364243828103099 98	0.086190161531953275917185202983 74
0.820001985973902921953949872669 75	0.073346481411080305734033615253 12
0.886415527004401034213154341982 20	0.059298584915436780746367758500 11
0.938274552002732758523649001708 72	0.044277438817419806168602748211 34
0.974728555971309498198391991003 17	0.028531388628933663181307815951 88
0.995187219997021360179997409700 74	0.012341229799987199546805667070 04
$n = 32$	
0.048307665687738316234812570440 50	0.096540088514727800566764830063 58
0.144471961582796493485186373598 81	0.095638720079274859419082002204 13
0.239287362252137074544603209165 50	0.093844399080804565639180237668 12
0.331868602282127649779916805730 19	0.091173878695763884712868577111 64
0.421351276130635345364119436172 43	0.087652093004403811142771462751 80
0.506899908932229390023747474377 82	0.083311924226946755222199074604 35
0.587715757240762329040745476401 83	0.078193895787070306471740918828 31
0.663044266930215200975115168663 24	0.072345794108848506225399356478 49
0.732182118740289680387426665091 27	0.065822222776361846837650063706 94
0.794483795967942406963097298970 43	0.058684093478535547145283637300 17
0.849367613732569970133693004967 74	0.050998059262376176196163244689 52
0.896321155766052123965307243719 21	0.042835898022226680656878646606 13
0.934906075937739689170919134835 41	0.034273862913021433102687732252 37
0.964762255587506430773811928118 27	0.025392065309262059455752589789 22
0.985611511545268335400175044630 90	0.016274394730905670605170562206 39
0.997263861849481563544981128665 04	0.007018610009470096600407063738 85

$n = 40$	
0.038772417506050821933193444024620	0.077505947978424811263723962958330
0.116084070675255208483451284408020	0.077039818164247965588307534283810
0.192697580701371099715516852065150	0.076110361900626242371558075922490
0.268152185007253681141184344808600	0.074723169057968264200189336261320
0.341994090825758473007492481179190	0.072886582395804059060510683442520
0.413779204371605001524879745803710	0.070611647391286779695483630855290
0.483075801686178712908566574244820	0.067912045815233903825690108231920
0.549467125095128202075931305529520	0.064804013456601038074554529566750
0.612553889667980237952612450230690	0.061306242492928939166537996408400
0.671956684614179548379354514961490	0.057439769099391551366617730910430
0.727318255189927103280996451754930	0.053227846983936824354996479772260
0.778305651426519387694971545506490	0.048695807635072232061434160448150
0.824612230833311663196320230666100	0.043870908185673271991674686041720
0.865959503212259503820781808354620	0.038782167974472017639972031290450
0.902098806968874296728253330868490	0.033460195282547847392678183086410
0.932812808278676533360852166845210	0.027937006980023401098489157507720
0.957916819213791655804540999452760	0.022245849194166957261504324184210
0.977259949983774262663370283712900	0.016421058381907888712863484882360
0.990726238699457006453054352221370	0.010498284531152813614742171067280
0.998237709710559200349622702420590	0.004521277098533191258471732878190
$n = 48$	
0.032380170962869362033322243152130	0.064737696812683922503024938736590
0.097004699209462698930053955853620	0.064466164435950082206504193657710
0.161222356068891718056437390783500	0.063924238584648186623906201825520
0.224763790394689061224865440174690	0.063114192286254025657126022750230
0.287362487355455576735886461316800	0.062039423159892663904197784137600
0.348755886292160738159817937270410	0.060704439165893880052969232027820
0.408686481990716729916225495814630	0.059114839698395635746474817433520
0.466902904750958404544928861650800	0.057277292100403215705150234684700
0.523160974722233033678225869137510	0.055199503699984162868203495191640
0.577224726083972703817809238540480	0.052890189485193667095505056264700
0.628867396776513623995164933070000	0.050359035553854474957807619087870
0.677872379632663905211851280675910	0.047616658492490474825906623478930
0.724034130923814654674482233493670	0.044674560856694280419448587125850
0.767159032515740339253855437522970	0.041545082943464749214058822361060
0.807066204029442627082553043024540	0.038241351065830706317217256523720
0.843588261624393530711089844519660	0.034777222564770438892548585963800
0.876572020274247885905693554805100	0.031167227832798088902065756846350
0.905879136715569672822074835671010	0.027426509708356948200073836262510
0.931386690706554333111417438010160	0.023570760839324379140519301378450
0.952987703160430860722960666025720	0.019616160457355527814460719652210
0.970591592546247250461411983800660	0.015579315722943848728176955834460
0.984124583722826857744583600026600	0.011477234579234539489592667609090
0.993530172266350757547928750849070	0.007327553901276262102383979621790
0.998771007252426118600541491563110	0.003153346052305838632677311543890

$n = 64$	
0.02435029266342443250895584285372	0.04869095700913972038336539073475
0.07299312178779903944954294194034	0.04857546744150342693479906678398
0.12146281929612055447037646349225	0.04834476223480295716976952715802
0.16964442042399281803731362974827	0.04799938859645830772812617987135
0.21742364374000708414964874898882	0.04754016571483030866228220694422
0.26468716220876741637396417251002	0.04696818281621001732532628575458
0.31132287199021095615751269856016	0.04628479658131441729595324923226
0.35722015833766811595044261504620	0.04549162792741814447977099697127
0.40227015796399160369576677126016	0.04459055816375656306013471003094
0.44636601725346408798494771475892	0.04358372452932345337682786097374
0.48940314570705295747852630702192	0.04247351512365358900733976790882
0.53127946401989454565801390354446	0.04126256324262352861015629747364
0.57189564620263403428387811665919	0.03995374113272034138665692612834
0.61115535517239325024885297101855	0.03855015317861562912896249694681
0.64896547125465733985776123199340	0.03705512854024004604041510180958
0.68523631305423324256355837103138	0.03547221325688238381069314671525
0.71988185017161082684894021783195	0.03380516183714160939156548211073
0.75281990726053189661186377488569	0.03205792835485155358546750434790
0.78397235894334140761022052521377	0.03023465707240247886797405981955
0.81326531512279755974192333808630	0.02833967261425948322751130520024
0.84062929625258036275169154469587	0.02637746971505465867169179262523
0.86599939815409281976078338507016	0.02435270256871087333817755040907
0.88931544599511410585340403827285	0.02227017380838325415929833038415
0.91052213707850280575638066800833	0.02013482315353020937234031672854
0.92956917213193957582149015455923	0.01795171577569734308504530200112
0.94641137485840281606248149134726	0.01572603047602471932196599529754
0.96100879965205371891861412189716	0.01346304789671864259806076668596
0.97332682778991096374185350735227	0.01116813946013112881859049301921
0.98333625388462595693129930215683	0.00884675982636394772303091465973
0.99101337147674432073938238344330	0.00650445796897836285611736039998
0.99634011677195527934692450067640	0.00414703326056246763528753572855
0.99930504173577213945690562434564	0.00178328072169643294729607914497

$n = 80$	
0.019511383256793997654351234107 45	0.039017813656306654811280439252 75
0.058504437152420668628993321883 42	0.038958395962769531198625524772 26
0.097408398441584599063278450104 94	0.038839651059051968931774182668 79
0.136164022809143886559241078000 72	0.038661759774076463327077110267 16
0.174712291832646812559339048011 29	0.038424993006959423185212436329 49
0.212994502857666132572388538666 32	0.038129711314477638344206791565 74
0.250952358392272120493158816035 00	0.037776364362001397489774976426 32
0.288528054884511853109139301434 71	0.037365490238730490026705377057 84
0.325664370747701914619112943627 36	0.036897714638276008839150996573 41
0.362304753499487315619043286358 96	0.036373749905835978043964991046 52
0.398393405881969227024379642517 53	0.035794393953416054602861588816 15
0.433875370831756093062386700363 18	0.035160529044747593495526592388 70
0.468696615170544477036078364935 81	0.034473120451753928792943642267 3103
0.502804111888784987593672750367 57	0.033733214984611522816675163064 24
0.536145920897131932019857253125 40	0.032941939397645401382836180901 96
0.568671268122709784725485786624 83	0.032100498673487773148056490287 25
0.600330622829751743154746299164 01	0.031210174188114701642442866720 60
0.631075773046871966247928387289 34	0.030272321759557980661220010090 90
0.660859898986119801735967122844 32	0.029288369583267847692767586019 58
0.689637644342027600771207612438 94	0.028259816057276862396753197965 01
0.717365185362099880254068258293 82	0.027188227500486380674418706680 54
0.744000297583597272316540527930 91	0.026075235767565117902968743600 27
0.769502420135041373865616068749 03	0.024922535764115491105117847003 22
0.793832717504605449948639311738 45	0.023731882865930101293192524613 57
0.816954138681463470371124994012 30	0.022505090246332461926221589686 17
0.838831473580255275616623043902 87	0.021244026115782006388710737250 61
0.859431406663111096977192123491 66	0.019950610878141998928891928715 11
0.878722567678213828703773343639 12	0.018626814208299031428735414152 16
0.896675579438770683194324071967 40	0.017274652056269306358584207131 29
0.913263102571757654164733656150 95	0.015896183583725688044902909229 18
0.928459877172445795953045959075 45	0.014493508040509076116962074583 46
0.942242761309872674752266004500 00	0.013068761592401339293786825897 06
0.954590766343634905493481517021 03	0.011624114120797826916466769995 43
0.965485089043799251452273155671 45	0.010161766041103064520831850352 41
0.974909140585727793385645230069 14	0.008683945269260858426409452204 03
0.982848572738629070418280277091 2	0.007192904768117312752675570867 96
0.989291302499755531026503167136 63	0.005690922451403198649269107117 16
0.994227540965688277892063503664 91	0.004180313124694895236739304201 68
0.997649864398237688899494208183 12	0.002663535895126816692935358316 7
0.999553822651630629880080499094 57	0.001144950003186941534544171941 32

$n = 96$	
0.01627674484960296957913456369524	0.03255061449236316624196141829729
0.04881298513604973111195820426128	0.03251611871386883598720549144778
0.08129749546442555899447126297475	0.03244716371406426936401278844885
0.11369585011066592091120809541433	0.03234382256857592842877483882894
0.14597371465489694198910733334333	0.03220620479403025066866711455723
0.17809688236761860275940262517649	0.03203445623199266321813897747021
0.21003131046056720360284718571230	0.03182875889441100653475373988553
0.24174315616384001232793190374063	0.03158933077072716855802074616996
0.27319881259104914148727215572456	0.03131642559686135581278426671506
0.30436494435449635302392977929520	0.03101033258631383742324977997063
0.33520852289262542261632562480563	0.03067137612366914901422883035620
0.36569686147231363503089559399547	0.03029991542082759379408876420650
0.39579764982890860328500024313516	0.02989634413632838598438807579440
0.42547898840730054536481920357000	0.02946108995816790597043633218286
0.45470942216774300863567614808631	0.02899461415055523654267878127968
0.48345797392059635976840560936390	0.02849741106508538564559951294581
0.51169417715466767358550974542885	0.02797000761684833443981857658902
0.53938810832435743622680259732556	0.02741296272602924282342108748909
0.56651041856139716840425019508346	0.02682686672559176219805672871416
0.59303236477757208068355575504376	0.02621234073567241391345796396446
0.61892584012546857038636928698786	0.02557003600534936149879716794360
0.64416340378496710679841235006330	0.02490063322248361028838218086833
0.66871831004391615395255720757277	0.02420484179236469128226733787268
0.69256453664217156134424576981824	0.02348339908592621984223593266761
0.71567681234896762622514414805762	0.02273706965832937400134784197749
0.73803064374440013285116573109650	0.02196664443874434919475638680156
0.75960234117664749870297044111962	0.02117293989219129898767386719100
0.87138850590929650287377476449304	0.01597056290256229138061645679149
0.80030874413914081722879614397133	0.01951908114014502241008522028120
0.81940031073793167553899962422448	0.01866067962741146738515675862213
0.83762351122818712149430281676469	0.01778250231604526083761422648607
0.85495903343460145546278696989357	0.01688547986424517245047754060686
0.87138850590929650287377476449304	0.01597056290256229138061645679149
0.88689451740242041605687743339379	0.01503872102699493800587627522210
0.90146063531585234131923269730969	0.01409094177231486091586162472464
0.91507142312089807420588446615042	0.01312822956696157263706366690260
0.92771245672308690964690472696555	0.01215160467108831963518135273667
0.93937033975275521693185737943162	0.01116210209983849859121326382856
0.95003271778443763575609894844386	0.01016077053500841575758763695383
0.95968829144874253930006802588473	0.00914867123078338663258460266521
0.96832682846326421217365936644365	0.00812687692569875921738242770786
0.97593917458513646645260103424188	0.00709647079115386526914416081214
0.98251726356301467744704583188255	0.00605854550423596168331674203173
0.98805412632962379948076277241845	0.00501420274292751769247019496903
0.99254390032376262457189229770160	0.00396455433844468667373341576742
0.99598184298720929065039908487936	0.00291073181793494640841061798940
0.99836437586318167772414943952672	0.00185396078894692173233592535089
0.99968950388323076682769010578437	0.00079679206555201242943814349694