รายการอ้างอิง

ภาษาไทย

ธีระพร วีระถาวร . 2531 . การอนุมานเชิงสถิติขั้นกลาง : โครงสร้างและความหมาย . กรุงเทพมหานคร : ภาควิชาสถิติ จุฬาลงกรณ์มหาวิทยาลัย.

สมเกียรติ เกตุเอี่ยม. 2528 . การเปรียบเทียบวิธีการประมาณค่าพารามิเตอร์เพื่อการทดสอบการแจกแจงของประชากรที่ให้ค่าสถิติไคสแควร์ต่ำสุด . 2528 . วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย.

ภาษาต่างประเทศ

Abdel-Aty, S. H. 1954 . Approximate formulae for the percentage points and the probability integral of the non-central χ^2 distribution, Biometrika, 41, 538-540.

Ashour, S. K. , Abdel-Samad, A. I. 1990 . On the computation of noncentral chi-square distribution function. Communications in Statistics-Simulation and Computation,19(4), 1279-1291.

Chattamvelli, R. ,Shanmugam, R. 1995 . Efficient computation of the noncentral χ^2 distribution. Communications in Statistics-Simulation and Computation, 24(3), 675-689.

Davies, R. B. 1980 . The Distribution of a Positive Linear Combination of χ^2 Random Variables. Applied Statistics, 29, 323-333.

Ding, C. G. 1990 . Computing the Non-central χ^2 Distribution Function. Applied Statistics, 41, 478-482.

Farebrother, R. W. 1984 . The Distribution of a Positive Linear Combination of χ^2 Random Variables. Applied Statistics, 33, 332-339.

Farebrother, R. W. 1987 . The Distribution of a Noncentral χ^2 Variable with Nonnegative Degrees of Freedom. Applied Statistics, 36, 402-405.

Fisher, R. A. 1928 . The general Sampling distribtion of multiple correlation coefficient, Proc. Roy. Soc. Lond., 121A, 654-673.

Johnson, N. L. and Kotz, S. 1970 . Continuous univariation distribution-2. John Wiley & Son , Inc.

Johnson, N. L. and Pearson, E. S. 1969 . Tables of pencentage points of non-cetral χ , Biometrika, 56, 255-272.

Hill, I. D. 1973 . The Normal Integral. Applied Statistics, 22, 424-427.

Han, C. P. 1975 . Some relationships between noncentral chi-square and normal distributions. Biometrika, 62, 213-214.

Lau, C.L. 1980 . A Simple series for the incomplete gamma integral. Applied Statistics, 29, 113-114.

Macleod, A. J. 1988 . A Robust and Reliable Algorithm for the logarithm of Gamma Function. Applied Statistics, 38, 397-403.

Narula, S. C. and Desu, M. M. 1981 . Computation of Probability and Non-centrality Parameter of a Non-central Chi-squared Distribution. Applied Statistics, 30, 349-352.

Patnaik, P. B. 1949 . The non-central χ^2 and F-distribution and their applications. Biometrika, 36, 203-232.

Posten, H. O. 1989 . An effective algorithm for the noncentral chi-square distributon function. American Statistcian, 43(4), 261-263.

Rice, L. R. and Gaines Das, R. E. 1985 . A remark on algorithms AS 32 and AS 147 : The incomplete gamma integral. Applied Statistics, 34, 326.

Sankaran, M. 1950 . On the non-cetral chi-square distribution, Biometrika, 46, 235-237.

Sankaran, M. 1963 . Approximations to the non-central chi-square distribution, Biometrika, 50, 199-204.

Shea, B.L. 1988 . Chi-square and Incomplete Gamma Integral. Applied Statistics, 37, 466-473.

Searle, S. R. 1971 . Linear Models. New York : John Wiley & Sons, Inc. 49 pages.

Seber, G. A. F. 1963 . The noncetral chi-square and beta distributions. Biometrika, 50, 542-544.

Tiku, M. L. 1965 . Laguere series forms of non-central χ^2 and F distributions. Biometrika, 52, 415-427.

ภาคผนวก

โปรแกรมที่ใช้ในการผลิตเลขสุ่ม

โปรแกรมหลักที่ใช้ในการผลิตเลขสุ่มนี้ประกอบด้วยโปรแกรมย่อย 2 โปรแกรม ได้แก่ โปรแกรมย่อย unifrm และโปรแกรมย่อย randu ขั้นตอนวิธีของโปรแกรมหลักอาจแสดงได้ดังนี้

```
main()
{
        int i, A, B;
        A=1;
        B=10;
        printf(" program  name : uniform.c\n");
        printf("A = %d  ; B = %d", A, B);
        scanf("%ld",&IX);
        unifrm(A, B, IX);
        getche();
        return(0);
}
```

ส่วนขั้นตอนวิธีของโปรแกรมย่อยที่ใช้ในการผลิตเลขสุ่มอาจแสดงได้ดังนี้

<u>โปรแกรมย่อย unifrm</u>

โปรแกรมนี้ทำหน้าที่สร้างเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอ และโปรแกรมนี้เรียกใช้งานโปรแกรมย่อย randu และขั้นตอนวิธีของโปรแกรมย่อย unifrm อาจแสดงได้ดังนี้

```
double unifrm(int A,int B,long int IX)/* uniform distribution*/
{
        int i;
        i=1;
        do
        {
                randu(IX);
                ix = A + (B - A) * Y;
                printf("  i = %4d ix = %12ld rn = %18.10f x = %18.10f
                        \n", i, IX, Y, ix);
                IX=IY;
                i=i+1;
        } while(i<=10);
        return(0);
}
```

<u>โปรแกรมย่อย randu</u>

โปรแกรมนี้ที่ทำหน้าที่สร้างเลขสุ่มให้กับโปรแกรมย่อย unifrm และขั้นตอนวิธีของโปรแกรมย่อย randu อาจแสดงได้ดังนี้

```
long int randu(long int IX)
{
        IY = IX * 16807;
        if (IY < 0) IY = IY + 2147483648;
        Y=IY;
        Y = Y / 2147483648;
        return(Y);
}
```

โปรแกรมที่ใช้ในการคำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลาง

โปรแกรมนี้เป็นโปรแกรมที่ใช้ในเปรียบเทียบการคำนวณค่าความคลาดเคลื่อนของฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางและการคำนวณเวลา(มิลลิวินาที)ที่ใช้ในการคำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางของโปรแกรมย่อย 4 โปรแกรม ขั้นตอนวิธีของโปรแกรมหลักอาจแสดงได้ดังนี้

```
main()
{
FILE *fp ;
long double _patnaik,_anyash,_oddash,_ruben;
long double x,n,delta,_delta,_x,_n,re_p_o,re_p_a,re_p_r,finalx,interx;
long double tp,tps,tpe;
long double ta,tas,tae;
long double to,tos,toe;
long double tr,trs,tre;
char ch;
struct tm *sys_time;
        time_t startp ,endp;
        time_t starta ,enda;
        time_t starto ,endo;
        time_t startr ,endr;
        time_t start;
struct timeb t;
fp = fopen("output.out","wt");
clrscr();
fprintf(fp,"\n");
start = time('\0');
```

```
sys_time = localtime(&start);
fprintf(fp,"                                    ");
fprintf(fp,asctime(sys_time));
fprintf(fp,"                          program  name : NCHIQN.C");
fprintf(fp,"\n");
fprintf(fp," df|delta|x|PATNAIK|ASHOUR(ANY)|");
fprintf(fp," ASHOUR(ODD)|RUBEN|%RE(P-A)|%RE(P-O)|%RE(P-R)\n");
printf("input n ");
scanf("%Lf",&n);
do
{
        printf("input delta ");
        scanf("%Lf",&delta);
        printf("input x ");
        scanf("%Lf",&x);
        printf("input finalx ");
        scanf("%Lf",&finalx);
        printf("input interx ");
        scanf("%Lf",&interx);
        for (_n=n;_n<n+1;_n=_n+2)
        {
                for (_delta=delta;_delta<delta+1;_delta=_delta+2)
                {
                        for(_x=x;_x<=finalx;_x=_x+interx)
                        {
                        ftime(&t);
                        tps = t.time*1000.00 + (t.millitm+0.00);
                        _patnaik = PATNAIK(_x,n,_delta);
```

```
endp = time('\0');
ftime(&t);
tpe = t.time*1000.00 + (t.millitm+0.00);
tp = tpe - tps;

ftime(&t);
tas = t.time*1000.00 + (t.millitm+0.00);
_anyash = ANYASH(_x,n,_delta);
ftime(&t);
tae = t.time*1000.00 + (t.millitm+0.00);
ta = tae - tas;

ftime(&t);
tos = t.time*1000.00 + (t.millitm+0.00);
_oddash= ODDASH(_x,n,_delta);
ftime(&t);
toe = t.time*1000.00 + (t.millitm+0.00);
to = toe -tos;

ftime(&t);
trs = t.time*1000.00 + (t.millitm+0.00);
_ruben= RUBEN(_x,n,_delta);
ftime(&t);
tre = t.time*1000.00 + (t.millitm+0.00);
tr = tre - trs;

re_p_a = (fabs(_patnaik - _anyash)) / _patnaik * 100;
re_p_o = (fabs(_patnaik - _oddash)) / _patnaik * 100;
```

```
            re_p_r = (fabs(_patnaik - _ruben)) / _patnaik * 100;

            fprintf(fp,"%8.4Lf | %8.4Lf | %8.4Lf | %15.8Lf | %15.8Lf |
                  %15.8Lf | %15.8Lf | %15.10Lf | %15.10Lf | %15.10Lf
                  | %Lf | %Lf | %Lf | %Lf\n"
                  ,_n, _delta, _x, _patnaik, _anyash, _oddash, _ruben,
                  re_p_a, re_p_o, re_p_r, tp, ta, to, tr);
          }
        }
        }
fprintf(fp,"\n");

/******* after print value *********/
/******* find total time *********/
ftime(&t);
tps = t.time*1000.00 + (t.millitm+0.00);
printf("%Lf \n",tps);
for (_delta=delta;_delta<delta+1;_delta=_delta+2)
{
  for(_x=x;_x<=finalx;_x=_x+interx)
        {
          _patnaik = PATNAIK(_x,n,_delta);
        }
}
ftime(&t);
tpe = t.time*1000.00 + (t.millitm+0.00);
printf("%Lf \n",tpe);
tp=tpe-tps;
```

```
ftime(&t);
tas = t.time*1000.00 + (t.millitm+0.00);
printf("%Lf \n",tas);
for (_delta=delta;_delta<delta+1;_delta=_delta+2)
{
  for(_x=x;_x<=finalx;_x=_x+interx)
        {
          _anyash = ANYASH(_x,n,_delta);
        }
}
 ftime(&t);
 tae = t.time*1000.00 + (t.millitm+0.00);
 printf("%Lf \n",tae);
 ta = tae-tas;
 ftime(&t);
 tos = t.time*1000.00 + (t.millitm+0.00);
 printf("%Lf \n",tos);

for (_delta=delta;_delta<delta+1;_delta=_delta+2)
{
  for(_x=x;_x<=finalx;_x=_x+interx)
  {
   _oddash = ODDASH(_x,n,_delta);
  }
}
  ftime(&t);
  toe = t.time*1000.00 + (t.millitm+0.00);
  printf("%Lf \n",toe);
```

```
        to=toe-tos;

        ftime(&t);
        trs = t.time*1000.00 + (t.millitm+0.00);
        printf("%Lf \n",trs);
       for (_delta=delta;_delta<delta+1;_delta=_delta+2)
       {
         for(_x=x;_x<=finalx;_x=_x+interx)
         {
                _ruben = RUBEN(_x,n,_delta);
         }
       }
        ftime(&t);
        tre = t.time*1000.00 + (t.millitm+0.00);
        printf("%Lf \n",tre);
        tr= tre-trs;

       /********* find total time *********/
       fprintf(fp,"total time require (millisec) is %10.4Lf %15.4Lf %15.4Lf
              %15.4Lf \n", tp, ta, to, tr);
       printf("***** Continue ?(y/n) *****\n");
       ch=getche();
}while ( ch != 'n' );
fclose(fp);
return(0);
}/*end main*/
```

ส่วนโปรแกรมย่อยที่ใช้ในการคำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางอาจแสดงได้ดังนี้

โปรแกรมย่อย patnaik

โปรแกรมนี้ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางของแพ็คเนด ซึ่งสามารถใช้ในการคำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลัง สองไร้ศูนย์กลางกรณีองศาความเป็นอิสระเป็นค่าใด ๆ ขั้นตอนวิธีของโปรแกรมย่อย patnaik อาจแสดงได้ดังนี้

```
long double PATNAIK(long double px,long double pn,long double pdelta)
{
long double i,fxp,patnaik;
long double x2,n2;
int fault;
        x2 = px / 2.0;
        n2 = pn / 2.0;
        fxp = gammds(x2,n2,fault);
        patnaik = chi2(x2,n2,pdelta,fxp);
        return(patnaik);
}/*end patnaik*/
```

โปรแกรมย่อย anyash

โปรแกรมนี้ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางของแอชอาวร์และแอ็บเดล-ซาแมด ซึ่งสามารถใช้ในการคำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางกรณีองศาความเป็นอิสระเป็นเลขคี่ ขั้นตอนวิธีของโปรแกรมย่อย anyash อาจแสดงได้ดังนี้

```
long double ANYASH(long double ax,long double an,long double adelta)
{
long double i,chisqn0,anyash;
long double x2,n2,aln,f;
int ifault;
        x2 = ax / 2.0;
        n2 = an / 2.0;
        aln = alngam(n2 + 1.0,ifault);
        f = exp( n2 * log(x2) - aln - x2);
        chisqn0 = gamma(x2,n2,ifault);
        anyash = f * chi1(x2,n2,adelta,chisqn0);
return(anyash);
}/*end main any ashour*/
```

โปรแกรมย่อย oddash

โปรแกรมนี้ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางของแอชอาวร์และแอ็บเดล-ซาแมด ซึ่งสามารถใช้ในการคำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางกรณีองศาความเป็นอิสระเป็นเลขคี่ ขั้นตอนวิธีของโปรแกรมย่อย oddash อาจแสดงได้ดังนี้

```
long double ODDASH(long double ox,long double on,long double odelta)
{
long double m,sumterm,aln,term,i,FBAR,delta2,odd_pd,oddash;
long double fact1,fact2,pow1,pow2,gam,pi,backterm;
int count,ifault;
        delta2 = odelta / 2.0;
        m = (on-1.0)/2.0;       /* n = 2m+1 */
        FBAR = gammds(ox/2.0,m+0.5,ifault);
```

```
        i = m;
        sumterm = 0;
        term = 0;
        count = 0;
        do
        {
                i = i + 1;
                count = count + 1.0;
                fact1 = factorial(2.0*i);
                fact2 = factorial(i);
                pow1 = Dopow(2.0,i);
                pow2 = Dopow(ox,i-0.5);
                odd_pd = fact1 / (pow1 * fact2);
                term = pow2 / odd_pd;
                gam = gammds(delta2,i-m,ifault);
                term = term * gam;
                sumterm = sumterm + term;
         } while (term >= 1.0e-5);
        backterm =  sqrt(2.0/M_PI) * exp(-ox/2.0) * sumterm;
        if (backterm < FBAR)
                oddash = FBAR - backterm;
        else
                oddash = 0.0;
        return(oddash);
} /* end main odd ashour */
```

โปรแกรมย่อย ruben

โปรแกรมนี้ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางของรูเบ็น เบิก และโกวินดาราจูลู ซึ่งสามารถใช้ในการคำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางกรณีองศาความเป็นอิสระเป็นเลขคี่ ขั้นตอนวิธีของโปรแกรมย่อย ruben อาจแสดงได้ดังนี้

```
long double RUBEN(long double rx,long double rn,long double rdelta)
{
long double z,k,j,sumterm,term,termx,termi1,termi2,termi3;
long double v,F1X,alna,alnb,expo,ruben,a,b;
long double w,tsumterm;
        k = ((rn-1)/2);  /* k is any integer number link m in odd(ashour)*/
        a = sqrt(rdelta) + sqrt(rx);
        b = sqrt(rdelta) - sqrt(rx);
        w = sqrt(rdelta*rx);
        z = rx/rdelta;
        j = 1.0;
        sumterm = 0;
        term = 0;
        termi1 = sqrt(2.0/(M_PI*w)) * (sinh(w) - cosh(w)/w);
        termi2 = sqrt(2.0/(M_PI*w)) * cosh(w);
        termi3 = 100.0;
        v = -0.5;
        if(j <= k)
        {
          while ((j <= k) && (termi3 > 0.0 )){
                termx = Dopow(z,(2.0*j-1.0)/4.0);
                termi3 = termi1 - (2.0 * v * termi2 / w);
                term = termx * termi3;
```

```
                    sumterm = sumterm + term;
                    if (termi3 > 0.00) tsumterm = sumterm;
                    termi1 = termi2;
                    termi2 = termi3;
                    v = v + 1.0;
                    j = j + 1.0;
                  }
            }
      alna = alnorm(a,0);
      alnb = alnorm(b,0);
      F1X = alna - alnb;
      expo = exp(-(rdelta+rx)/2.0);
      if (expo*sumterm > 0.00 && expo*sumterm < 1.0)
            ruben = F1X - (expo * sumterm);
      else
            ruben = F1X - (expo * tsumterm);
      return(ruben);
      } /* end ruben */
```

ฟังก์ชันที่ใช้ในการคำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลาง

ฟังก์ชันที่ใช้ในการคำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลาง ประกอบด้วย ฟังก์ชันดังต่อไปนี้

ฟังก์ชัน alnorm

ฟังก์ชันนี้ทำหน้าที่คำนวณค่าการแจกแจงปกติสะสม ซึ่งโปรแกรมย่อยที่เรียกใช้งานฟังก์ชันนี้ได้แก่ โปรแกรมย่อย ruben และขั้นตอนวิธีของฟังก์ชัน alnorm อาจแสดงได้ดังนี้

```
long double alnorm(long double XVALUE,int UPPER)
{
const long double ltone = 7.0;
const long double utzero = 18.66;
const long double con = 1.28;
long double z,y,alnorm;
int UP;
        UP = UPPER;
        z = XVALUE;
        if(z >= 0)
         {
                if(z <= ltone || UP && z <= utzero)
                {
                        y = 0.5 * z * z;
                        if( z > con )
                        {
                                alnorm = 0.398942280385 * exp(-y) /
                                        ( z - 3.8052e-8 + 1.00000615302 /
                                        ( z + 3.98064794e-4 +1.98615381364 /
                                        ( z -0.151679116635 +5.29330324926 /
                                        ( z + 4.8385912808 - 15.1508972451 /
                                        ( z + 0.742380924027+ 30.789933034/
                                        ( z + 3.99019417011 ))))));
                                if (!UP) alnorm = 1 - alnorm;
                                        return(alnorm);
                }
                else
                {
```

```
                alnorm = 0.5 - z * (0.398942280444 - 0.399903438504 * y /
                        ( y + 5.75885480458 - 29.8213557808 /
                        ( y + 2.62433121679 + 48.6959930692 /
                        ( y + 5.92885724438 ))));
                if (!UP) alnorm = 1 - alnorm;
                        return(alnorm);
                }
        }
        else
        {
                alnorm = 0;
                if (!UP) alnorm = 1 - alnorm;
                        return(alnorm);
        }
}
else
{
        UP = !UP;
        z = -z;
        if(z <= ltone || UP && z <= utzero)
        {
                y = 0.5 * z * z;
                if( z > con )
                {
                        alnorm = 0.398942280385 * exp(-y) /
                                ( z - 3.8052e-8 + 1.00000615302 /
                                ( z + 3.98064794e-4 + 1.98615381364 /
                                ( z - 0.151679116635 + 5.29330324926 /
```

```
                    ( z + 4.8385912808 - 15.1508972451 /
                    ( z + 0.742380924027 + 30.789933034 /
                    ( z + 3.99019417011 ))))));
                if (!UP) alnorm = 1 - alnorm;
                    return(alnorm);
            }
            else
            {
            alnorm = 0.5 - z * (0.398942280444 - 0.399903438504 * y /
                    ( y + 5.75885480458 - 29.8213557808 /
                    ( y + 2.62433121679 + 48.6959930692 /
                    ( y + 5.92885724438 ))));
            if (!UP) alnorm = 1 - alnorm;
                    return(alnorm);
            }
        }
        else
        {
            alnorm = 0;
            if (!UP) alnorm = 1 - alnorm;
                    return(alnorm);
        }
    }
} /*end function*/
```

ฟังก์ชัน alngam

ฟังก์ชันนี้ทำหน้าที่คำนวณค่าล็อกกาลิทึมของฟังก์ชันการแจกแจงแกมมา ซึ่งโปรแกรมย่อยที่เรียกใช้งานฟังก์ชันนี้ได้แก่ โปรแกรมย่อย patnaik โปรแกรมย่อย anyash และโปรแกรมย่อย oddash และขั้นตอนวิธีของฟังก์ชัน alngam อาจแสดงได้ดังนี้

```
long double alngam(long double xvalue,int ifault)
{
long double R1[] ={ 0,
 -2.66685511495e0, -2.44387534237e1, -2.19698958928e1,
  1.11667541262e1,  3.13060547623e0,  6.07771387771e-1,
  1.19400905721e1,  3.14690115749e1,  1.52346874070e1
  };
long double R2[] ={ 0,
 -7.83359299449e1, -1.42046296688e2,  1.37519416416e2,
  7.86994924154e1,  4.16438922228e0,  4.70668766060e1,
  3.13399215894e2,  2.63505074721e2,  4.33400022514e1
  };
long double R3[] ={ 0,
 -2.12159572323e5,  2.30661510616e5,  2.74647644705e4,
 -4.02621119975e4, -2.29660729780e3, -1.16328495004e5,
 -1.46025937511e5, -2.42357409629e4, -5.70691009324e2, /**/
  };
long double R4[] ={ 0,
  2.79195317918525e-1,  4.917317610505968e-1,  6.92910599291889e-2,
  3.350343815022304e0,  6.012459259764103e0
  };
const long double alr2pi = 9.18938533204673e-1;
const long double xlge = 2.8e2;
const long double xlgst = 2.0e36;
```

```
long double _x,x1,x2,y,alngam;
        _x = xvalue;
        alngam = 0.0;
        ifault = 2;
        if (_x >= xlgst)
                return(ifault);
        ifault = 1;
        if (_x <= 0.0)
                return(ifault);
        else /* (_x<=0.0) */
        {
                ifault = 0;
                if (_x < 1.5)
                {
                        if (_x < 0.5)
                        {
                                alngam = -log(_x);
                                y = _x + 1.0;
                                if(!(y != 1.0))
                                        return(alngam);
                        }
                    else
                    {
                        alngam = 0.0;
                        y = _x;
                        _x = (_x - 0.5) - 0.5;
                    }
                    alngam = alngam + _x * ((((R1[5] * y + R1[4]) * y + R1[3])
```

```
        * y + R1[2]) * y + R1[1]) / ((((y + R1[9]) * y + R1[8])
        * y + R1[7]) * y + R1[6]);
    return(alngam);
}
else /* (_x<1.5) */
{
  if ( _x < 4.0 )
  {
          y = (_x - 1.0) - 1.0;
           alngam = y * ((((R2[5] * _x + R2[4]) * _x + R2[3])
                  * _x + R2[2])  * _x + R2[1]) /
                       ((((_x + R2[9]) * _x + R2[8]) * _x +
                  R2[7]) * _x + R2[6]);
    return(alngam);
  }
  else /* (_x<4.0) */
  {
if ( _x < 12.0 )
    {
        alngam = ((((R3[5] * _x + R3[4]) * _x + R3[3]) * _x +
          R3[2]) * _x + R3[1]) / ((((_x + R3[9]) * _x +
          R3[8]) * _x + R3[7]) * _x + R3[6]);
       return(alngam);
    }
    else
    {
            y = log(_x);
            alngam = _x * (y - 1.0) - 0.5 * y + alr2pi;
```

```
                    if ( _x > xlge )
                            return(alngam);  /*return alngam*/
                    else
                    {
                            x1 = 1.0 / _x;
                            x2 = x1 * x1;
                             alngam = alngam + x1 * ((R4[3] * x2
                               + R4[2]) * x2 + R4[1]) /
                                       ((x2 + R4[5]) * x2 + R4[4]);
                            return(alngam); /*alngam*/
                    }
                }
            }
        } /* (_x<1.5) */
    }/* (_x<=0.0) */
} /* end function */
```

<u>ฟังก์ชัน gammds</u>

ฟังก์ชันนี้ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงแกมมาโดยที่ฟังก์ชันนี้เรียกใช้งานฟังก์ชัน alngam ด้วย ซึ่งโปรแกรมย่อยที่เรียกใช้งานฟังก์ชันนี้ได้แก่ โปรแกรมย่อย patnaik และโปรแกรมย่อย oddash และขั้นตอนวิธีของฟังก์ชัน gammds อาจแสดงได้ดังนี้

```
long double gammds(long double X,long double df,int ifault)
{
long double a,c,gammds,aln,f;
        ifault = 1;
         gammds = 0.0;
         if ( X <= 0.0 || df <= 0.0 )        return (ifault = 1);
```

```
        else
          {
          ifault = 2;
          aln = alngam(df+1.0,ifault);
          f = exp( df * log(X) - aln - X);
          }
        if ( f > 0.0)
          {
          ifault = 0;
          c = 1.0;
          gammds = 1.0;
          a = df;
          do
          {
                  a = ++a;
                  c = c * X / a;
                  gammds = gammds + c;
          } while (c / gammds > 1.0e-5);
    gammds = gammds * f;
    return(gammds);
  }
 else
   return (ifault = 2);
}      /*end function*/
```

<u>ฟังก์ชัน gamma</u>

ฟังก์ชันนี้ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงแกมมา ซึ่งโปรแกรมย่อยที่เรียกใช้งานฟังก์ชันนี้ได้แก่ โปรแกรมย่อย anyash และขั้นตอนวิธีของฟังก์ชัน gamma อาจแสดงได้ดังนี้

```
long double gamma(long double X,long double df,int ifault)
{
long double b,c,gamma,aln,f;
        ifault = 1;
        gamma = 0.0;
        if ( X <= 0.0 || df <= 0.0 )
          return (ifault=1);
        else
          {
          ifault = 0;
          c = 1.0;
          gamma = 1.0;
          b = df;
          do
          {
                    b = ++b;
                    c = c * X / b;
                    gamma = gamma + c;
          } while (c / gamma > 1.0e-5);
          return(gamma);
   }
}     /*end function*/
```

ฟังก์ชัน chil

ฟังก์ชันนี้ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงไคกำลังสอง ซึ่งโปรแกรมย่อยที่เรียกใช้งานฟังก์ชันนี้ได้แก่ โปรแกรมย่อย anyash และขั้นตอนวิธีของฟังก์ชัน chil อาจแสดงได้ดังนี้

```
long double chi1(long double _X,long double DF,long double DELTA,long
double chisqn0)
{
long double chi1,df1,delta2,T,term,z,d,a;
int ifault;
        chi1 = chisqn0;
        df1 = DF;
        delta2 = DELTA / 2.0;
        z = _X * delta2;
        d = 1.0;
        T = 0.0;
        a = DF;
        do
        {
                a = a + 1.0;
                T = T + 1.0;
                d = d * z / (a * T);
                df1 = df1 + 1.0;
                term = d * gamma(_X,df1,ifault);
                chi1 = chi1 + term;
        } while (term >= 1.0e-5);
        chi1 = chi1 * exp(-delta2);
        return(chi1);
} /* end chi1 */
```

ฟังก์ชัน chi2

ฟังก์ชันนี้ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงไคกำลังสอง ซึ่งโปรแกรมย่อยที่เรียกใช้งานฟังก์ชันนี้ได้แก่ โปรแกรมย่อย patnaik และขั้นตอนวิธีของฟังก์ชัน chi2 อาจแสดงได้ดังนี้

```
long double chi2(long double _X,long double DF,long double DELTA,long
double FXP)
{
long double chi2,df1,delta2,c,T,term;
int ifault;
        chi2 = FXP;
        df1 = DF;
        delta2 = DELTA / 2.0;
        c = 1.0;
        T = 0.0;
        do
        {
                T = ++T;
                c = c * delta2 / T;
                df1 = df1 + 1.0;
                term = c * gammds(_X,df1,ifault);
                chi2 = chi2 + term;
         } while (term >= 1.0e-5);
 chi2 = chi2 * exp(-delta2);
 return(chi2);
} /* end chi2 */
```

ฟังก์ชันที่ใช้ในการคำนวณค่าทางคณิตศาสตร์ ประกอบด้วย 2 ฟังก์ชัน ดังนี้

1. ฟังก์ชัน factorail ฟังก์ชันนี้เป็นฟังก์ชันที่ใช้ในการคำนวณค่าแฟคทอเรียล
2. ฟังก์ชัน Dopow ฟังก์ชันนี้เป็นฟังก์ชันที่ใช้ในการคำนวณค่าเลขยกกำลัง

ฟังก์ชัน factorial ฟังก์ชันนี้เป็นฟังก์ชันที่ใช้ในการคำนวณค่าแฟคทอเรียล อาจแสดงได้ดังนี้

```
long double factorial(long double num)
{
if (num < 1.0 )
  return(1.0);
else
  return(num * factorial(num - 1.0));
}
```

ฟังก์ชัน Dopow ฟังก์ชันนี้เป็นฟังก์ชันที่ใช้ในการคำนวณค่าเลขยกกำลัง อาจแสดงได้ดังนี้

```
long double Dopow(long double x,long double y)
{
long double Y;
if( y < 1)
return(pow(x,y));
else
 return(x*Dopow(x,y-1));
 }
```

ประวัติผู้เขียน

นางสาววนิดา จงทำดีสกุล เกิดเมื่อวันที่ 12 เมษายน พ.ศ. 2515 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาวิทยาศาสตร์บัณฑิต สาขาสถิติ จากคณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ในปีการศึกษา 2536 และได้เข้าศึกษาต่อในหลักสูตรสถิติศาสตร์มหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2537