N 70 12947

NASA TECHNICAL MEMORANDUM

NASA TM X-52728

NASA TM X-52728

CASE FILE COPY

DIRECT AND INDIRECT EVALUATION OF THE CUMULATIVE BINOMIAL FUNCTION ON A DESK-TOP-TYPE ELECTRONIC COMPUTOR

by Darl D. Bien and Robert E. English
Lewis Research Center
Cleveland, Ohio
December 1969

E- 5335

DIRECT AND INDIRECT EVALUATION OF THE CUMULATIVE BINOMIAL FUNCTION ON A DESK-TOP-TYPE ELECTRONIC COMPUTOR

by Darl D. Bien and Robert E. English

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

DIRECT AND INDIRECT EVALUATION OF THE CUMULATIVE BINOMIAL FUNCTION ON A DESK-TOP-TYPE ELECTRONIC COMPUTOR

by Darl D. Bien and Robert E. English

Lewis Research Center

SUMMARY

Two programs for the cumulative binomial function

$$C = \sum_{j=r}^{n} \frac{n!}{j!(n - j)!} p^j(1 - p)^{n-j}$$

are presented. The direct program computes C for specified values of n, r, and p. The indirect program computes p for specified values of n, r, and C. The programs, which are suitable for $n \leq 1000$, are useful in applications such as reliability, industrial sampling, and statistics.

INTRODUCTION

Calculation of probabilities using the cumulative binomial function is useful in such applications as reliability, industrial sampling, statistical-hypothesis testing, and confidence-interval determination. The cumulative binomial function is

$$C = \sum_{j=r}^{n} \frac{n!}{j!(n - j)!} p^j(1 - p)^{n-j}$$

Two calculator programs are presented for the occasional user of the binomial function who has available a Hewlett-Packard 9100 A desk-top calculator. The direct program computes the cumulative binomial sum C for given values of n, r, and p. The indirect program uses an iterative procedure to determine the value of the binomial parameter p for specified values of n, r, and C.

Instructions for use of the programs are given along with a discussion of limitations and applications. A more thorough treatment of the indirect problem is presented in the reference along with extensive tables of p and several applications.

SYMBOLS

C cumulative binomial probability

$\hat{C}_i$ cumulative binomial sum using p_i as the binomial parameter

j summation variable

n upper limit on binomial summation

p binomial parameter; element success probability

r lower limit on binomial summation

Subscripts:

f final value

i iteration number

0 initial value

THEORY

The probability C of at least r successes among n independent identical elements of a system is given by the cumulative binomial function

$$C = \sum_{j=1}^{n} \binom{n}{j} p^j (1 - p)^{n-j} \tag{1}$$

where

$$\binom{n}{j} = \frac{n!}{j!(n - j)!} \tag{2}$$

and p is the element success probability. The negative binomial distribution is equivalent to equation (1); its equation is

$$C = p^r \sum_{j=0}^{n-r} \binom{r + j - 1}{j} (1 - p)^j \tag{3}$$

Equation (3) is used in both the direct and indirect programs because it requires fewer program steps than equation (1). In the expansion of the summation, the first term is always unity. Successive terms of the summation are obtained by multiplication by

$$\frac{r + j - 1}{j} (1 - p) \tag{4}$$

DIRECT PROGRAM

The function of the direct program is to determine the value of C for given values of n, r, and p by summing the terms of equation (3). The user instructions and program steps are described in figure 1. The three inputs n, r, and p are loaded by the user into the z, y, and x registers, respectively, before beginning execution. After loading, the user must press "Go to (0)(0)" and "Continue." The display is C in the x register and zeros in y and z.

The following sample cases illustrate the use of the direct program. The times given are the approximate delays between pressing "Continue" and the display of the result.

Case 1:	$n = 10$	$C = 0.967\ 206\ 502$
	$r = 6$	
	$p = 0.8$	less than 1 second
Case 2:	$n = 100$	$C = 0.999\ 192\ 426$
	$r = 80$	
	$p = 0.9$	about 2 seconds
Case 3:	$n = 1000$	$C = 0.526\ 599\ 080$
	$r = 900$	
	$p = 0.9$	about 6 seconds

Enter program: (Starting address is 0 - 0)

Enter data: $n \rightarrow z$, $r \rightarrow y$, $p \rightarrow x$

Press: Go to (0)(0) or End

Press: Continue

Display:

0 —— z

0 —— y

C —— x

(a) User instructions.

Step	Key	Code	Display x	Display y	Display z	Remarks
0-0	x→( )	23	p	r	n	Store p
0-1	b	14				
0-2	y→( )	40				Store r
0-3	c	16				
0-4	Roll ↓	31				
0-5	If x>y	53				Test if r > n and, if so, force error light to come on
0-6	Clear x	37				
0-7	÷	35				
0-8	-	34				
0-9	y→( )	40				Compute and store n - r
0-a	d	17				
0-b	↓	25				
0-c	.	21				
0-d	5	05				Test if p is outside the interval 0 to 1 and, if so, force error light to come on
1-0	-	34				
1-1	|y|	55				
1-2	If x<y	52				
1-3	Clear x	37				
1-4	÷	35				
1-5	Clear	20				Enter 1's in f and e
1-6	1	01				
1-7	↑	27				
1-8	Acc +	60				
1-9	↓	25				Skip to 4 - 0 if r = n
1-a	d	17				
1-b	If x = y	50				
1-c	4	04				
1-d	0	00				To 4 - 0
2-0	1	01				From 3 - d; Increment counter
2-1	+	33				
2-2	y→( )	40				Store counter
2-3	a	13				
2-4	c	16				Compute successive term of binomial summation
2-5	+	33				
2-6	1	01				
2-7	-	34				
2-8	a	13				
2-9	÷	35				
2-a	1	01				
2-b	↑	27				
2-c	b	14				
2-d	-	34				

Step	Key	Code	Display x	Display y	Display z	Remarks
3-0	↓	25				
3-1	×	36				
3-2	f	15				
3-3	×	36				
3-4	Clear x	37				Store sum and individual term
3-5	Acc +	60				
3-6	y→( )	40				
3-7	f	15				
3-8	a	13				Test if all terms have been calculated
3-9	↑	27				
3-a	d	17				
3-b	If x > y	53				
3-c	2	02				
3-d	0	00				To 2 - 0
4-0	b	14				From 1-d; Compute p^r
4-1	ln x	65				
4-2	↑	27				
4-3	c	16				
4-4	×	36				
4-5	↓	25				
4-6	e^x	74				
4-7	↑	27				Form binomial sum
4-8	e	12				
4-9	×	36				
4-a	y→( )	40				Store C
4-b	a	13				
4-c	Clear	20				
4-d	a	13				
5-0	Stop	41	C	0	0	
5-1	End	46				
5-2 to 5-d	Unused					Blocks 6 to 9 are unused; blocks a to f are used for storage

(b) Program steps.

Figure 1. - Direct-program user instructions and program steps.

Limitations

The direct program incorrectly sets C to zero whenever p^r is less than 10^{-99}, the limit of the calculator. This occurs because C is proportional to p^r (see eq. (3)). Figure 2 shows the allowable pairs of p and r for r from 10 to 1000. The curve separating the allowable and unallowable regions is the equation $p^r = 10^{-99}$. When the calculator displays either C = 0 or C = 1, the input data should be checked.

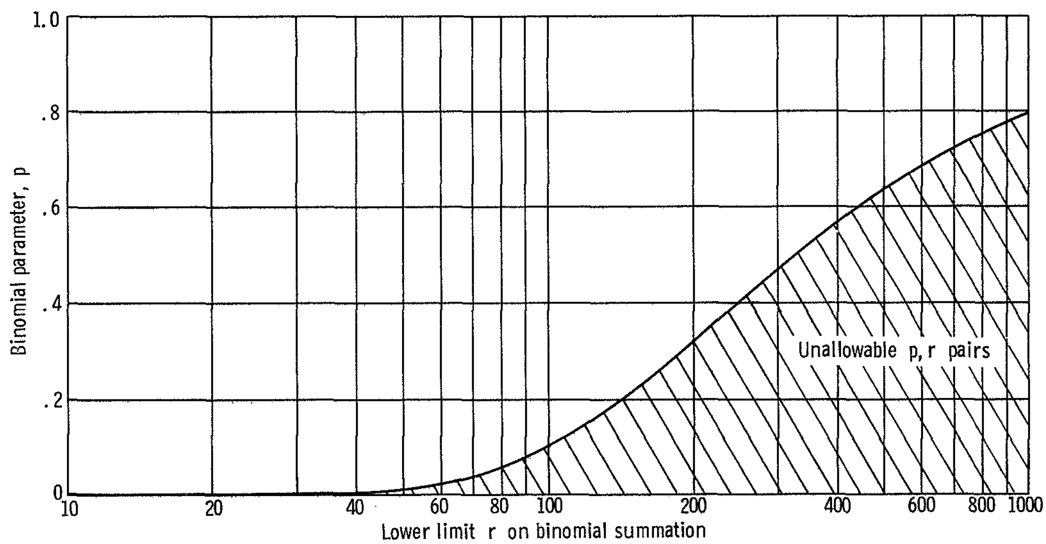

Figure 2. - Maximum r and minimum p values allowed in direct program. Allowable values determined from $p^r \geq 10^{-99}$.

Error Light

The error light comes on only when the input data are improper. This occurs when r is greater than n or when p is outside the allowable range of 0 to 1.

INDIRECT PROGRAM

The user instructions and steps of the indirect program are presented in figure 3. This program determines the value of p for given values of n, r, and C. This is done with the aid of the Newton-Raphson iterative procedure described in the reference. The

Enter program: (Starting address is 0 - 0)

Enter data: n → z, r → y, C → x

Press: Go to (0)(0) or End

Press: Continue

Pause display:

$(C - \hat{C}_i)$ —— z

$|p_{i+1} - p_i|$ —— y

p_i —— x

Final display:

$(C - \hat{C}_{f-1})$ —— z

$|p_f - p_{f-1}|$ —— y

p_f —— x

(a) User instructions.

STEP		KEY	CODE	DISPLAY x	y	z
0	0	+	33	C	r + C	n
	1	y→( )	40	Compute and store r + C		
	2	b	14			
	3	Roll ↓	31			
	4	Int x	64			
	5	-	34	Compute and store n - r		
	6	y→( )	40			
	7	c	16			
	8	Roll ↓	31			
	9	.	21			
	a	5	05			
	b	-	34	Test if C is outside the inter-		
	c	|y|	55	val 0 to 1 and, if so, force		
	d	If x > y	53	error light to come on		
1	0	1	01			
	1	4	04			
	2	0	00			
	3	÷	35			
	4	c	16	Enter n - r		
	5	Roll ↓	31			
	6	1	01	Compute r - 1		
	7	-	34			
	8	↓	25	Compute n - 1		
	9	+	33			
	a	x⇄y	30	Calculate p_0		
	b	÷	35			
	c	y→( )	40	Store p_0		
	d	d	17			

STEP		KEY	CODE	DISPLAY x	y	z
2 (From a-0 or a-5)	0	d	17			
	1	↑	27			
	2	1	01	Calculate $1 - p_i$		
	3	x⇄y	30			
	4	-	34			
	5	y→( )	40	Store $1 - p_i$		
	6	d	17			
	7	Clear	20			
	8	1	01	Enter 1's in f and e		
	9	↑	27			
	a	Acc +	60			
	b	↓	25			
(From 4-7)	c	1	01	Increment counter		
	d	+	33			
3	0	↑	27			
	1	b	14			
	2	Int x	64			
	3	x⇄y	30			
	4	-	34			
	5	↓	25	Compute successive term of		
	6	x⇄y	30	the binomial summation		
	7	+	33			
	8	÷	35			
	9	d	17			
	a	×	36			
	b	f	15			
	c	×	36			
	d	Clear x	37			

(b) Program steps.

Figure 3. - Indirect-program user instructions and program steps.

Step	Key	Code	Comment	
4-0	Acc +	60	Store sum and individual term	
4-1	y→( )	40		
4-2	f	15		
4-3	↓	25	Test if all terms have been computed	
4-4	c	16		
4-5	If x>y	53		
4-6	2	02		
4-7	c	16		To 2-c
4-8	↑	27	Store r	
4-9	b	14		
4-a	Int x	64		
4-b	x→( )	23		
4-c	f	15		
4-d	c	16	Compute n	From 6-8
5-0	↑	27		
5-1	b	14		
5-2	Int x	64		
5-3	+	33		
5-4	1	01	Compute product of $(1 - p_i)$ and successive term of the factorials	
5-5	+	33		
5-6	↓	25		
5-7	x⇄y	30		
5-8	-	34		
5-9	÷	35		
5-a	d	17		
5-b	×	36		
5-c	f	15		
5-d	×	36		
6-0	y→( )	40	Store product of $(1 - p_i)$ and the factorial terms	
6-1	f	15		
6-2	↓	25	Increment counter	
6-3	1	01		
6-4	-	34		
6-5	Clear x	37	Test counter to see if $r\binom{n}{r}(1 - p_i)^{n-r}$ is complete	
6-6	If x<y	52		
6-7	4	04		
6-8	d	17		To 4-d
6-9	1	01	Compute p_i	
6-a	↑	27		
6-b	d	17		
6-c	-	34		
6-d	y→( )	40	Store p_i	
7-0	d	17		
7-1	b	14		
7-2	Int x	64		
7-3	x⇄y	30		
7-4	ln x	65	Compute p_i^r	
7-5	×	36		
7-6	↓	25		
7-7	e^x	74		
7-8	y⇄( )	24	Compute binomial sum	
7-9	e	12		
7-a	×	36		
7-b	y→( )	40	Store $\hat{C}_i$	
7-c	e	12		
7-d	y⇄( )	24		

Step	Key	Code	Comment	
8-0	f	15	Form binomial derivative	
8-1	×	36		
8-2	d	17		
8-3	÷	35		
8-4	y→( )	40	Store $\frac{\partial \hat{C}_i}{\partial p_i}$	
8-5	f	15		
8-6	b	14	Compute C from r + C	
8-7	↑	27		
8-8	Int x	64		
8-9	-	34		
8-a	e	12	Compute $C - \hat{C}_i$	
8-b	-	34		
8-c	↑	27	Compute increment of p	
8-d	↓	25		
9-0	f	15		
9-1	÷	35		
9-2	d	17	Compute and store new approximation of p	
9-3	+	33		
9-4	y→( )	40		
9-5	d	17		
9-6	-	34	Pause to display p_i, $\lvert\Delta p\rvert$, and ΔC	
9-7	|y|	55		
9-8	Pause	57		
9-9	Enter exp	26	Form 10^{-6}	
9-a	Chg sign	32		
9-b	6	06		
9-c	If x<y	52	Test increment of p and Stop if $<10^{-6}$	
9-d	2	02		
a-0	0	00		To 2-0
a-1	d	17		
a-2	Stop	41	Display: x = p, y = $\lvert\Delta p\rvert$, z = ΔC	
a-3	Go to	44	Branch to 2-0 for another iteration	
a-4	2	02		
a-5	0	00		To 2-0
a-6	End	46		
a-7 to a-d	Unused		Blocks b to f are used for storage	

(b) Concluded.

Figure 3. - Concluded.

$(i + 1)^{th}$ estimate of p is determined from the i^{th} estimate of p as follows:

$$p_{i+1} = p_i + \frac{C - \hat{C}_i}{\frac{\partial \hat{C}_i}{\partial p_i}} \tag{5}$$

where

$$\hat{C}_i = p_i^r \sum_{j=0}^{n-r} \binom{r + j - 1}{j} \left(1 - p_i\right)^j \tag{6}$$

and

$$\frac{\partial \hat{C}_i}{\partial p_i} = r\binom{n}{r} p_i^{r-1} \left(1 - p_i\right)^{n-r} \tag{7}$$

In order for convergence to be assured, the initial guess for p is chosen at the inflection point on a curve of C as a function of p; that is,

$$p_0 = \frac{r - 1}{n - 1} \tag{8}$$

Equation (5) is then used to determine all subsequent approximations of p until the increment of p is less than 10^{-6}; that is,

$$\left|p_{i+1} - p_i\right| < 10^{-6} \tag{9}$$

The three inputs n, r, and C are loaded by the user into the z, y, and x registers, respectively. After loading, the user must press "Go to (0)(0)" and "Continue." The program contains a "Pause" in the iteration loop to display p_i, $|p_{i+1} - p_i|$, and $C - \hat{C}_i$ in the x, y, and z registers, respectively. Keeping the "Pause" button depressed results in a "Stop" at this display. The final display is p_f, $|p_f - p_{f-1}|$, $C - \hat{C}_{f-1}$ in the x, y, and z registers, respectively. Pressing "Continue" after the final display sends the calculator through another iteration for added accuracy in p.

The following sample cases illustrate the use of the indirect program. The times given are the approximate delays between pressing "Continue" and the final display of the results.

Case 4: $n = 10$ $\quad C - \hat{C}_{f-1} = 0.000\,000\,562$
$r = 7$ $\quad |p_f - p_{f-1}| = 0.000\,000\,525$
$C = 0.95$ $\quad p_f = 0.849\,972$
5 iterations, about 4 seconds

Case 5: $n = 100$ $\quad C - \hat{C}_{f-1} = 0.000\,000\,019$
$r = 75$ $\quad |p_f - p_{f-1}| = 0.000\,000\,004$
$C = 0.90$ $\quad p_f = 0.797\,094$
5 iterations, about 17 seconds

Case 6: $n = 1000$ $\quad C - \hat{C}_{f-1} = 0.000\,005\,791$
$r = 950$ $\quad |p_f - p_{f-1}| = 0.000\,000\,335$
$C = 0.95$ $\quad p_f = 0.959\,946$
5 iterations, about 32 seconds

Limitations

The convergence criterion is based on the increment of p (eq. (9)). Since comparable accuracy does not always exist in $\hat{C}$, the error $C - \hat{C}_{f-1}$ is displayed. More accuracy in p and, hence, in $\hat{C}$ can be obtained by pressing "Continue" and going through another iteration.

Error Lights

The following conditions result in an error light:

(1) An intermediate p_{i+1} outside the allowable range of 0 to 1. There are two possible causes of this condition: (a) $\hat{C}_i$ is incorrectly set to zero when p_i^r is less than 10^{-99} (see eq. (6)); this erroneous $\hat{C}_i$ is used in equation (5). (b) The derivative is incorrect when intermediate steps in computing equation (7) exceed the capacity of the calculator; this erroneous derivative is used in equation (5).

Figure 4 shows the allowable pairs of n and r that yield correct solutions for $C \geq 0.05$. Use of C values less than 0.05 further contracts the allowable region, whereas values of C greater than 0.05 slightly increase the size of the allowable region. For $n \leq 300$, r can take any value from 2 to $n - 1$. For $300 < n \leq 550$, there are few re-

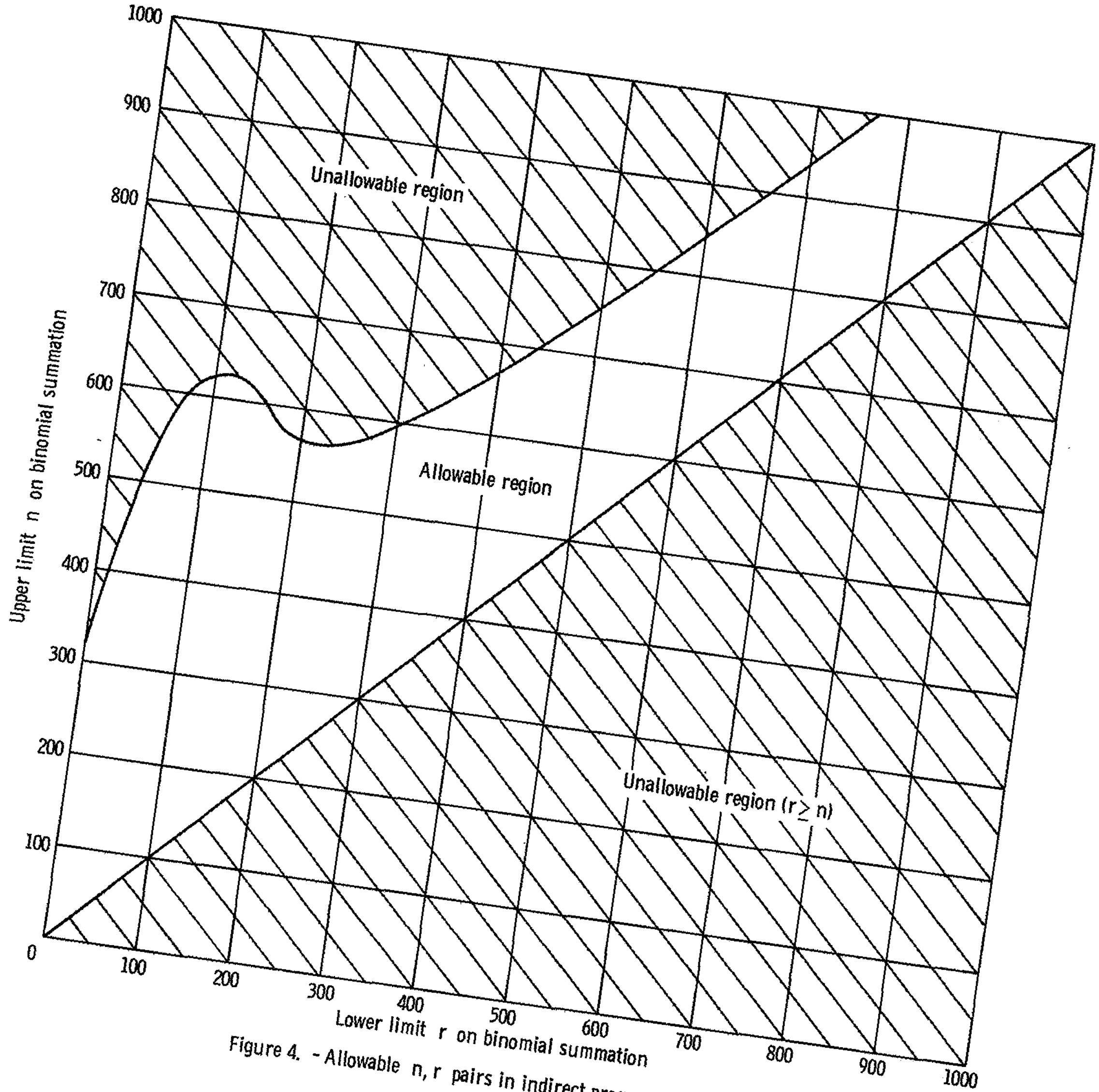

Figure 4. - Allowable n, r pairs in indirect program for $C \geq 0.05$.

strictions on r. For $550 < n \leq 1000$, there are limitations on r, as seen in figure 4. For $n > 1000$, the program has not been explored adequately for its limitations to be discussed.

(2) Unallowable values of r or C. Values of $r \leq 1$ and $r \geq n$ are not allowed; C values outside the range of 0 to 1 are not allowed. Although $r = 1$ and $r = n$ are legitimate cases for which to use the binomial function, this program cannot handle them because p_0 is set to 0 and 1, respectively (see eq. (8)). Since the logarithms of p_0 and $1 - p_0$ are used, the error light appears.

There is no need to use an iterative procedure to determine p in the cases $r = 1$ and $r = n$ since it can be determined directly. When $r = 1$,

$$C = 1 - (1 - p)^n \tag{10}$$

so

$$p = 1 - (1 - C)^{1/n} \tag{11}$$

When $r = n$,

$$C = p^n \tag{12}$$

so

$$p = C^{1/n} \tag{13}$$

APPLICATIONS

Several applications of the cumulative binomial function are discussed in the reference. In so-called r-out-of-n redundancy problems, the indirect program is useful in determining the required element reliability p. For binomial sampling plans, either program can be used to determine points on operating characteristic curves. The indirect program is useful in determining binomial confidence intervals, whereas the direct program is more conveniently used in hypothesis testing.

CONCLUDING REMARKS

The computer programs presented herein for the cumulative binomial function are useful for quickly determining the cumulative binomial sum C or, alternatively, the binomial parameter p. The probability of exactly r successes among n elements can be obtained from the direct program by subtracting the two appropriate cumulative sums; that is, the individual r^{th} binomial term is obtained by subtraction of the cumulative sum over the limits $r + 1$ to n from the cumulative sum over the limits r to n.

REFERENCE

1. Bien, Darl D.: Tables of Component Reliability for Binomial Redundancy Applications. NASA TN D-5549.