F77 Program 23

```
c     Fortran 77 program to do a simulation of the Weibull distribution
c     of the censored model. Realistic case.
c     Accelerated life model with 2 treatments. With subroutines.
c     This program may be adapted for other forms of censored models as
c     well as the simplest censoring model.
c     ***************************************************************
c           By Derek Dhammaloka FDX3 - 11th Mar. 1991
c     ***************************************************************
c           Define the following variables
c
c           cdf is the cumulative density function of the probability
c           function and is between 0 and 1. The function urand will
c           generate the random numbers between 0 and 1. It has 1
c           parameter iy, the seed to initialise the generator.
c           t is the remission time in arbitary units
c           knew is the new value of kappa to be entered by the user
c           rho is the rate to be entered by the user
c           loop is used in loop counters
c           n is the no. of individuals per treatment to be entered
c           by the user.
c           d is the no. of uncensored individuals in each
c           treatment group.
c           c is the censor time to be obtained using the equation
c           c=cs+((n-loop)/(n-1)) where cs is
c           the initial censor time to be entered by the user
c           i is the indicator variable (1 if censored, 0 otherwise)
c           x is equal to t if t is less than C, C otherwise
c           swops is the no. of swops
c           sorts is the no. of sorts
c           temp is used in sorting data values
c           list assigns the values 1,n so that it can be used in
c           sorting 1 variable in ascending or descending order and
c           the other variables have the same values as the original
c           data, but is being sorted
c           psn is the no. of trials in view at a certain time
c           ple is the product limit estimator
c           prcensor is the probability of censoring
c           a is the acceleration factor to be entered by the user
c           iy is the seed to be entered by the user
c     ***************************************************************
c           Find the product limit estimator
c     ***************************************************************
      integer i(2,5000),list(2,5000)
      real cdf(2,5000),t(2,5000),c(2,5000),x(2,5000)
      real knew,cs
      real rho,ple,prcensor,a
      integer loop,temp,swops,sorts,n,d(2),psn(2),iy(2),trt
      trt=2
c     ***************************************************************
c           Input the no. of individuals
c           Also the index (kappa) and the rate (rho)
c     ***************************************************************
      print*, 'How many individuals per treatment'
      read*,n
```

```
c
c        Set the no. of failures per treatment to the no. of
c        observations per treatment
c
      do 1 loop=1,trt
         d(loop)=n
1     continue
c
c        Set the no. of trials in view at time=0 to n+1 in
c        each treatment group
c
      do 2 loop=1,trt
         psn(loop)=n+1
2     continue
      do 3 loop=1,trt
         print*, 'Seed for the Weibull distribution in treatment ',trt
3     continue
      print*, 'Seed for the Weibull distribution'
      read*,iy
      print*, 'Enter the index parameter'
      read*,knew
      print*, 'Enter the rate parameter'
      read*,rho
      print*
      print*,'For the censor times'
      print*, 'Enter the initial censor time'
      print*
      read*,cs
      print*
      do 4 loop=1,trt
         if(loop.eq.1) then
          call simulate(knew,rho,n,cs,trt,d,iy,cdf,c,t,x,i,list,loop)
         else
          call simulate(knew,rho*a,n,cs,trt,d,iy,cdf,c,t,x,i,list,loop)
         endif
         call sort(n,trt,x,i,list,sorts,swops,temp,loop)
         call prlim(n,trt,x,i,list,psn,prcensor,ple,loop)
4     continue
      stop
      end

      subroutine simulate(knew,rho,n,cs,trt,d,iy,cdf,c,t,x,i,list,loop)
c
c        This subroutine simulates the Weibull distribution with
c        parameters rho and knew. It then decides on whether
c        each individual is to be censored using the censoring
c        rules.
c
c        tn is the treatment number
c
      integer n,loop,trt,tn
      integer d(trt),iy(trt),i(trt,n),list(trt,n)
      real cdf(trt,n),c(trt,n),t(trt,n),x(trt,n)
      real knew,rho,cs
c*************************************************************
c        Simulate the Weibull distribution
```

```
c             using the two parameters to obtain the remission times.
c             However, use an equation to obtain the censor times.
c     ************************************************************
      do 20 loop=1,n
             cdf(tn,loop)=urand(iy(tn))
             t(tn,loop)=(-log(1-cdf(tn,loop))/(rho**knew)**(1/knew)
             c(tn,loop)=(cs*(n-1))+(n-loop)
             c(tn,loop)=(c(tn,loop))/(n-1)
c
c            Decide whether the individual is to be censored
c            using the censoring rules
c
             if(t(tn,loop).ge.c(tn,loop)) then
                                     i(tn,loop)=1
                                     d(tn)=d(tn)-1
                                     x(tn,loop)=c(tn,loop)
             endif
             if(t(tn,loop).lt.c(tn,loop)) then
                                     i(tn,loop)=0
                                     x(tn,loop)=t(tn,loop)
             endif
             list(tn,loop)=loop
20    continue
c     ************************************************************
c            Print the headings
c     ************************************************************
      print*
      print*, Realistic case model I'
      print*
      print*,'Simulation of the Weibull distribution with '
      print*,'Index = ',knew,' and rate = ',rho
      print*
      print*, Initial censor time = ',cs
      print*
      write(*,25)
25    format(t3,'T',t10,'C',t40,'I',t55,'x')
c     ************************************************************
c            Output the remission and censor times
c            Also the indicator variable
c     ************************************************************
      do 30 loop=1,n
             write(*,40)t(tn,loop),c(tn,loop),i(tn,loop),x(tn,loop)
40           format(f7.3,t8,f7.3,t37,i7.0,t50,f7.3,t68)
30    continue
      print*
      return
      end

      subroutine sort(n,trt,x,i,list,sorts,swops,temp,tn)
c
c            This subroutine sorts the x values in ascending order
c
      integer n,trt,tn,sorts,swops,temp,loop
      integer i(trt,n),list(trt,n)
      real x(trt,n)
c
```

```
c
c             Set the no. of swops to 1 and the no. of sorts to 0
c
      swops=1
      sorts=0
c
c
c             Sort the x values in ascending order
c
50    if(swops.ne.0.and.sorts.lt.n-1) then
        swops=0
        sorts=sorts+1
        do 60 loop=1,n-sorts
              if(x(tn,list(tn,loop)).gt.x(tn,list(tn,loop+1))) then
                temp=list(tn,loop)
                list(tn,loop)=list(tn,loop+1)
                list(tn,loop+1)=temp
                swops=swops+1
                endif
60      continue
        goto 50
      endif
      return
      end

      subroutine prlim(n,trt,x,i,list,psn,prcensor,ple,tn)
c
c             This subroutine calculates the product limit
c             estimator of the data
c
      integer n,trt,tn,loop
      integer i(trt,n),list(trt,n),psn(trt)
      real x(trt,n),prcensor,ple
c
c             Set the product limit estimator to 1
c
      ple=1
c
c             Print headings
c
      print*
      print*, 'Calculation of the product limit estimator'
      print*, '(No. of failures assumed to be 1 for uncensored obs.)'
      print*
      write(*,63)
63    format(t3,'atom ',t14,'r ',t27,'(1-(1/r))',t42,'ple')
c
c             Calculate the product limit estimator using a do loop
c             Assume that the no. of failures for each x value is 1
c             providing that the individual is not censored since
c             the times are on a continuous scale and it is rare
c             for ties to occur.
c
      do 65 loop=1,n
              psn(tn)=psn(tn)-1
```

```
c
c           Only print the atom, position, censor probability and
c           the product limit if there is no censoring in the
c           individual.
c
            if(i(tn,list(tn,loop)).eq.0) then
                    prcensor=psn(tn)-1
                    prcensor=prcensor/psn(tn)
                    ple=ple*(prcensor)
                    write(*,75)x(tn,(list(tn,loop))),psn(tn),prcensor,ple
75                  format(f7.3,t8,i7.0,t25,f7.4,t40,f7.4,t55)
                    endif
65      continue
c
c           Print the final product limit estimator
c
            print*
            print*,'Product limit estimator =  ',ple
        return
        end


        real function urand(iy)
        integer iy
c       ********************************************************************
c           Urand is a uniform random number generator based on
c           theory and suggestions given by KNUTH (1969). The
c           integer iy should be initialised to an arbitary integer
c           prior to the first call to urand. The calling program
c           should not alter the value of iy between subsequent
c           calls to urand. Values of urand will be returned in the
c           interval (0,1).
c       ********************************************************************
c           Reference - Problem solving with Fortran 77
c                       Brian D.Hahn 1987
c       ********************************************************************
        integer ia,ic,itwo,m2,m,mic
        double precision halfm
        real s
        data m2/0/,itwo/2/

c       If first entry, compute machine integer word length
        if(m2.eq.0) then
                    m=1
10      if(m.gt.m2) then
                    m2=m
                    m=itwo*m2
                    goto 10
        endif
        halfm=m2

c       Compute multiplier and increment for linear congruential method
        ia=8*int(halfm*atan(1.d0)/8.d0)+5
        ic=2*int(halfm*(0.5d0-sqrt(3.d0)/6.d0))+1
        mic=(m2-ic)+m2
```

```
c       s is the scale factor for converting to floating point
        s=0.5/halfm
        endif

c       Compute next random number
        iy=iy*ia

c       The following statement is for computers which do not allow
c       integer overflow on addition
        if(iy.gt.mic) iy=(iy-m2)-m2
        iy=iy+ic

c       The following statement is for computers where the word length
c       is greater than for multiplication
        if(iy/2.gt.m2) iy=(iy-m2)-m2
c
c       The following statement is for computers where integer overflow
c       affects sign bit
        if(iy.lt.0) iy=(iy+m2)+m2
        urand=float(iy)*s
        return
        end
```

Output from F77 Program 23

```
How many individuals per treatment
175
Seed for the Weibull distribution for treatment                1
2
Seed for the Weibull distribution for treatment                2
-1
Enter the index parameter
3
Enter the rate parameter
1.25
Enter the acceleration factor
2

For the censor times
Enter the initial censor time
0.4

Realistic case model I

Simulation of the Weibull distribution with
Index =      .3000000E+01 and rate =      .1250000E+01

Initial censor time =      .4000000E+00
```

T	C	I	X
1.431	1.400	1	1.400
.536	1.394		.536
.589	1.389		.589
.803	1.383		.803
.639	1.377		.639
.485	1.371		.485
.444	1.366		.444
.839	1.360		.839
.761	1.354		.761
.538	1.348		.538
.624	1.343		.624
.703	1.337		.703
.333	1.331		.333
.678	1.325		.678
.769	1.320		.769
1.312	1.314		1.312
.745	1.308		.745
1.162	1.302		1.162
.798	1.297		.798
.812	1.291		.812
.347	1.285		.347
.655	1.279		.655
.807	1.274		.807
.662	1.268		.662
.981	1.262		.981
1.258	1.256	1	1.256
1.070	1.251		1.070
1.010	1.245		1.010
.623	1.239		.623

```
  .875   1.233                                                  .875
  .353   1.228                                                  .353
  .259   1.222                                                  .259
  .684   1.216                                                  .684
  .618   1.210                                                  .618
 1.030   1.205                                                 1.030
  .822   1.199                                                  .822
  .806   1.193                                                  .806
  .960   1.187                                                  .960
 1.007   1.182                                                 1.007
  .286   1.176                                                  .286
  .797   1.170                                                  .797
  .617   1.164                                                  .617
  .604   1.159                                                  .604
 1.407   1.153                                        1         1.153
  .730   1.147                                                  .730
  .519   1.141                                                  .519
  .396   1.136                                                  .396
  .764   1.130                                                  .764
  .457   1.124                                                  .457
  .443   1.118                                                  .443
 1.088   1.113                                                 1.088
  .961   1.107                                                  .961
  .594   1.101                                                  .594
  .902   1.098                                                  .902
  .702   1.090                                                  .702
  .394   1.084                                                  .394
  .631   1.078                                                  .631
  .664   1.072                                                  .664
  .274   1.067                                                  .274
  .732   1.061                                                  .732
 1.215   1.055                                        1         1.055
  .668   1.049                                                  .668
 1.064   1.044                                        1         1.044
  .977   1.038                                                  .977
  .512   1.032                                                  .512
 1.052   1.026                                        1         1.026
  .658   1.021                                                  .658
  .537   1.015                                                  .537
  .620   1.009                                                  .620
  .713   1.003                                                  .713
  .802    .998                                                  .802
 1.107    .992                                        1         .992
 1.027    .986                                        1         .986
  .749    .980                                                  .749
  .993    .975                                        1         .975
  .310    .969                                                  .310
  .541    .963                                                  .541
  .419    .957                                                  .419
  .940    .952                                                  .940
  .814    .946                                                  .814
  .846    .940                                                  .846
 1.305    .934                                        1         .934
  .578    .929                                                  .578
 1.186    .923                                        1         .923
  .561    .917                                                  .561
```

.384	.911		.384
.510	.906		.510
.614	.900		.614
.733	.894		.733
.619	.889		.619
.761	.883		.761
1.110	.877	1	.877
.723	.871		.723
.590	.866		.590
.822	.860		.822
.627	.854		.627
.690	.848		.690
.712	.843		.712
.366	.837		.366
.810	.831		.810
.601	.825		.601
.532	.820		.532
.634	.814		.634
.880	.808	1	.808
.985	.802	1	.802
.790	.797		.790
.558	.791		.558
.481	.785		.481
.877	.779	1	.779
.420	.774		.420
.504	.768		.504
.918	.762	1	.762
.869	.756	1	.756
.368	.751		.368
.521	.745		.521
.804	.739	1	.739
1.011	.733	1	.733
.474	.728		.474
.959	.722	1	.722
.846	.716	1	.716
.869	.710	1	.710
.300	.705		.300
.867	.699	1	.699
.663	.693		.663
.452	.687		.452
.793	.682	1	.682
.688	.676	1	.676
.996	.670	1	.670
.495	.664		.495
.763	.659	1	.659
.775	.653	1	.653
.892	.647	1	.647
.815	.641	1	.641
.733	.636	1	.636
.437	.630		.437
.353	.624		.353
1.031	.618	1	.618
.561	.613		.561
.812	.607	1	.607
.812	.601	1	.601
.879	.595	1	.595

.855	.590		1	.590
.258	.584			.258
.900	.578		1	.578
.850	.572		1	.572
.362	.567			.362
.865	.561		1	.561
.432	.555			.432
.589	.549		1	.549
.500	.544			.500
.592	.538		1	.538
.965	.532		1	.532
.505	.526			.505
.857	.521		1	.521
.536	.515		1	.515
.426	.509			.426
.624	.503		1	.503
.195	.498			.195
.734	.492		1	.492
.820	.486		1	.486
.574	.480		1	.480
.678	.475		1	.475
.536	.469		1	.469
.508	.463		1	.463
.878	.457		1	.457
.562	.452		1	.452
.205	.446			.205
.941	.440		1	.440
.475	.434		1	.434
.563	.429		1	.429
.725	.423		1	.423
.622	.417		1	.417
.657	.411		1	.411
.913	.406		1	.406
1.406	.400		1	.400

Calculation of the product limit estimator
(No. of failures assumed to be 1 for uncensored observations)

atom	r	(1-(1/r))	ple
.195	175	.9943	.9943
.205	174	.9943	.9886
.258	173	.9942	.9829
.259	172	.9942	.9771
.274	171	.9942	.9714
.286	170	.9941	.9657
.300	169	.9941	.9600
.310	168	.9940	.9543
.333	167	.9940	.9486
.347	166	.9940	.9429
.353	165	.9939	.9371
.353	164	.9939	.9314
.362	163	.9939	.9257
.366	162	.9938	.9200
.368	161	.9938	.9143

.384	160	.9937	.9086
.394	159	.9937	.9029
.396	158	.9937	.8971
.419	153	.9935	.8913
.420	152	.9934	.8854
.426	150	.9933	.8795
.432	148	.9932	.8736
.437	146	.9932	.8676
.443	144	.9931	.8616
.444	143	.9930	.8555
.452	141	.9929	.8495
.457	140	.9929	.8434
.474	136	.9926	.8372
.481	133	.9925	.8309
.485	132	.9924	.8246
.495	129	.9922	.8182
.500	128	.9922	.8118
.504	126	.9921	.8054
.505	125	.9920	.7989
.510	124	.9919	.7925
.512	123	.9919	.7861
.519	121	.9917	.7796
.521	119	.9916	.7730
.532	117	.9915	.7664
.536	116	.9914	.7598
.537	115	.9913	.7532
.538	113	.9912	.7465
.541	112	.9911	.7399
.558	110	.9909	.7331
.561	109	.9908	.7264
.561	107	.9907	.7196
.578	104	.9904	.7127
.589	103	.9903	.7058
.590	101	.9901	.6988
.594	100	.9900	.6918
.601	98	.9898	.6847
.604	96	.9896	.6776
.614	94	.9894	.6704
.617	93	.9892	.6632
.618	92	.9891	.6560
.619	90	.9889	.6487
.620	89	.9888	.6414
.623	88	.9886	.6341
.624	87	.9885	.6268
.627	86	.9884	.6195
.631	85	.9882	.6123
.634	84	.9881	.6050
.639	82	.9878	.5976
.655	78	.9872	.5899
.658	77	.9870	.5823
.662	75	.9867	.5745
.663	74	.9865	.5667
.664	73	.9863	.5590
.668	72	.9861	.5512
.678	69	.9855	.5432
.684	67	.9851	.5351

.690	66	.9848	.5270
.702	64	.9844	.5188
.703	63	.9841	.5105
.712	61	.9836	.5022
.713	60	.9833	.4938
.723	57	.9825	.4851
.730	56	.9821	.4765
.732	55	.9818	.4678
.733	54	.9815	.4591
.745	51	.9804	.4501
.749	50	.9800	.4411
.761	48	.9792	.4319
.761	47	.9787	.4228
.764	45	.9778	.4134
.769	44	.9773	.4040
.790	42	.9762	.3943
.797	41	.9756	.3847
.798	40	.9750	.3751
.802	39	.9744	.3655
.803	37	.9730	.3556
.806	36	.9722	.3457
.807	35	.9714	.3359
.810	33	.9697	.3257
.812	32	.9687	.3155
.814	31	.9677	.3053
.822	30	.9667	.2951
.822	29	.9655	.2850
.839	28	.9643	.2748
.846	27	.9630	.2646
.875	26	.9615	.2544
.902	24	.9583	.2438
.940	21	.9524	.2322
.960	20	.9500	.2206
.961	19	.9474	.2090
.977	17	.9412	.1967
.981	16	.9375	.1844
1.007	13	.9231	.1702
1.010	12	.9167	.1560
1.030	10	.9000	.1404
1.070	7	.8571	.1204
1.088	6	.8333	.1003
1.162	4	.7500	.0752
1.312	2	.5000	.0376

Product limit estimator = .3761759E-01

Realistic case model I

Simulation of the Weibull distribution with
Index = .3000000E+01 and rate = .2500000E+01

Initial censor time = .4000000E+00

T	C	I	X
.478	1.400		.478
.355	1.394		.355
.297	1.389		.297
.417	1.383		.417
.468	1.377		.468
.491	1.371		.491
.241	1.366		.241
.420	1.360		.420
.364	1.354		.364
.344	1.348		.344
.219	1.343		.219
.263	1.337		.263
.373	1.331		.373
.378	1.325		.378
.113	1.320		.113
.304	1.314		.304
.378	1.308		.378
.301	1.302		.301
.357	1.297		.357
.408	1.291		.408
.547	1.285		.547
.237	1.279		.237
.037	1.274		.037
.265	1.268		.265
.396	1.262		.396
.575	1.256		.575
.318	1.251		.318
.280	1.245		.280
.474	1.239		.474
.119	1.233		.119
.430	1.228		.430
.402	1.222		.402
.229	1.216		.229
.270	1.210		.270
.413	1.205		.413
.370	1.199		.370
.262	1.193		.262
.288	1.187		.288
.597	1.182		.597
.388	1.176		.388
.194	1.170		.194
.357	1.164		.357
.351	1.159		.351
.479	1.153		.479
.320	1.147		.320
.535	1.141		.535
.230	1.136		.230
.266	1.130		.266

.232	1.124	.232
.365	1.118	.365
.408	1.113	.408
.533	1.107	.533
.355	1.101	.355
.326	1.095	.326
.388	1.090	.388
.294	1.084	.294
.296	1.078	.296
.395	1.072	.395
.440	1.067	.440
.449	1.061	.449
.387	1.055	.387
.411	1.049	.411
.344	1.044	.344
.186	1.038	.186
.575	1.032	.575
.408	1.026	.408
.233	1.021	.233
.395	1.015	.395
.247	1.009	.247
.366	1.003	.366
.660	.998	.660
.432	.992	.432
.527	.986	.527
.411	.980	.411
.249	.975	.249
.369	.969	.369
.511	.963	.511
.258	.957	.258
.236	.952	.236
.544	.946	.544
.366	.940	.366
.438	.934	.438
.417	.929	.417
.276	.923	.276
.366	.917	.366
.511	.911	.511
.417	.906	.417
.184	.900	.184
.361	.894	.361
.285	.889	.285
.420	.883	.420
.362	.877	.362
.476	.871	.476
.340	.866	.340
.278	.860	.278
.434	.854	.434
.369	.848	.369
.492	.843	.492
.241	.837	.241
.377	.831	.377
.232	.825	.232
.255	.820	.255
.363	.814	.363
.250	.808	.250

.542	.802		.542
.513	.797		.513
.336	.791		.336
.285	.785		.285
.582	.779		.582
.325	.774		.325
.758	.768		.758
.199	.762		.199
.116	.756		.116
.326	.751		.326
.269	.745		.269
.395	.739		.395
.260	.733		.260
.292	.728		.292
.391	.722		.391
.407	.716		.407
.417	.710		.417
.446	.705		.446
.255	.699		.255
.216	.693		.216
.417	.687		.417
.481	.682		.481
.425	.676		.425
.428	.670		.428
.293	.664		.293
.390	.659		.390
.451	.653		.451
.322	.647		.322
.192	.641		.192
.246	.636		.246
.155	.630		.155
.209	.624		.209
.215	.618		.215
.426	.613		.426
.138	.607		.138
.293	.601		.293
.178	.595		.178
.323	.590		.323
.454	.584		.454
.340	.578		.340
.684	.572	1	.572
.409	.567		.409
.474	.561		.474
.197	.555		.197
.286	.549		.286
.396	.544		.396
.612	.538	1	.538
.584	.532	1	.532
.273	.526		.273
.282	.521		.282
.168	.515		.168
.362	.509		.362
.505	.503	1	.503
.272	.498		.272
.347	.492		.347
.179	.486		.179

.398	.480			.398
.284	.475			.284
.401	.469			.401
.354	.463			.354
.324	.457			.324
.310	.452			.310
.273	.446			.273
.297	.440			.297
.364	.434			.364
.270	.429			.270
.373	.423			.373
.234	.417			.234
.234	.411			.234
.271	.406			.271
.507	.400		1	.400

Calculation of the product limit estimator
(No. of failures assumed to be 1 for uncensored obs.)

atom	r	(1-(1/r))	ple
.037	175	.9943	.9943
.113	174	.9943	.9886
.116	173	.9942	.9829
.119	172	.9942	.9771
.138	171	.9942	.9714
.155	170	.9941	.9657
.168	169	.9941	.9600
.178	168	.9940	.9543
.179	167	.9940	.9486
.184	166	.9940	.9429
.186	165	.9939	.9371
.192	164	.9939	.9314
.194	163	.9939	.9257
.197	162	.9938	.9200
.199	161	.9938	.9143
.209	160	.9937	.9086
.215	159	.9937	.9029
.216	158	.9937	.8971
.219	157	.9936	.8914
.229	156	.9936	.8857
.230	155	.9935	.8800
.232	154	.9935	.8743
.232	153	.9935	.8686
.233	152	.9934	.8629
.234	151	.9934	.8571
.234	150	.9933	.8514
.236	149	.9933	.8457
.237	148	.9932	.8400
.241	147	.9932	.8343
.241	146	.9932	.8286
.246	145	.9931	.8229
.247	144	.9931	.8171
.249	143	.9930	.8114
.250	142	.9930	.8057
.255	141	.9929	.8000

.255	140	.9929	.7943
.258	139	.9928	.7886
.260	138	.9928	.7829
.262	137	.9927	.7771
.263	136	.9926	.7714
.265	135	.9926	.7657
.266	134	.9925	.7600
.269	133	.9925	.7543
.270	132	.9924	.7486
.270	131	.9924	.7429
.271	130	.9923	.7371
.272	129	.9922	.7314
.273	128	.9922	.7257
.273	127	.9921	.7200
.276	126	.9921	.7143
.278	125	.9920	.7086
.280	124	.9919	.7029
.282	123	.9919	.6971
.284	122	.9918	.6914
.285	121	.9917	.6857
.285	120	.9917	.6800
.286	119	.9916	.6743
.288	118	.9915	.6686
.292	117	.9915	.6629
.293	116	.9914	.6571
.293	115	.9913	.6514
.294	114	.9912	.6457
.296	113	.9912	.6400
.297	112	.9911	.6343
.297	111	.9910	.6286
.301	110	.9909	.6229
.304	109	.9908	.6171
.310	108	.9907	.6114
.318	107	.9907	.6057
.320	106	.9906	.6000
.322	105	.9905	.5943
.323	104	.9904	.5886
.324	103	.9903	.5829
.325	102	.9902	.5771
.326	101	.9901	.5714
.326	100	.9900	.5657
.336	99	.9899	.5600
.340	98	.9898	.5543
.340	97	.9897	.5486
.344	96	.9896	.5429
.344	95	.9895	.5371
.347	94	.9894	.5314
.351	93	.9892	.5257
.354	92	.9891	.5200
.355	91	.9890	.5143
.355	90	.9889	.5086
.357	89	.9888	.5029
.357	88	.9886	.4971
.361	87	.9885	.4914
.362	86	.9884	.4857
.362	85	.9882	.4800

.363	84	.9881	.4743
.364	83	.9880	.4686
.364	82	.9878	.4629
.365	81	.9877	.4571
.366	80	.9875	.4514
.366	79	.9873	.4457
.366	78	.9872	.4400
.369	77	.9870	.4343
.369	76	.9868	.4286
.370	75	.9867	.4229
.373	74	.9865	.4171
.373	73	.9863	.4114
.377	72	.9861	.4057
.378	71	.9859	.4000
.378	70	.9857	.3943
.387	69	.9855	.3886
.388	68	.9853	.3829
.388	67	.9851	.3771
.390	66	.9848	.3714
.391	65	.9846	.3657
.395	64	.9844	.3600
.395	63	.9841	.3543
.395	62	.9839	.3486
.396	61	.9836	.3429
.396	60	.9833	.3371
.398	59	.9831	.3314
.401	57	.9825	.3256
.402	56	.9821	.3198
.407	55	.9818	.3140
.408	54	.9815	.3082
.408	53	.9811	.3024
.408	52	.9808	.2965
.409	51	.9804	.2907
.411	50	.9800	.2849
.411	49	.9796	.2791
.413	48	.9792	.2733
.417	47	.9787	.2675
.417	46	.9783	.2617
.417	45	.9778	.2558
.417	44	.9773	.2500
.417	43	.9767	.2442
.420	42	.9762	.2384
.420	41	.9756	.2326
.425	40	.9750	.2268
.426	39	.9744	.2210
.428	38	.9737	.2151
.430	37	.9730	.2093
.432	36	.9722	.2035
.434	35	.9714	.1977
.438	34	.9706	.1919
.440	33	.9697	.1861
.446	32	.9687	.1803
.449	31	.9677	.1744
.451	30	.9667	.1686
.454	29	.9655	.1628
.468	28	.9643	.1570

.474	27	.9630	.1512
.474	26	.9615	.1454
.476	25	.9600	.1395
.478	24	.9583	.1337
.479	23	.9565	.1279
.481	22	.9545	.1221
.491	21	.9524	.1163
.492	20	.9500	.1105
.511	18	.9444	.1043
.511	17	.9412	.0982
.513	16	.9375	.0921
.527	15	.9333	.0859
.533	13	.9231	.0793
.535	12	.9167	.0727
.542	10	.9000	.0654
.544	9	.8889	.0582
.547	8	.8750	.0509
.575	6	.8333	.0424
.575	5	.8000	.0339
.582	4	.7500	.0254
.597	3	.6667	.0170
.660	2	.5000	.0085
.758	1	.0000	.0000

Product limit estimator = .0000000E+00