
HW4 EMA 471 Intermediate Problem Solving for Engineers

SPRING 2016
ENGINEERING MECHANICS DEPARTMENT
UNIVERSITY OF WISCONSIN, MADISON

INSTRUCTOR: PROFESSOR ROBERT J. WITT

BY

NASSER M. ABBASI

DECEMBER 30, 2019

Contents

0.1	Problem 1	3
0.1.1	Part (a)	3
0.1.2	Part (b)	4
0.2	Problem 2	8
0.2.1	High LET radiation	10
0.2.2	Low LET radiation	14
0.3	Problem 3	22

List of Tables

List of Figures

1	problem 1 description	3
2	Showing how sampling improves with increased N	5
3	The case for $N = 10^6$ only.	6
4	problem 2 description	8
5	High LET radiation, 10^4 trials, normal plot	10
6	High LET radiation, 10^4 trials log plot	11
7	High LET radiation, 10^5 trials, normal plot	11
8	High LET radiation, 10^5 trials log plot	12
9	High LET radiation, 10^6 trials, normal plot	12
10	High LET radiation, 10^6 trials log plot	13
11	High LET radiation, 10^7 trials, normal plot	13
12	High LET radiation, 10^7 trials log plot	14
13	Low LET radiation, 10^4 trials, normal plot	15
14	Low LET radiation, 10^4 trials log plot	15
15	Low LET radiation, 10^5 trials, normal plot	16
16	Low LET radiation, 10^5 trials log plot	16
17	Low LET radiation, 10^6 trials, normal plot	17
18	Low LET radiation, 10^6 trials log plot	17
19	Low LET radiation, 10^7 trials, normal plot	18
20	Low LET radiation, 10^7 trials log plot	18
21	problem 3 description	23
22	10^4 trials, normal plot	24
23	10^5 trials, normal plot	24
24	10^6 trials, normal plot	25
25	10^7 trials, normal plot	25
26	$\{10^4, 10^5, 10^6, 10^7, 10^8\}$ trials all on one plot.	27

0.1 Problem 1

Important note on Matlab: I used Matlab 2016a to write all the software. In particular, the function `histcounts()` was used for binning as recommended by Matlab help pages. This function do not exist in Matlab 2014a and was added in Matlab 2014b.

Note: For each of the problems below, you will be asked to run a large number of trials (10^6 or 10^7). During debugging, it may be useful to set this to a much lower number (maybe 10^3 or 10^4) until you are confident in your results. Once you are satisfied with your script, you can then increase the number of trials to the requested values to get a more statistically pleasing set of plots.

Note: Pre-allocating space for arrays is particularly important when you are running 10^n cases!

(1) (10 pts) An alternative representation of a peaked PDF is the Lorentzian shape:

$$f(x) = A \frac{1}{1+x^2},$$

Here x takes on values ranging from $\pm\infty$.

- Find the normalization constant A so that the probability of x taking on some value within its admissible range is 1.
- Use Matlab's `rand` utility to replicate this PDF by sampling its associated CDF and inverting it. Use 10^6 trials and plot your results in bins of width $\Delta x = 0.1$ over a range from $x = -10$ to $+10$.

Figure 1: problem 1 description

0.1.1 Part (a)

$$f(x) = A \frac{1}{1+x^2}$$

To normalize it, we first solve for A from

$$\int_{-\infty}^{\infty} f(x) dx = 1$$

Hence

$$\begin{aligned} 1 &= \int_{-\infty}^{\infty} A \frac{1}{1+x^2} dx \\ &= A [\arctan(x)]_{-\infty}^{\infty} \\ &= A \left(\frac{\pi}{2} + \frac{\pi}{2} \right) \end{aligned}$$

Therefore

$$A = \frac{1}{\pi}$$

Now we need to find cumulative probability distribution $F(x)$ and invert then it.

$$\begin{aligned} F(x) &= \int_{-\infty}^x f(x) dx \\ &= \int_{-\infty}^x \frac{1}{\pi} \frac{1}{1+x^2} dx \\ &= \frac{1}{\pi} (\arctan(x))_{-\infty}^x \\ &= \frac{1}{\pi} \left(\arctan(x) + \frac{\pi}{2} \right) \end{aligned}$$

Hence

$$F(x) = \frac{1}{\pi} \arctan(x) + \frac{1}{2}$$

To invert,

$$\pi \left(F(x) - \frac{1}{2} \right) = \arctan(x)$$

Hence

$$x = F^{-1}(x) = \tan \left(\pi \left(F(x) - \frac{1}{2} \right) \right)$$

0.1.2 Part(b)

Matlab rand was used to generate number of samples. To see the effect, the sampling was increased from 10^2 to 10^6 . The area under the pdf generated using sampling was normalized so that its area is one. The following plots show the result. The source code is included.

The algorithm used is the following

Algorithm 1 Algorithm to generate random variables from pdf using rand

```

1: procedure GENERATE_RV
2:    $F(x) \leftarrow$  cumulative distribution
3:    $N \leftarrow$  Number of trials
4:   for  $i \leftarrow 1, N$  do
5:      $x(i) \leftarrow$  rand()
6:      $data(i) \leftarrow F^{-1}(x(i))$ 
7:   end for
8:   generate normalized histogram from  $data$ 
9: end procedure

```

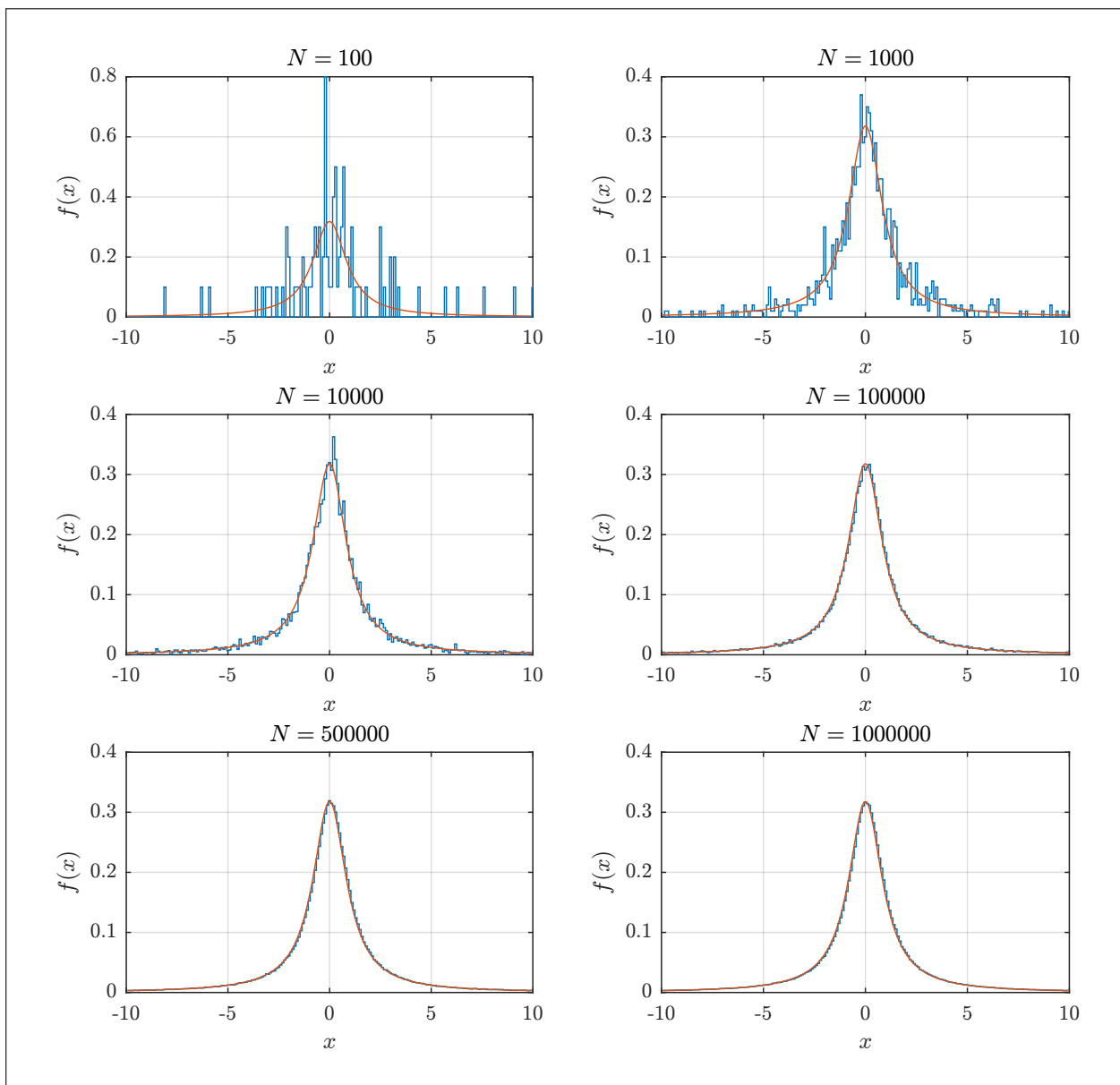


Figure 2: Showing how sampling improves with increased N

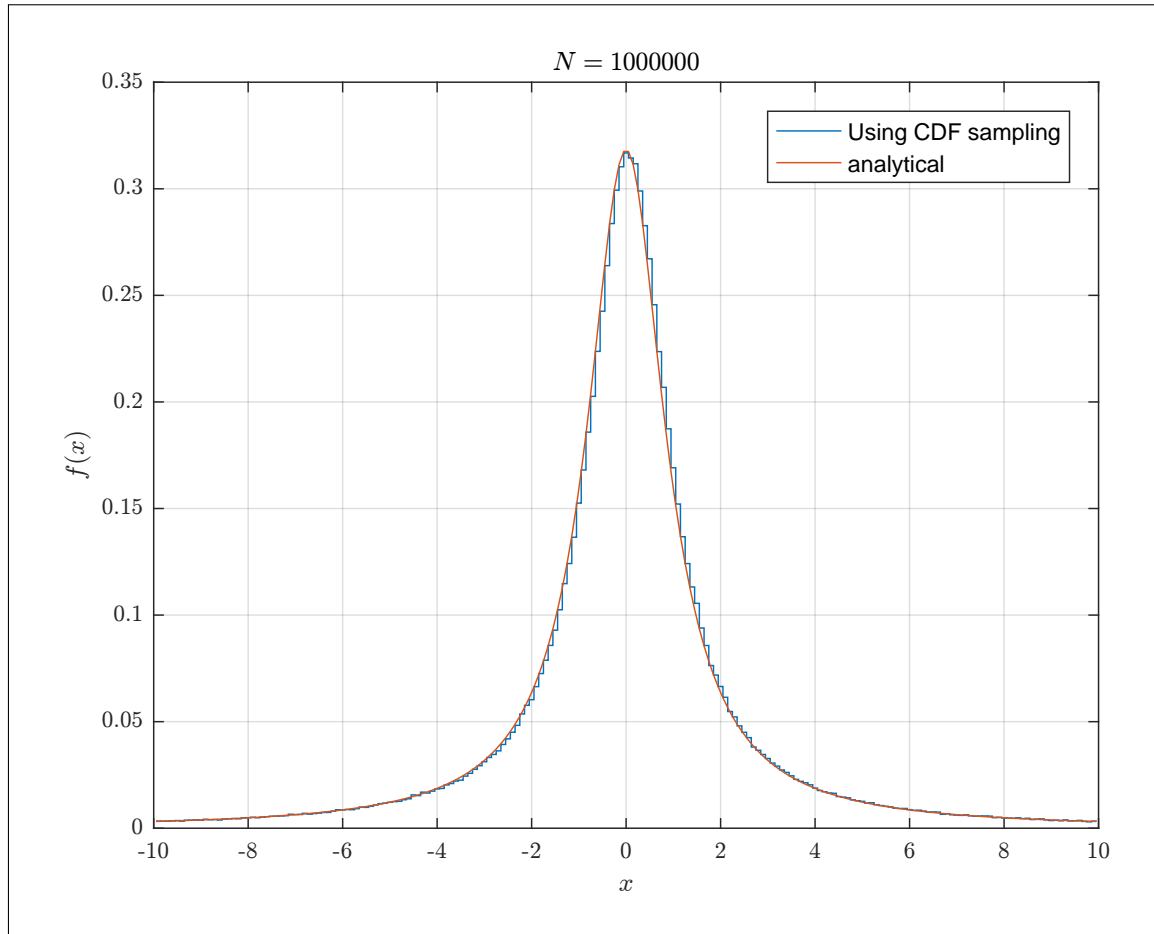


Figure 3: The case for $N = 10^6$ only.

```

1 function nma_HW4_problem_1()
2 %Problem 1, EMA 471, HW4
3 %by Nasser M. Abbasi
4 %
5 %
6 close all; clc;
7
8 N = [10^2 10^3 10^4 10^5 5*10^5 10^6];
9 for i=1:length(N) %make a plot for each N
10     subplot(3,2,i);
11     process(N(i));
12 end
13
14 %do last one on its own
15 figure();
16 process(N(end));
17 legend('Using CDF sampling','analytical');
18 end
19
20 %=====
21 %This function solve the problem for specific N. This is done
22 %to see how increasing N improves the process.

```

```

23 function process(N)
24 %Initialize RNG
25 rng('default');
26 rng(1);
27
28 %generate N random numbers from U(0,1)
29 xi = rand(N,1);
30
31 %sample the CDF using the inverse of F(x). Report
32 %shows the analytical result
33 data = finv(xi);
34
35 %generate the bins using histcounts
36 bin_width = 0.1;
37 edges = -10:bin_width:10;
38 x_bins = histcounts( data(data<=10&data>=-10),edges);
39
40 %now comes the tricky part. We need to find the real area
41 %in order to normalize with. If we use the area from the
42 %above, which is from -10<x<10, then this will not be
43 %accurate (it is a little smaller than the true area). To get
44 %the real data, we have to rebin the full data. This below is just
45 %to get the full area
46
47 edges = min(data):bin_width:max(data);
48 x_bins2 = histcounts( data,edges);
49 area = bin_width*sum(x_bins2);
50
51 %This below is another way to get the full area.
52 %h = histogram(data,'BinWidth',bin_width,...
53 %             'BinLimits',[min(data),max(data)]);
54 %area = bin_width*sum(h.Values); %use this to normalize with
55
56 %Continue with our bins -10<=x<=10. Now normalize
57 %it and Plot the pdf generated and compare it with the
58 %analytical one
59 x = -9.95:bin_width:9.95; %use center of bins
60 stairs(x,x_bins/area); %notice, we divide by the full area
61 hold on;
62
63 %add the analytical pdf. This is normalized
64 plot(x,(1/pi)*(1./(1+x.^2)));
65 title(sprintf('$N=%d$',N),'interpreter','Latex','FontSize',7);
66 xlabel('$x$', 'interpreter','Latex','FontSize',7);
67 ylabel('$f(x)$', 'interpreter','Latex','FontSize',7);
68 set(gca,'TickLabelInterpreter','Latex','fontSize',8);
69 grid;
70
71 end
72 %=====
73 %This is the inverse of the CDF. Derived by hand
74 %from inverting the analytical pdf given in the problem
75 function r= finv(y)
76 r = tan(pi*(y-1/2));

```

0.2 Problem 2

(2) (15 pts) The extent to which radiation causes cell damage depends strongly on its Linear Energy Transfer (LET) rate. Heavy charged particles have high LET values and cause far more damage than light charged particles with low LET values. The probability of cell survival can be described by the following two pdfs:

High LET radiation:

$$f_H(D) = \exp(-0.0187D)$$

Low LET radiation:

$$f_L(D) = \begin{cases} \exp(a_1D + a_2D^2 + a_3D^3 + a_4D^4), & D \leq 600 \text{ rads} \\ A \exp(a_5D), & D > 600 \text{ rads} \end{cases}$$

In the latter expression for $f_L(D)$, the coefficients have these values:

$$a_1 = -6.84806 \times 10^{-4}$$

$$a_2 = +4.87285 \times 10^{-6}$$

$$a_3 = -3.28988 \times 10^{-8}$$

$$a_4 = +2.66992 \times 10^{-11}$$

$$a_5 = -0.0073$$

$$A = 7.9839$$

Use a rejection algorithm to reproduce the probabilities of cell survival when cells are subjected to radiation doses up to 1000 rads. Use 1 rad bin widths and run your simulation for 10^7 trials for both the high and low LET cases. Plot your results on a semilog plot (`semilogy`) with the axes in x (D) from 0 to 1000 rads and in y (f) from 10^{-3} to 1. Note that the PDFs have already been normalized to have a maximum value of 1.

Figure 4: problem 2 description

The rejection method was used to first generate $f_H(D)$ and then was used again to generate $F_L(D)$. So the result shows each of these separately below.

For each case, number of trials was increased to see the effect. The following trials were used $\{10^4, 10^5, 10^6, 10^7\}$. The last amount is the one asked for in this problem. For each case, both the similogy plot is shown and the normal scale plot is also shown. These results shows the rejection method improves as more trials are used. The analytic pdf is also plotted against the generated one to compare.

Matlab source code is included. Implemented on Matlab 2016a.

The algorithm used is the following

Algorithm 2 Algorithm to generate random variables from pdf using rejection method

```
1: procedure GENERATE_RV
2:    $f(x) \leftarrow$  probability distribution
3:    $N \leftarrow$  Number of trials
4:    $k \leftarrow 0$ 
5:   for  $i \leftarrow 1, N$  do
6:      $\zeta_1 \leftarrow \text{rand}()$ 
7:      $\zeta_2 \leftarrow \text{rand}()$ 
8:     if  $\zeta_2 \leq f(\zeta_1)$  then
9:        $k \leftarrow k + 1$ 
10:      count[k]  $\leftarrow \zeta_1$ 
11:    end if
12:  end for
13:  generate normalized histogram from count
14: end procedure
```

0.2.1 High LET radiation

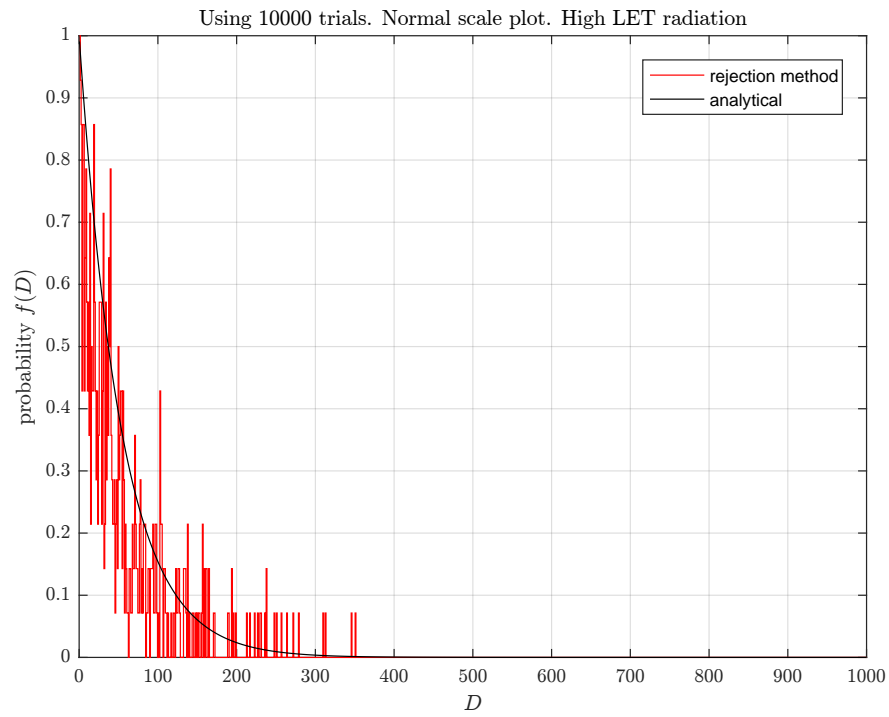
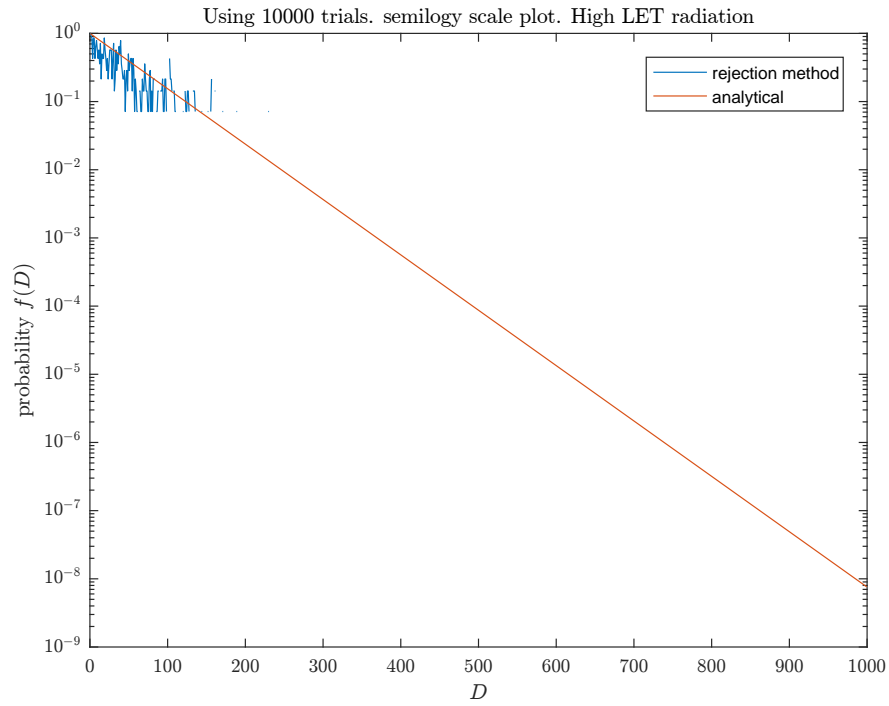
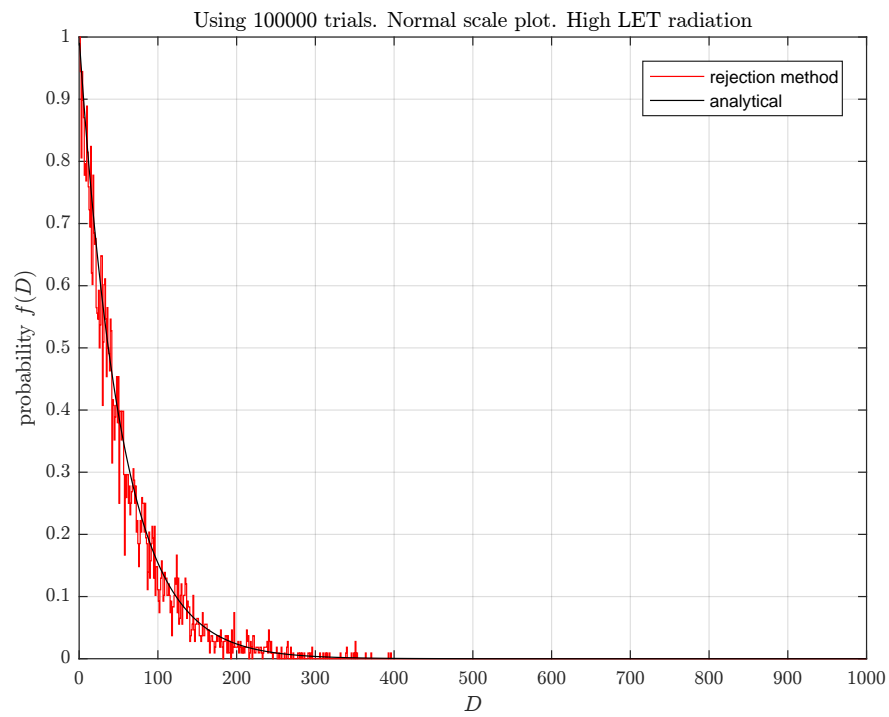


Figure 5: High LET radiation, 10^4 trials, normal plot

Figure 6: High LET radiation, 10^4 trials log plotFigure 7: High LET radiation, 10^5 trials, normal plot

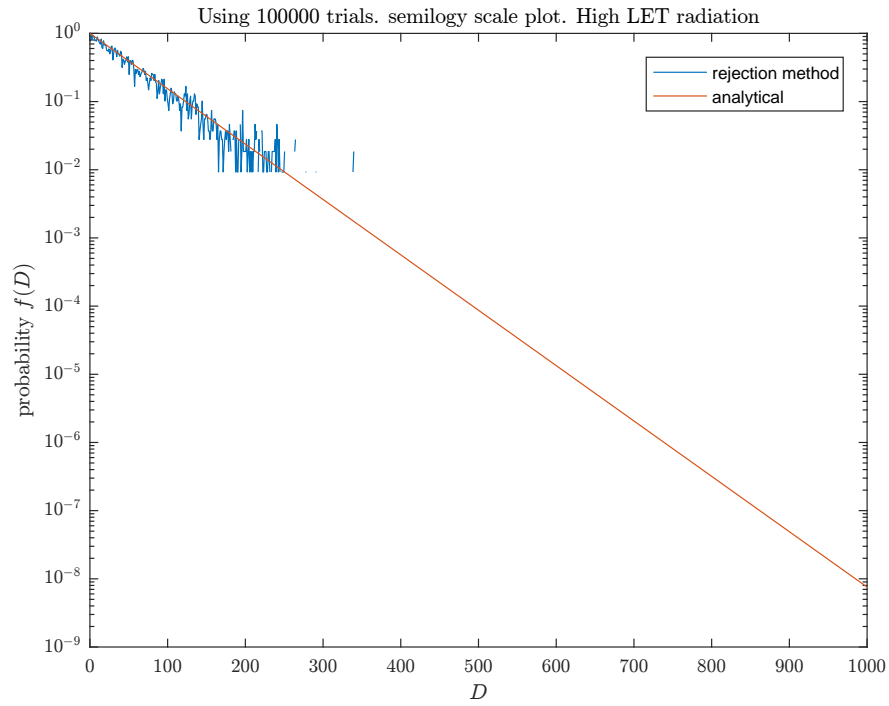


Figure 8: High LET radiation, 10^5 trials log plot

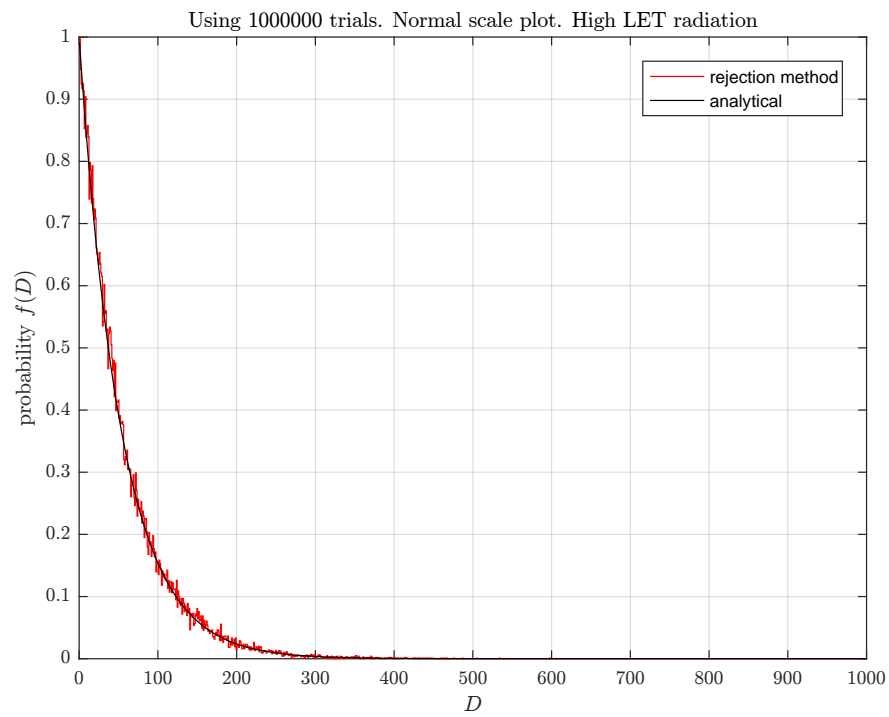


Figure 9: High LET radiation, 10^6 trials, normal plot

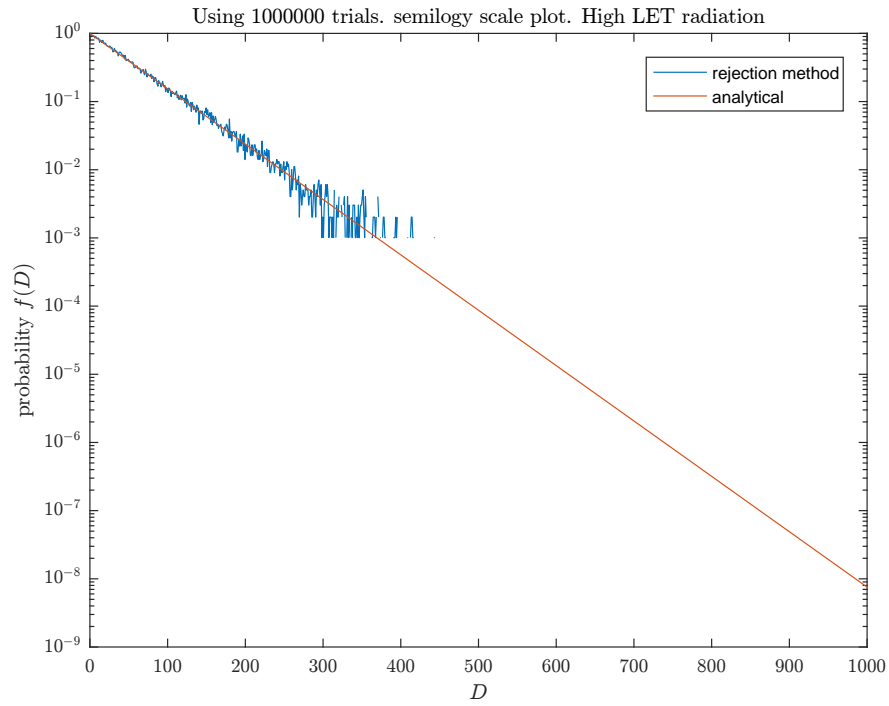


Figure 10: High LET radiation, 10^6 trials log plot

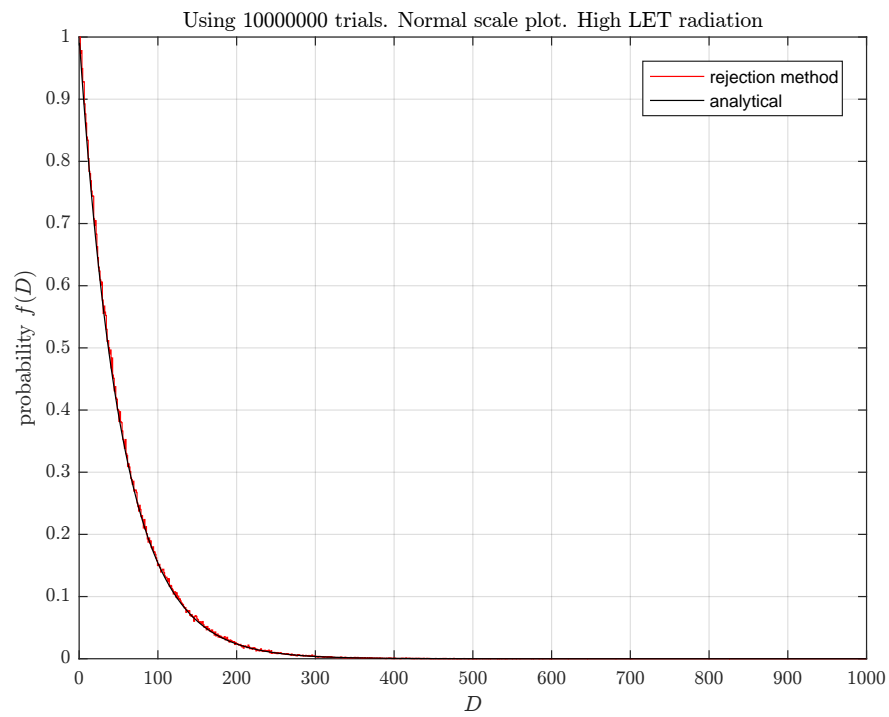


Figure 11: High LET radiation, 10^7 trials, normal plot

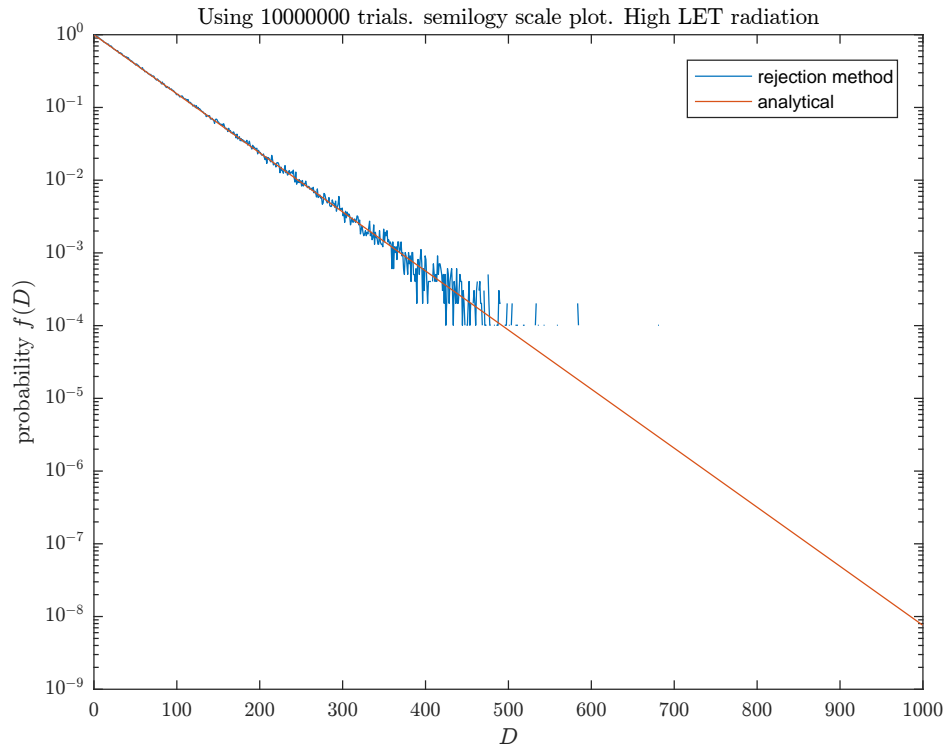


Figure 12: High LET radiation, 10^7 trials log plot

0.2.2 Low LET radiation

The above was repeated for the low LET radiation. The result is given below

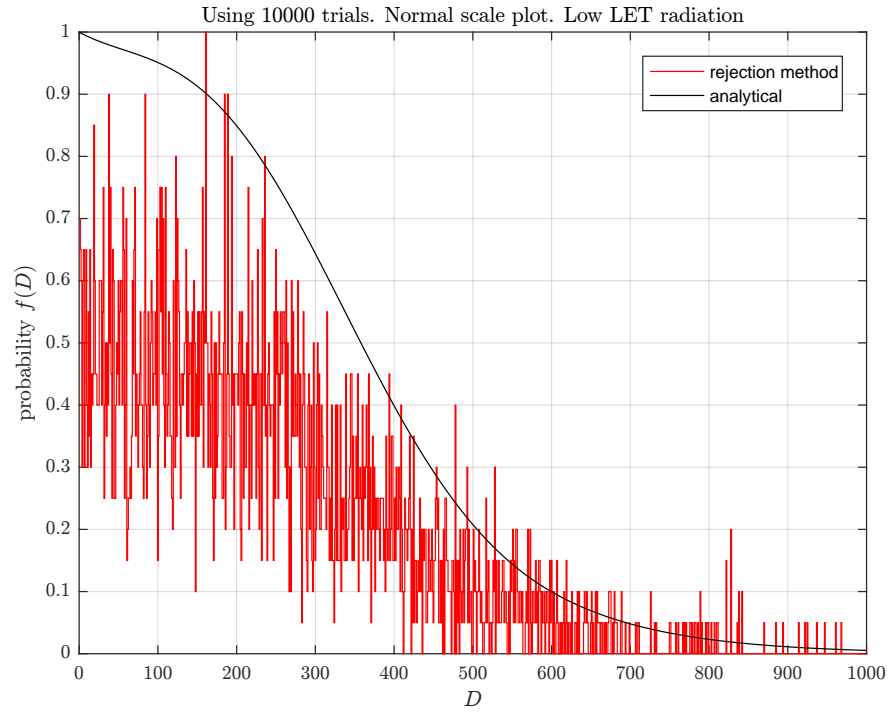


Figure 13: Low LET radiation, 10^4 trials, normal plot

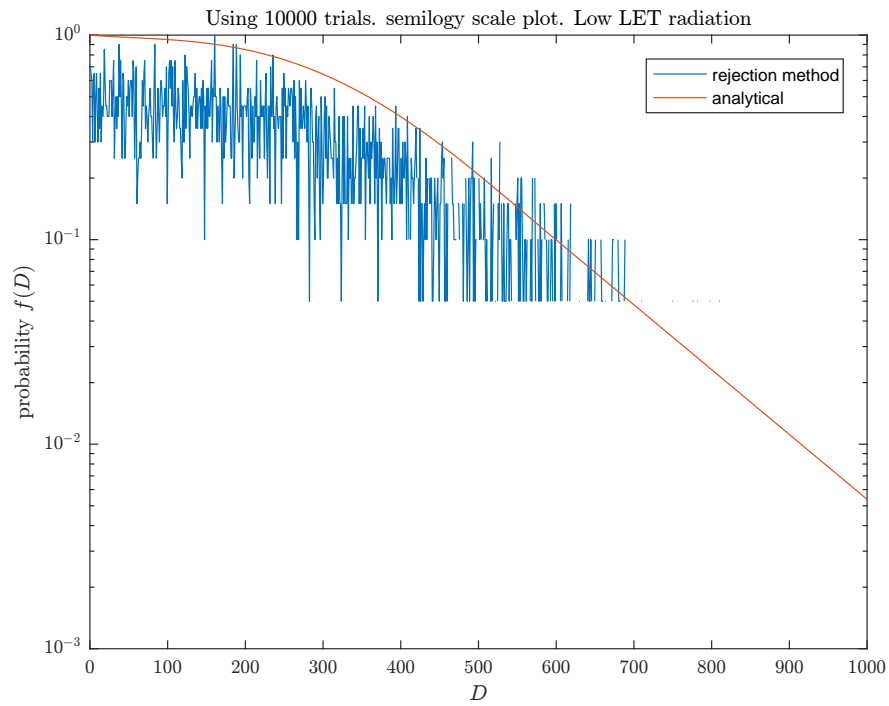


Figure 14: Low LET radiation, 10^4 trials log plot

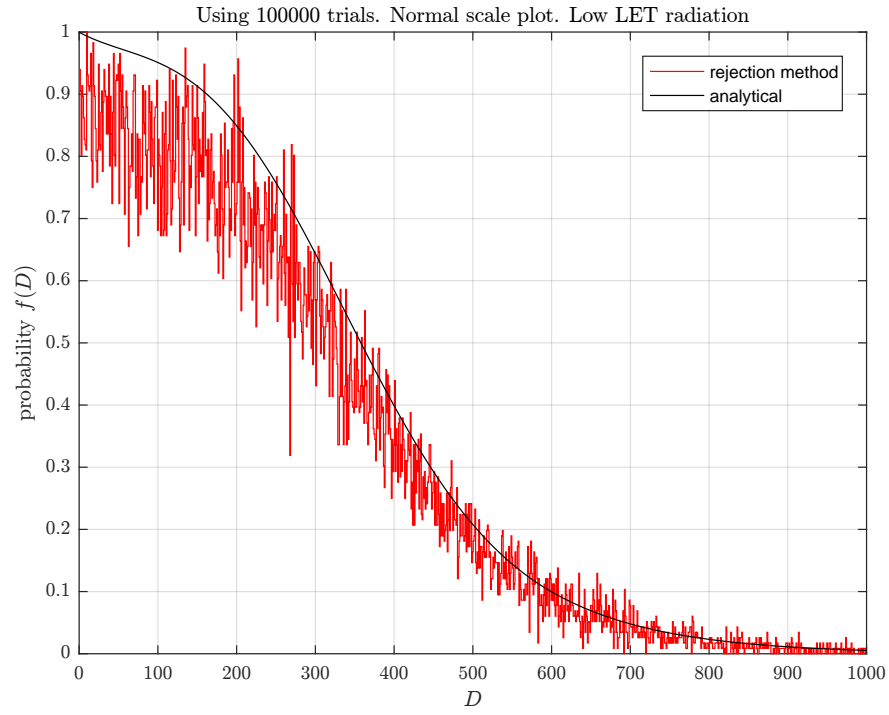


Figure 15: Low LET radiation, 10^5 trials, normal plot

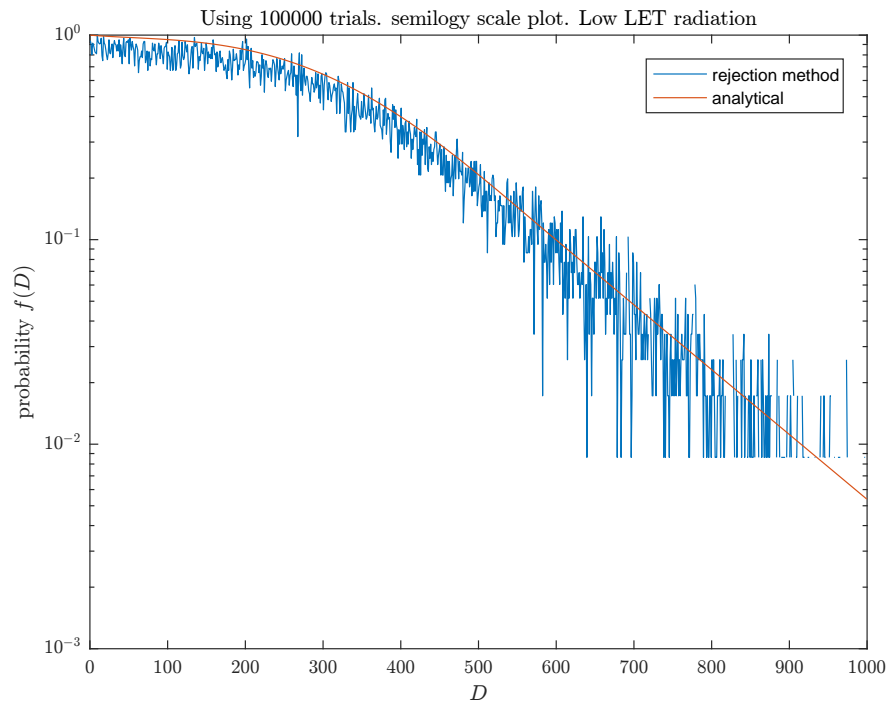
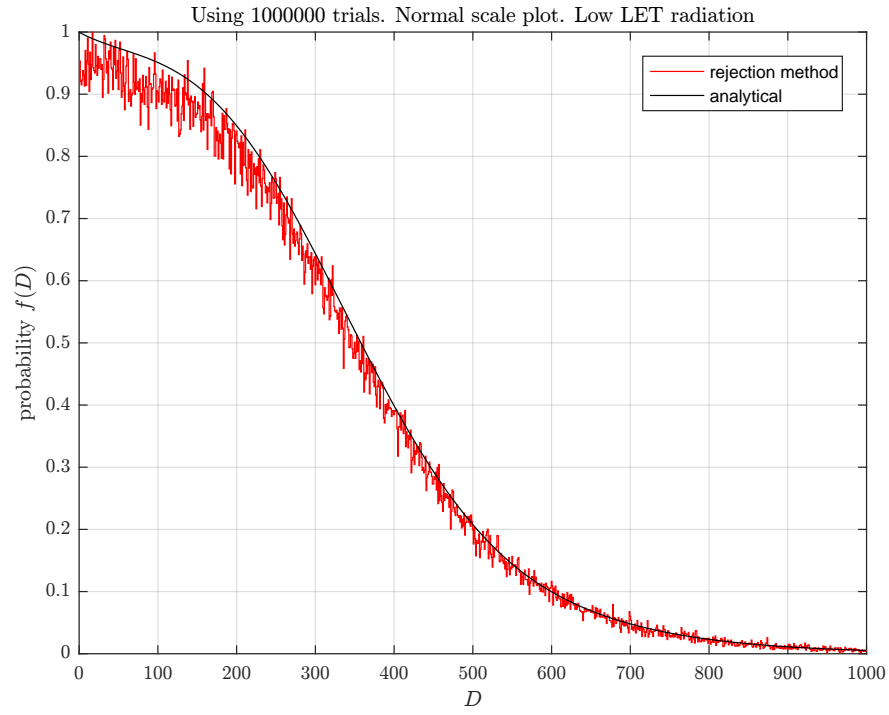
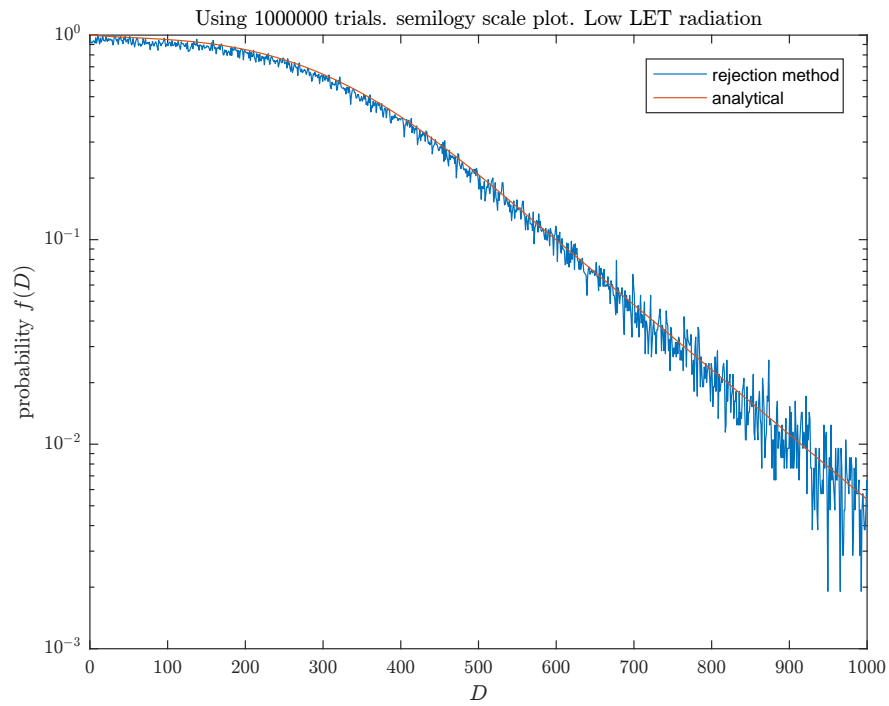
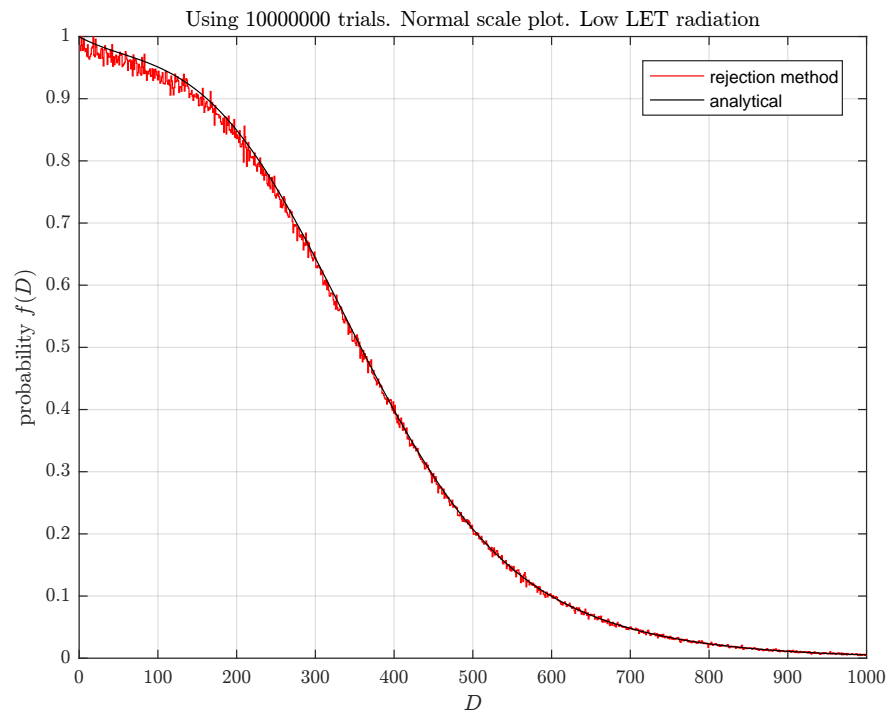
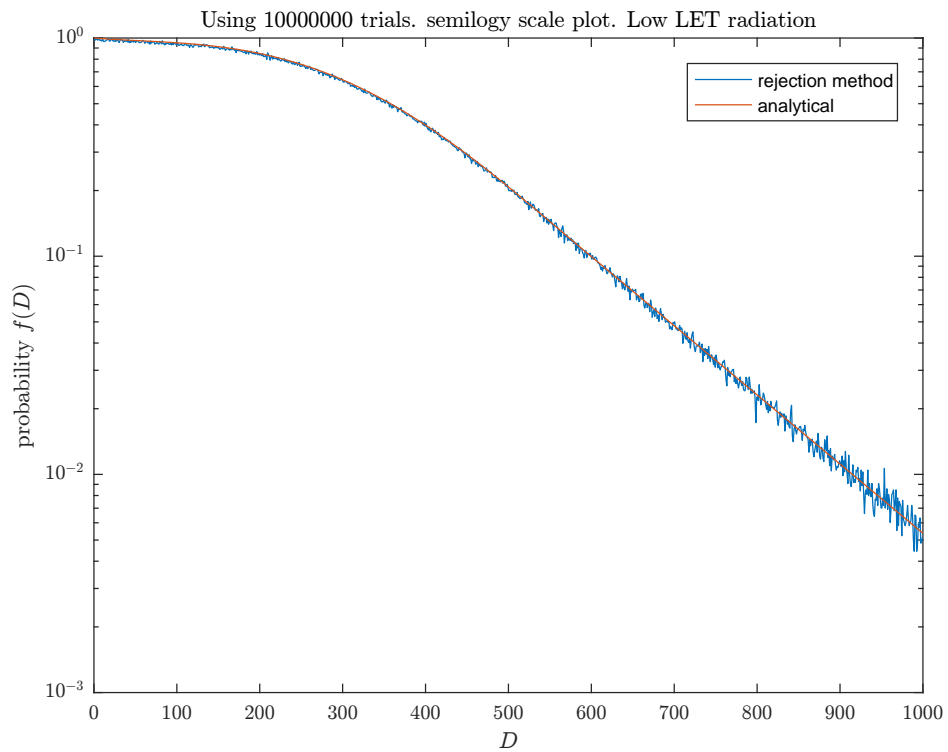


Figure 16: Low LET radiation, 10^5 trials log plot

Figure 17: Low LET radiation, 10^6 trials, normal plotFigure 18: Low LET radiation, 10^6 trials log plot

Figure 19: Low LET radiation, 10^7 trials, normal plotFigure 20: Low LET radiation, 10^7 trials log plot

```

1 function nma_HW4_problem_2()
2 %Problem 2, EMA 471, HW4
3 %by Nasser M. Abbasi
4 %
5 %
6 close all; clc;
7
8 ntrials = [10^4 10^5 10^6 10^7];
9 for i=1:length(ntrials)
10     process(ntrials(i));
11 end
12 end
13 %=====
14 function process(N)
15 %N: number of trials
16
17 %Initialize random number generator
18 rng('default');
19 rng(1);
20
21 MAX      = 1000;          %maximum rad to simulate
22 counts   = zeros(N,1);  %to save accepted trials into
23 k        = 0;           %counter for keeping accepted
24
25 for i=1:N
26     x_1 = rand();
27     x_2 = rand();
28     D = x_1*MAX; %convert to actual D from 0..1000
29     if x_2 < exp(-0.0187*D)
30         k = k + 1;
31         counts(k) = D;
32     end
33 end
34
35 generate_plots(N,counts(1:k),MAX);
36 end
37 %=====
38 function generate_plots(number_of_trials,counts,MAX)
39 generate_normal_plots(number_of_trials,counts,MAX);
40 generate_log_plots(number_of_trials,counts,MAX);
41 end
42 %=====
43 function generate_normal_plots(number_of_trials,counts,MAX)
44 figure;
45 bin_width = 1;
46 edges     = 0:bin_width:MAX;
47 x_bins    = histcounts( counts,edges);
48 x         = 0.5:bin_width:MAX-0.5; %use center of bins
49 stairs(x,x_bins/max(x_bins),'r'); %notice, we divide by the full area
50 hold on;
51 plot(x,exp(-0.0187*x),'k');
52 grid;
53

```

```

54 title(sprintf( ...
55     'Using %d trials. Normal scale plot. High LET radiation',...
56     number_of_trials),'interpreter','Latex','FontSize',7);
57 xlabel('$D$', 'interpreter','Latex','FontSize',7);
58 ylabel('probability $f(D)$', 'interpreter','Latex','FontSize',7);
59 set(gca,'TickLabelInterpreter', 'Latex','fontsize',8);
60 legend('rejection method','analytical');
61 print(gcf, '-dpdf', '-r600', ...
62     sprintf('../images/%d_part_a_normal_plot.pdf',number_of_trials));
63
64 end
65 %=====
66 function generate_log_plots(number_of_trials,counts,MAX)
67 figure;
68 bin_width = 1;
69 edges     = 0:bin_width:MAX;
70 x_bins    = histcounts( counts,edges);
71 x         = 0.5:bin_width:MAX-0.5; %use center of bins
72 semilogy(x,x_bins/max(x_bins));
73 hold on;
74 semilogy(x,exp(-0.0187*x));
75 %grid;
76
77 title(sprintf(...
78     'Using %d trials. semilogy scale plot. High LET radiation',...
79     number_of_trials),'interpreter','Latex','FontSize',7);
80 xlabel('$D$', 'interpreter','Latex','FontSize',7);
81 ylabel('probability $f(D)$', 'interpreter','Latex','FontSize',7);
82 set(gca,'TickLabelInterpreter', 'Latex','fontsize',8);
83 legend('rejection method','analytical');
84 print(gcf, '-dpdf', '-r600', ...
85     sprintf('../images/%d_part_a_log_plot.pdf',number_of_trials));
86
87 end

1 function nma_HW4_problem_2_second_part()
2 %Problem 2, Low Let radiation. EMA 471, HW4
3 %by Nasser M. Abbasi
4 %
5 %
6 close all; clc;
7
8 ntrials = [10^4 10^5 10^6 10^7];
9 for i=1:length(ntrials)
10     process(ntrials(i));
11 end
12 end
13 %=====
14 function process(N)
15 %N: number of trials
16
17 %Initialize random number generator
18 rng('default');
19 rng(1);

```

```

20
21 MAX      = 1000;          %maximum rad to simulate
22 counts   = zeros(N,1);  %to save accepted trials into
23 k        = 0;           %counter for keeping accepted
24
25 for i=1:N
26     x_1 = rand();
27     x_2 = rand();
28     D = x_1*MAX; %convert to actual D from 0..1000
29
30     if x_2 < pdf_low_let(D)
31         k = k + 1;
32         counts(k) = D;
33     end
34 end
35
36 generate_plots(N,counts(1:k),MAX);
37 end
38 %=====
39 function generate_plots(number_of_trials,counts,MAX)
40 analytical_pdf = zeros(MAX,1);
41 for i=1:MAX
42     analytical_pdf(i) = pdf_low_let(i);
43 end
44
45 generate_normal_plots(number_of_trials,counts,MAX,analytical_pdf);
46 generate_log_plots(number_of_trials,counts,MAX,analytical_pdf);
47 end
48 %=====
49 function generate_normal_plots(number_of_trials,counts, ...
50                               MAX,analytical_pdf)
51 figure;
52 bin_width = 1;
53 edges     = 0:bin_width:MAX;
54 x_bins    = histcounts( counts,edges);
55 x         = 0.5:bin_width:MAX-0.5; %use center of bins
56 %notice, we divide by the full area
57 stairs(x,x_bins/max(x_bins),'r');
58 hold on;
59 plot(x,analytical_pdf,'k');
60 grid;
61
62 title(sprintf(...
63         'Using %d trials. Normal scale plot. Low LET radiation',...
64         number_of_trials),'interpreter','Latex','FontSize',7);
65 xlabel('$D$', 'interpreter','Latex','FontSize',7);
66 ylabel('probability $f(D)$', 'interpreter','Latex','FontSize',7);
67 set(gca, 'TickLabelInterpreter', 'Latex', 'fontSize', 8);
68 legend('rejection method','analytical');
69 print(gcf, '-dpdf', '-r600', ...
70       sprintf('../images/%d_part_b_normal_plot.pdf',number_of_trials));
71
72 end
73 %=====

```

```

74 function generate_log_plots(number_of_trials,counts,...
75                             MAX,analytical_pdf)
76 figure;
77 bin_width = 1;
78 edges     = 0:bin_width:MAX;
79 x_bins    = histcounts( counts,edges);
80 x         = 0.5:bin_width:MAX-0.5; %use center of bins
81 semilogy(x,x_bins/max(x_bins));
82 hold on;
83 semilogy(x,analytical_pdf);
84 %grid;
85
86 title(sprintf(...
87         'Using %d trials. semilogy scale plot. Low LET radiation',...
88         number_of_trials),'interpreter','Latex','FontSize',7);
89 xlabel('$D$','interpreter','Latex','FontSize',7);
90 ylabel('probability $f(D)$','interpreter','Latex','FontSize',7);
91 set(gca,'TickLabelInterpreter','Latex','fontSize',8);
92 legend('rejection method','analytical');
93 print(gcf, '-dpdf', '-r600', ...
94        sprintf('../images/%d_part_b_log_plot.pdf',number_of_trials));
95
96 end
97 %=====
98 function r= pdf_low_let(D)
99 a1=-6.84806*10^(-4);
100 a2= 4.87285*10^(-6);
101 a3=-3.28988*10^(-8);
102 a4= 2.66992*10^(-11);
103 a5=-0.0073;
104 A=7.9839;
105
106 if D<=600
107     r = exp(a1*D+a2*D^2+a3*D^3+a4*D^4);
108 else
109     r = A*exp(a5*D);
110 end
111 end

```

0.3 Problem 3

(3) (15 pts) When populations are subjected to relatively high doses of radiation, there is an increased probability of members of the population developing certain kinds of cancer at times following exposure. The incidence of excess cancer has certain characteristics: there is a “latency period” where no excess incidence beyond the background rate is observed, followed by a relatively rapid rise to a “plateau period” where the cancer rate is definitely observed to be higher than (in excess of) the background rate. In the simplest representation of this excess incidence, the plateau period looks like a square wave, but in reality the transition between the latency period and the plateau period is smooth. In order to model this, consider the following normalized PDF:

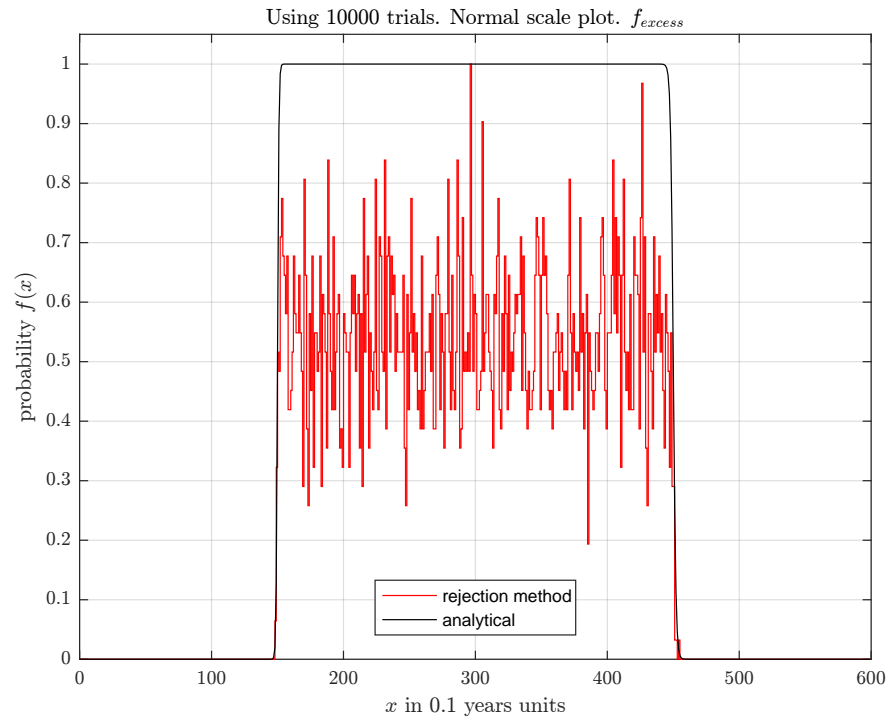
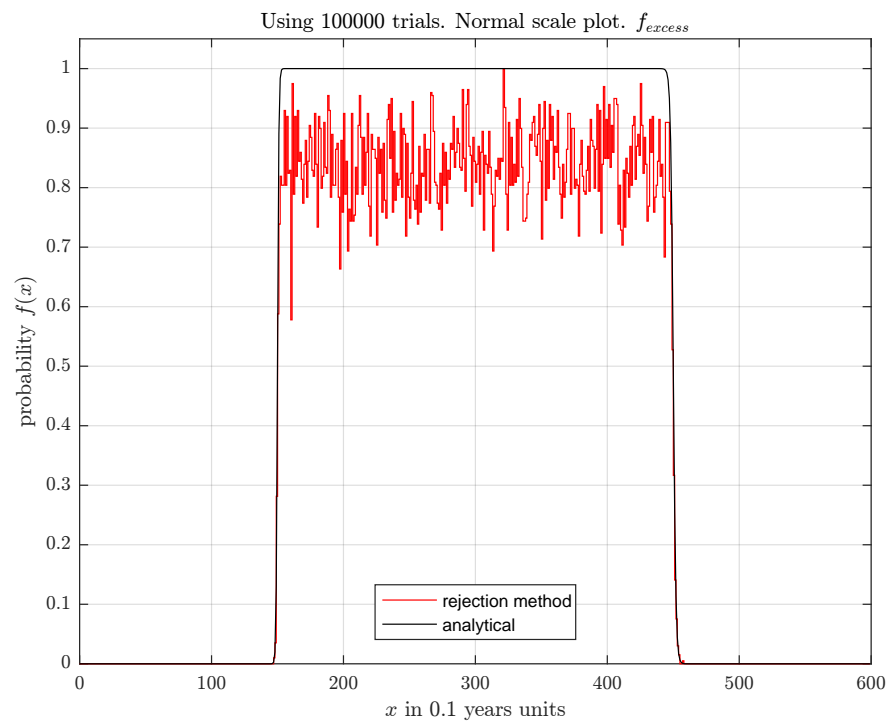
$$f_{\text{excess}} = \frac{1}{2} [\tanh(x - l) - \tanh[0.5(x - (l + p))]]$$

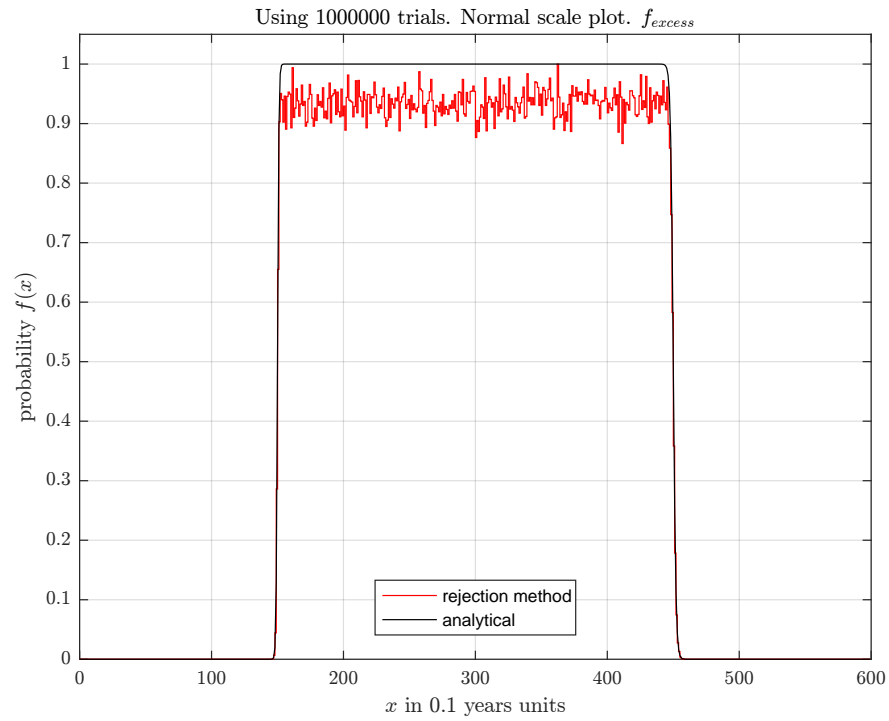
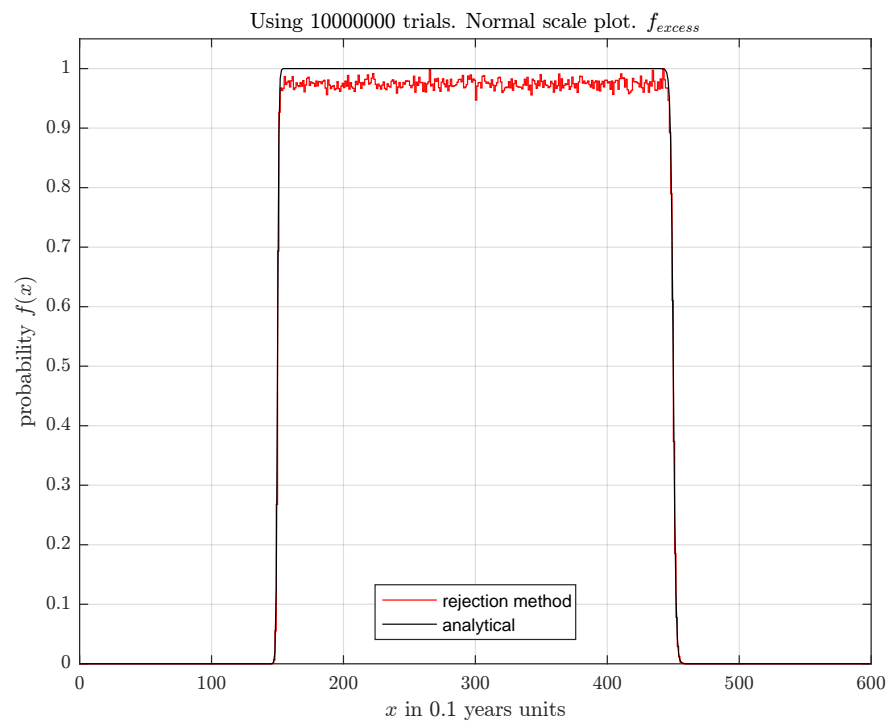
Here x is the number of years following exposure, l is the latency period (years) and p is the plateau period (years). For pancreatic cancer, the latency and plateau periods are 15 and 30 years respectively. Replicate this normalized PDF with the rejection algorithm, sorting incidences of excess cancer into bins of width 0.1 years for 60 years following exposure. Run 10^7 trials and plot your results.

Note: As an aside, note that $f_{\text{excess}} = 1$ does not mean there is a 100% chance of getting cancer; it means that the incidence of excess cancers is 100% of the observed rate within the plateau period.

Figure 21: problem 3 description

The following trials were used $\{10^4, 10^5, 10^6, 10^7\}$. The last amount is the one asked for in this problem. For each case, analytic pdf is plotted against the generated one to compare. As more trials used, the approximation to the true pdf is improved.

Figure 22: 10^4 trials, normal plotFigure 23: 10^5 trials, normal plot

Figure 24: 10^6 trials, normal plotFigure 25: 10^7 trials, normal plot

The following plots adds 10^8 number of trials in order to see how close the generated pdf will be to the theoretical pdf in the plateau region (the region between 15 and 45 years). Generated pdf becomes closer to the theoretical pdf, but there always be a small gap at the top. With more trials, the generated pdf will converge to the theoretical pdf.

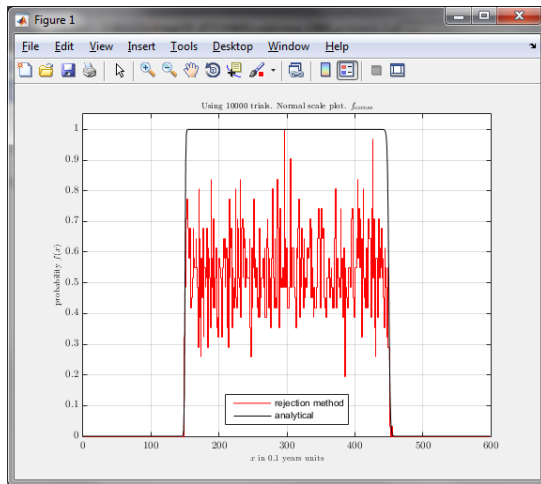
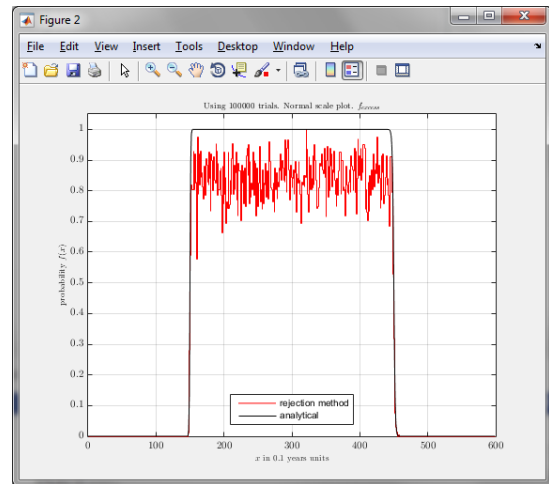
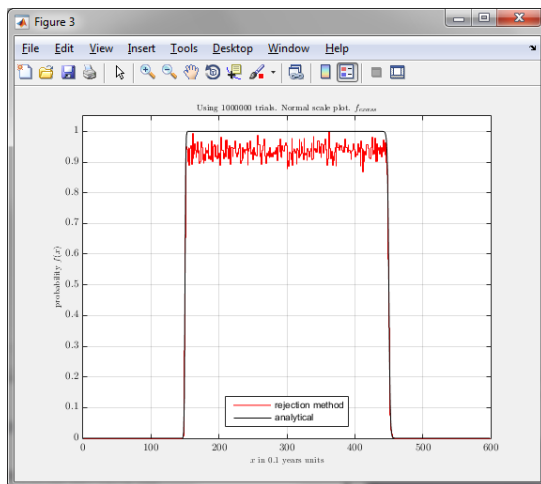
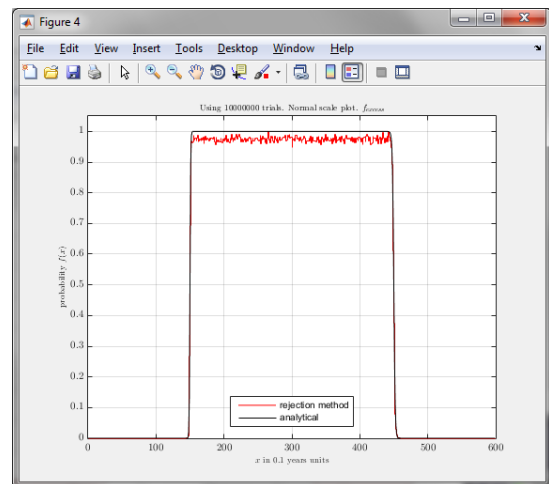
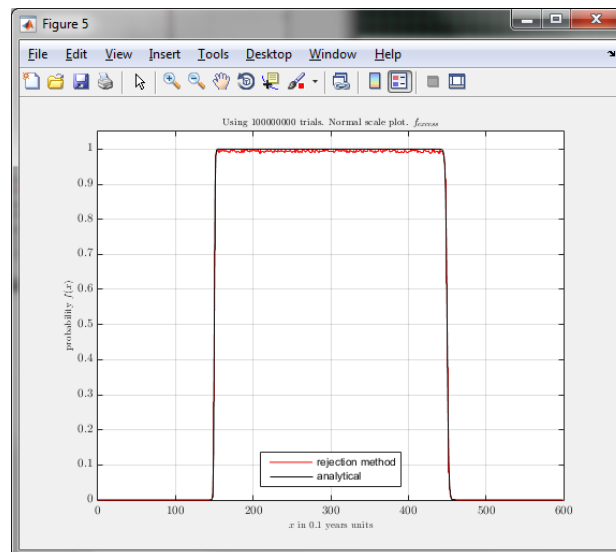
 10^4  10^5  10^6  10^7  10^8

Figure 26: $\{10^4, 10^5, 10^6, 10^7, 10^8\}$ trials all on one plot.

```

1 function nma_HW4_problem_3()
2 %Problem 2, Low Let radiation. EMA 471, HW4
3 %by Nasser M. Abbasi
4 %
5 %
6 close all; clc;
7
8 ntrials = [10^4 10^5 10^6 10^7];
9 for i=1:length(ntrials)
10     process(ntrials(i));
11 end
12 end
13 %=====
14 function process(N)
15 %N: number of trials
16
17 %Initialize random number generator
18 rng('default');
19 rng(1);
20
21 bin_width = 0.1;
22 MAX      = 60/bin_width; %60 years with 0.1 divisions
23 counts   = zeros(N,1); %to save accepted trials into
24 k        = 0; %counter for keeping accepted
25
26 for i=1:N
27     z_1 = rand();
28     z_2 = rand();
29 %convert to actual x from 0..600 (60 years, with .1 division)
30     x = z_1*MAX;
31
32     if z_2 < find_pdf_at(x,bin_width)
33         k = k + 1;
34         counts(k) = x;
35     end
36 end
37
38 generate_plots(N,counts(1:k),MAX,bin_width);
39 end
40 %=====
41 function generate_plots(number_of_trials,counts,MAX,bin_width)
42 analytical_pdf = find_pdf_at((1/2)* ...
43     bin_width:MAX-(1/2)*bin_width,bin_width);
44 generate_normal_plots(number_of_trials,counts,...
45     MAX,analytical_pdf,bin_width);
46 %generate_log_plots(number_of_trials,counts,...
47 %     MAX,analytical_pdf,bin_width);
48 end
49 %=====
50 function generate_normal_plots(number_of_trials,counts,...
51     MAX,analytical_pdf,bin_width)
52 figure;
53 edges = 0:MAX;

```

```

54 x_bins    = histcounts(counts,edges);
55
56 %use center of bins
57 x        = (1/2)*bin_width:MAX-(1/2)*bin_width;
58
59 %notice, we divide by the full area
60 stairs(x,x_bins/max(x_bins),'r');
61 hold on;
62 plot(x,analytical_pdf,'k');
63 ylim([0 1.05]);
64 grid;
65
66 title(sprintf('Using %d trials. Normal scale plot. $f_{excess}$',...
67             number_of_trials),'interpreter','Latex','FontSize',7);
68 xlabel('$x$ in 0.1 years units','interpreter','Latex','FontSize',7);
69 ylabel('probability $f(x)$','interpreter','Latex','FontSize',7);
70 set(gca,'TickLabelInterpreter','Latex','fontSize',8);
71 legend('rejection method','analytical','Location','south');
72 print(gcf, '-dpdf', '-r600', ...
73       sprintf('../images/%d_problem_3_normal_plot.pdf',...
74             number_of_trials));
75
76 end
77 %=====
78 function generate_log_plots(number_of_trials,counts,MAX,...
79                             analytical_pdf,bin_width)
80 figure;
81 edges    = 0:MAX;
82 x_bins   = histcounts( counts,edges);
83
84 %use center of bins
85 x        = (1/2)*bin_width:MAX-(1/2)*bin_width;
86 semilogy(x,x_bins/max(x_bins));
87 hold on;
88 semilogy(x,analytical_pdf);
89 %grid;
90
91 title(sprintf('Using %d trials. semilogy scale plot. $f_{excess}$',...
92             number_of_trials),'interpreter','Latex','FontSize',7);
93 xlabel('$x$', 'interpreter','Latex','FontSize',7);
94 ylabel('probability $f(D)$','interpreter','Latex','FontSize',7);
95 set(gca,'TickLabelInterpreter','Latex','fontSize',8);
96 legend('rejection method','analytical');
97 print(gcf, '-dpdf', '-r600', ...
98       sprintf('../images/%d_problem_3_log_plot.pdf',number_of_trials));
99
100 end
101 %=====
102 function r= find_pdf_at(x,bin_width)
103 L = 15/bin_width; %Latency in years
104 p = 30/bin_width; %plateau in years
105
106 r = (1/2)*(tanh(x-L)-tanh(0.5*(x-(L+p))));
107

```

108 `end`