Codes

C.

Figure : f.m file

function y=f(x)

y=exp((-x.^2)/2)/sqrt(2.\*pi);

end

Figure : Code.m file

clc;

clear all;

x=0:0.04:4;

l=length(x);

y=f(x);

A = zeros(1,l);

A(1)=0;

for i=2:l

z=0;

A(i)=A(i-1)+(f(x(i-1))+f(x(i))).\*0.02; %trapz

end

actual=normcdf(x)-0.5;

error=abs(actual-A);

plot(x,error);

grid on;

scaled\_x=0:0.01:3.99;

scaled\_y=spline(x,A,scaled\_x);

%printing table

for i=1:length(scaled\_y)

fprintf('%f ',scaled\_y(i));

if(mod(i,10)==0)fprintf('\n');end

end

Figure : max error at z=1

>>[m]=max(error)

m =3.2264e-05

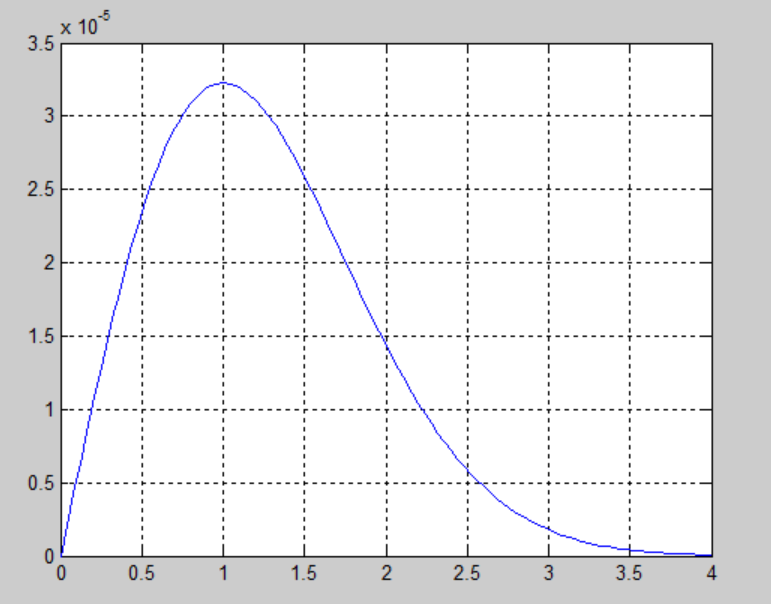


Figure : Error plot for Standard Normal Distribution

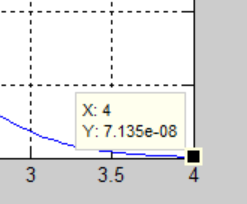
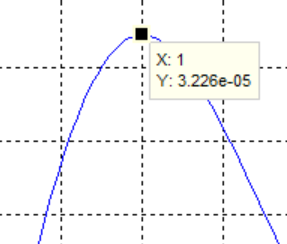


Figure :value when z=1 (maximum value)

Figure : value when z=4

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Z[y] | .00 | .01 | .02 | .03 | .04 | .05 | .06 | .07 | .08 | .09 |
| 0.0 | 0.000000 | 0.003989 | 0.007977 | 0.011965 | 0.015951 | 0.019936 | 0.023919 | 0.027899 | 0.031877 | 0.035852 |
| 0.1 | 0.039823 | 0.043789 | 0.047752 | 0.051710 | 0.055663 | 0.059610 | 0.063551 | 0.067486 | 0.071414 | 0.075336 |
| 0.2 | 0.079249 | 0.083155 | 0.087053 | 0.090942 | 0.094822 | 0.098693 | 0.102555 | 0.106406 | 0.110247 | 0.114077 |
| 0.3 | 0.117896 | 0.121704 | 0.125500 | 0.129283 | 0.133055 | 0.136813 | 0.140558 | 0.144290 | 0.148008 | 0.151712 |
| 0.4 | 0.155402 | 0.159077 | 0.162737 | 0.166381 | 0.170010 | 0.173623 | 0.177220 | 0.180800 | 0.184364 | 0.187910 |
| 0.5 | 0.191439 | 0.194950 | 0.198444 | 0.201920 | 0.205377 | 0.208815 | 0.212235 | 0.215635 | 0.219017 | 0.222378 |
| 0.6 | 0.225720 | 0.229042 | 0.232344 | 0.235625 | 0.238886 | 0.242126 | 0.245345 | 0.248543 | 0.251719 | 0.254874 |
| 0.7 | 0.258007 | 0.261119 | 0.264208 | 0.267275 | 0.270320 | 0.273343 | 0.276342 | 0.279320 | 0.282274 | 0.285205 |
| 0.8 | 0.288114 | 0.290999 | 0.293861 | 0.296699 | 0.299514 | 0.302306 | 0.305074 | 0.307818 | 0.310539 | 0.313235 |
| 0.9 | 0.315908 | 0.318557 | 0.321182 | 0.323782 | 0.326359 | 0.328912 | 0.331440 | 0.333945 | 0.336425 | 0.338881 |
| 1.0 | 0.341312 | 0.343720 | 0.346104 | 0.348463 | 0.350798 | 0.353109 | 0.355396 | 0.357658 | 0.359897 | 0.362111 |
| 1.1 | 0.364302 | 0.366469 | 0.368611 | 0.370730 | 0.372825 | 0.374896 | 0.376944 | 0.378968 | 0.380969 | 0.382946 |
| 1.2 | 0.384899 | 0.386830 | 0.388737 | 0.390621 | 0.392482 | 0.394320 | 0.396135 | 0.397928 | 0.399697 | 0.401445 |
| 1.3 | 0.403170 | 0.404873 | 0.406553 | 0.408212 | 0.409848 | 0.411463 | 0.413056 | 0.414628 | 0.416178 | 0.417707 |
| 1.4 | 0.419215 | 0.420702 | 0.422169 | 0.423614 | 0.425039 | 0.426444 | 0.427828 | 0.429193 | 0.430537 | 0.431862 |
| 1.5 | 0.433167 | 0.434453 | 0.435719 | 0.436966 | 0.438195 | 0.439404 | 0.440595 | 0.441768 | 0.442922 | 0.444059 |
| 1.6 | 0.445177 | 0.446278 | 0.447361 | 0.448426 | 0.449475 | 0.450506 | 0.451521 | 0.452518 | 0.453500 | 0.454464 |
| 1.7 | 0.455413 | 0.456346 | 0.457263 | 0.458164 | 0.459050 | 0.459921 | 0.460776 | 0.461617 | 0.462443 | 0.463254 |
| 1.8 | 0.464051 | 0.464833 | 0.465602 | 0.466357 | 0.467098 | 0.467825 | 0.468540 | 0.469241 | 0.469929 | 0.470604 |
| 1.9 | 0.471267 | 0.471917 | 0.472555 | 0.473181 | 0.473794 | 0.474396 | 0.474987 | 0.475566 | 0.476133 | 0.476690 |
| 2.0 | 0.477235 | 0.477770 | 0.478294 | 0.478808 | 0.479311 | 0.479804 | 0.480288 | 0.480761 | 0.481225 | 0.481679 |
| 2.1 | 0.482123 | 0.482559 | 0.482985 | 0.483402 | 0.483811 | 0.484211 | 0.484603 | 0.484986 | 0.485360 | 0.485727 |
| 2.2 | 0.486086 | 0.486437 | 0.486781 | 0.487116 | 0.487445 | 0.487766 | 0.488080 | 0.488387 | 0.488687 | 0.488980 |
| 2.3 | 0.489267 | 0.489547 | 0.489821 | 0.490089 | 0.490350 | 0.490605 | 0.490855 | 0.491098 | 0.491336 | 0.491569 |
| 2.4 | 0.491795 | 0.492017 | 0.492233 | 0.492444 | 0.492650 | 0.492851 | 0.493047 | 0.493238 | 0.493425 | 0.493607 |
| 2.5 | 0.493784 | 0.493958 | 0.494127 | 0.494291 | 0.494452 | 0.494609 | 0.494761 | 0.494910 | 0.495055 | 0.495196 |
| 2.6 | 0.495334 | 0.495468 | 0.495599 | 0.495726 | 0.495850 | 0.495971 | 0.496089 | 0.496203 | 0.496315 | 0.496424 |
| 2.7 | 0.496529 | 0.496632 | 0.496732 | 0.496830 | 0.496925 | 0.497017 | 0.497107 | 0.497194 | 0.497279 | 0.497362 |
| 2.8 | 0.497442 | 0.497520 | 0.497596 | 0.497670 | 0.497742 | 0.497811 | 0.497879 | 0.497945 | 0.498009 | 0.498071 |
| 2.9 | 0.498132 | 0.498191 | 0.498248 | 0.498303 | 0.498357 | 0.498409 | 0.498460 | 0.498509 | 0.498557 | 0.498603 |
| 3.0 | 0.498648 | 0.498692 | 0.498734 | 0.498776 | 0.498816 | 0.498854 | 0.498892 | 0.498928 | 0.498964 | 0.498998 |
| 3.1 | 0.499031 | 0.499063 | 0.499094 | 0.499125 | 0.499154 | 0.499182 | 0.499210 | 0.499237 | 0.499263 | 0.499288 |
| 3.2 | 0.499312 | 0.499335 | 0.499358 | 0.499380 | 0.499401 | 0.499422 | 0.499442 | 0.499461 | 0.499480 | 0.499498 |
| 3.3 | 0.499516 | 0.499533 | 0.499549 | 0.499565 | 0.499580 | 0.499595 | 0.499610 | 0.499624 | 0.499637 | 0.499650 |
| 3.4 | 0.499663 | 0.499675 | 0.499686 | 0.499698 | 0.499709 | 0.499719 | 0.499729 | 0.499739 | 0.499749 | 0.499758 |
| 3.5 | 0.499767 | 0.499776 | 0.499784 | 0.499792 | 0.499800 | 0.499807 | 0.499814 | 0.499821 | 0.499828 | 0.499834 |
| 3.6 | 0.499841 | 0.499847 | 0.499852 | 0.499858 | 0.499863 | 0.499869 | 0.499874 | 0.499878 | 0.499883 | 0.499888 |
| 3.7 | 0.499892 | 0.499896 | 0.499900 | 0.499904 | 0.499908 | 0.499911 | 0.499915 | 0.499918 | 0.499921 | 0.499925 |
| 3.8 | 0.499928 | 0.499930 | 0.499933 | 0.499936 | 0.499938 | 0.499941 | 0.499943 | 0.499945 | 0.499948 | 0.499950 |
| 3.9 | 0.499952 | 0.499954 | 0.499956 | 0.499957 | 0.499959 | 0.499961 | 0.499962 | 0.499964 | 0.499965 | 0.499967 |

Figure ::Generated Z score table

h.

Figure : g.m file

function y=g(x)

y1 = gamma(11/2)/(sqrt(10\*pi)\*gamma(5));

y2 = 1 + (x.^2)/10;

y = y1\*y2.^(-11/2);

end

clc;

clear all;

x=0:0.05:5;

l=length(x);

y=g(x);

A = zeros(1,l);

for i=1:l

x1=x(1:i);

y1=g(x1);

A(i)=0.5-trapz(x1,y1,2);

%trapz in matlab we have to use the dimension of the poly.here its 2

end

scaled\_x=0:0.001:5; %as T distribution minimum val 0.001

actual=1-tcdf(scaled\_x,10);

linearx=interp1(x,A,scaled\_x,'linear');

splinex=spline(x,A,scaled\_x);

error1=abs(linearx-actual);

error2=abs(splinex-actual);

plot(scaled\_x,error1,'r');

hold on;

plot(scaled\_x,error2,'b');

legend('Linear','Spline');

xlabel('t value');

Figure : codes2.m file

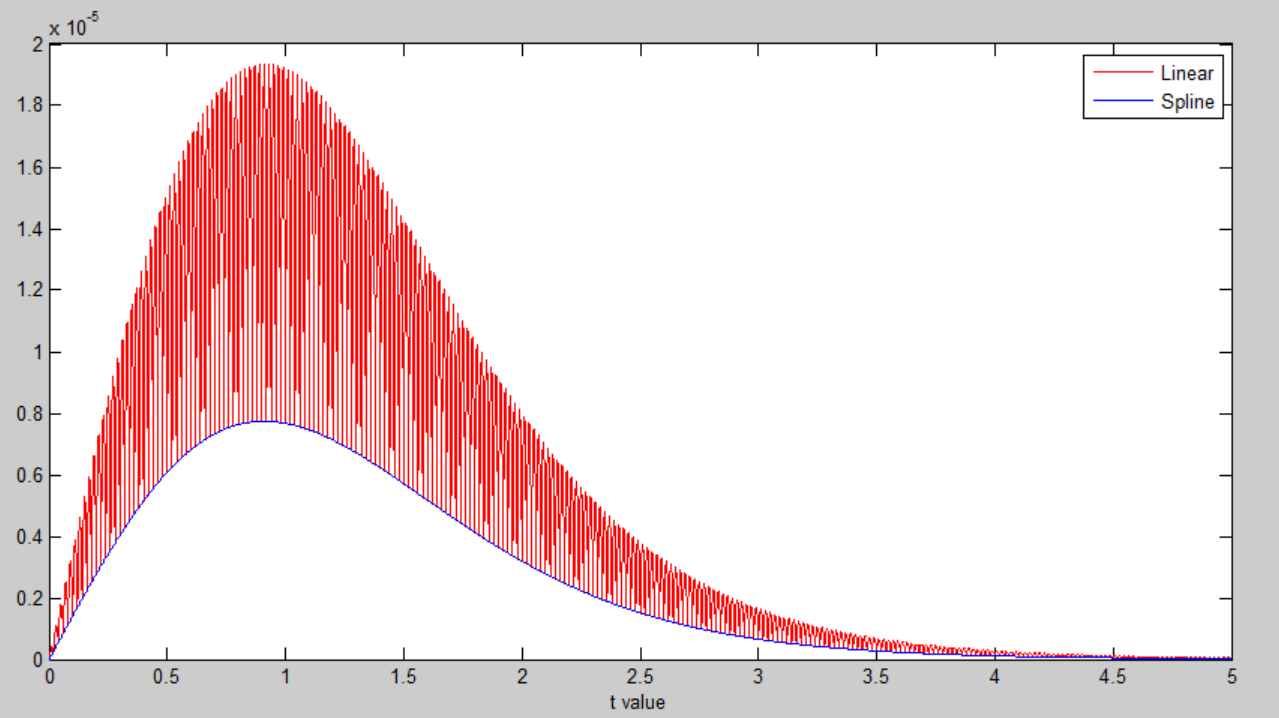
to get the values we can multiply the index of t by 1000 and add to the generated *splinex[]* array. For example to take the value of t=0.260; ***splinex[0.26\*1000]*** was taken.

Figure : Error plotted using spline and linear

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **t** | **0.260** | **0.300** | **0.500** | **0.700** | **1.000** | **1.372** | **1.500** | **1.812** | **2.000** | **2.228** |
| **P(t <= T)** | 0.40 | 0.3855 | 0.3143 | 0.25 | 0.1707 | 0.10 | 0.0824 | 0.05 | 0.0368 | 0.025 |
| **t** | **2.500** | **2.764** | **2.900** | **3.169** | **3.400** | **3.581** | **3.750** | **4.4144** | **4.300** | **4.587** |
| **P(t <= T)** | 0.0158 | 0.01 | 0.0079 | 0.005 | 0.0034 | 0.0025 | 0.0019 | 0.0001 | 0.0008 | 0.0005 |

Figure : T distribution