sci_gsl TOOLBOX

10 June 2020- 30 June 2020

TEAM: STH109

Dr. Rajan Goyal
Dr. Rashmi Verma
Ms. Sakshi Verma(B.Sc(H) Physics II year)
Ms. Mahiguhappriyaprakash(B.Sc(H) Physics I year)

Department of Physics
Shaheed Rajguru College of Applied Sciences for Women
(University of Delhi)
Vasundhara Enclave, Delhi-110096

1 A Quick Glance

- Name of the toolbox:
  sci_gsl Toolbox
- Purpose of the toolbox:
  Interfacing of gsl function in scilab so that one can use these in a more efficient manner
- Target Operating System:
  Ubuntu(Linux) 20.04 and 18.04 versions
  (Tested on Scilab 6.0.2)
- Name of the external software/library that the Toolbox interfaces to:
  GNU scientific library
- Link to the source code of the chosen external software/library:
  `ftp://ftp.gnu.org/gnu/gsl/gsl-latest.tar.gz`
- Detailed step by step build instructions to compile the source code of the chosen external software/library:
  Section 3
- Software pre-requisites for building the scilab toolbox with suitable instructions to satisfy them:
  gcc and g++ compiler
- Detailed step-by step instructions to build the toolbox:
  Section 4
- Detailed step-by-step instructions to load the toolbox:
  Section 4

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

Contents

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

2 Directions to download and install GSL

- Download the latest version of the library (gsl-latest.tar.gz) from the link:

  ftp://ftp.gnu.org/gnu/gsl/

- Place the file in your home directory and unpack the file with the following command:

  tar -zxvf gsl-2.6.tar.gz

  This will create a directory called gsl-2.6 in your home directory. Change to this directory.

  cd gsl-2.6
  ./configure && make sudo
  make install
  If this doesn't work then follow the steps given below:

- The next step is to configure the installation and tell the system where to install the files. Create a directory to install your gsl package, say "/home/yourname/gsl" with the following command

  mkdir /home/yourname/gsl

- Now configure the installation and tell it to use your new directory with the following code. This step may take a few minutes.

  ./configure –prefix=/home/yourname/gsl

- If there are no errors, compile the library. This step will take several minutes.

  make

- Now it is necessary to check and test the library before actually installing it. This step will take some time.

  make check

- If there are no errors, go ahead and install the library with:

  make install

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

3 Compiling And linking the program with GSL

Now we can write a test program to see if the library works. Create the following program and name it example.c

```
    #include <stdio.h>
#include <gsl/gsl_sf_bessel.h>
int main (void)
{
double x = 15.0;
double y = gsl_sf_bessel_J0 (x);
printf ("J0(%g) = %.18e/n", x, y);
return 0;
}
```

Compile and link the program with the following commands (but use the correct path for your username):

```
    gcc -Wall -I/home/yourname/gsl/include -c example.c
gcc -L/home/yourname/gsl/lib example.o -lgsl -lgslcblas -lm
```

Now run your program!

```
./a.out
```

4 Introduction to the Toolbox

This repository is a toolbox that consists of functions that computes integration of some special functions with Monte Carlo Method, normalized radial wavefunction of hydrogen like atoms, computes 45 special functions like Airy's function and Bessel function, returns the values of about 25 physical constants in S.I or MKS units, computes the Associated Legendre Polynomial and Normalized Associated Legendre Polynomial which are suitable for spherical harmonics, solves linear system of equations using LU decomposition, QR decomposition and Householder method, computes distribution functions like Gaussian distribution and Student's T-distribution for Scilab. One has to install GSL(GNU Scientific Library) in their system to work with this toolbox. Extract the file in a directory which is easily accessible to you. On the scilab console run "exec builder.sce" to build the toolbox and then run "exec loader.sce" to load the toolbox. Type "help" in the scilab console and browse through the help content of "sci_gsl".

5 An Introduction to Monte Carlo Integration

Monte Carlo integration is a technique for numerical integration using random numbers. The method relies on random sampling to approximate the results. The more the random data points, the more approximate the result will be, i.e., more number of random data points results in better accuracy in estimating the integration. There are different methods in Monte Carlo integration:
1. Uniform Sampling (PLAIN)

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

2. Stratified Sampling (MISER)
3. Importance Sampling (VEGAS)

- PLAIN Monte Carlo:
  The plain Monte Carlo algorithm samples points randomly from the integration region to estimate the integral and its error. Using this algorithm the estimate of the integral $E(f;N)$ for N randomly distributed points x_i is given by,

  $E(f;N) = V < f >= \frac{V}{N}\sum_i^N f(x_i)$

  where V is the volume of the integration region. The error on this estimate $\sigma(E;N)$ is calculated from the estimated variance of the mean,

  $\sigma^2(E;N) = \frac{V^2}{N^2}\sum_i^N (f(x_i) - < f >)^2$

- MISER Monte Carlo:
  The MISER algorithm of Press and Farrar is based on recursive stratified sampling. This technique aims to reduce the overall integration error by concentrating integration points in the regions of highest variance.

  The idea of stratified sampling begins with the observation that for two disjoint regions a and b with Monte Carlo estimates of the integral $E_a(f)$ and $E_b(f)$ and variances $\sigma_a^2(f)$ and $\sigma_b^2(f)$, the variance $Var(f)$ of the combined estimate $E(f) = \frac{1}{2}(E_a(f) + E_b(f))$ is given by,

  $Var(f) = \frac{\sigma_a^2(f)}{4N_a} + \frac{\sigma_b^2(f)}{4N_b}$

- VEGAS Monte Carlo:
  The VEGAS algorithm of Lepage is based on importance sampling. It samples points from the probability distribution described by the function $|f|$, so that the points are concentrated in the regions that make the largest contribution to the integral.

  In general, if the Monte Carlo integral of f is sampled with points distributed according to a probability distribution described by the function g, we obtain an estimate $E_g(f;N)$,

  $E_g(f;N) = E(f/g;N)$

  with a corresponding variance,

  $Var_g(f;N) = Var(f/g;N)$

6 Associated Legendre (Ass_legendre)

It computes the Associated Legendre Polynomial.

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

6.1 Syntax

$y = Ass\_legendre(l, m, x)$

6.2 Parameters

y : output
l and m : are integers. m greater than equal to 0 and l greater than equal to m.
x : is a real number lies in the range [-1,1]

6.3 Description

This function computes the Associated Legendre Polynomial for given values of l, m and x.

The associated Legendre polynomials $P_l^m(x)$ are solutions of the differential equation

$(1-x^2)\frac{d^2}{dx^2}P_l^m(x) - 2x\frac{d}{dx}P_l^m(x) + (l(l+1) - \frac{m^2}{(1-x^2)})P_l^m(x) = 0$ where the degree l and order m satisfy 0≤l and 0≤m≤l. The functions $P_l^m(x)$ grow combinatorially with l and can overflow for l larger than about 150.

7 Monte Carlo integration (montecarlo)

Performs Monte Carlo integration to calculate the values of various integrals and special functions.

7.1 Syntax

$y = montecarlo(intgeral\_function, c1, c2, lo\_limit, up\_limit, N\_itr, method)$

7.2 Parameters

y : output
$c1$ and $c2$: constant which depends on the type of the integral.
lo_limit and up_limit : lower and upper limit of the integral.
N_itr : number of times the user wants to perform the Monte Carlo integration to calculate the value of integral. Larger the value of N_itr better will be the accuracy.
method : Three different Monte Carlo integration algorithms namely Plain Monte Carlo, Miser Monte Carlo and Vegas Monte Carlo are available. The user can set the *method* to select the desired algorithm for Monte Carlo integration. For $method = 1$, plain Monte Carlo algorithm will be used, $method = 2$ miser Monte Carlo algorithm will be used and for all other values of the parameter *method* the function will use Vegas Monte Carlo algorithm. Out of these three algorithm, the vegas Monte Carlo algorithm is the most accurate while the plain Monte Carlo algorithm is the least accurate.
$integral\_function$: user would have to select the value from 0 to 8 for the

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

function for which they want to perform the Monte Carlo Integration.

- $integral\_function = 1$

  Evaluate Gamma function Γn, $c1$ represents n and can be only non-negative fraction or integer while $c2 = 0$, $lo\_limit = 0$, up_limit is infinite, can be set to a large integer e.g. 100.

  The Gamma function is defined by the following integral,
  $\Gamma(x) = \int_0^\infty dt t^{x-1} \exp(-t)$.

- $integral\_function = 2$

  Gaussian function $G(\mu, \sigma) = \int_{-\infty}^{+\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} dx$
  Evaluate Gaussian function $G(\mu, \sigma)$, $c1$ and $c2$ represent the mean μ and standard deviation σ, lo_limit and up_limit are $-\infty$ and $+\infty$ respectively and can be set to a large integer e.g. 100.

- $integral\_function = 3$

  Evaluate Airy's Function $Ai(x)$, $c1$ represent x, can take any value while $c2 = 0$, lo_limit and up_limit are 0 and $+\infty$ respectively and can be set to a large integer e.g. 100.

  Airy's Function $Ai(x)$, $c1$ represents the x while $c2$ can be set to any value.
  $Ai(x) = \frac{1}{\pi} \int_0^\infty \cos(t^3/3 + xt)\ dt$
  where $-\infty \leq x \leq +\infty$.

- $integral\_function = 4$

  Evaluate Airy's Function $Bi(x)$, $c1$ represent x, can take any value while $c2 = 0$, lo_limit and up_limit are 0 and $+\infty$ respectively and can be set to a large integer e.g. 100.

  Airy's Function $Bi(x)$, $c1$ represents the x while $c2$ can be set to any value.
  $Bi(x) = \frac{1}{\pi} \int_0^\infty (e^{-t^3/3+xt} + \sin(t^3/3 + xt))\, dt$
  where $-\infty \leq x \leq +\infty$.

- $integral\_function = 5$

  Evaluate Elliptic integral of first kind $F(\phi, k)$, $c1$ represents k and lies in [-1,1], $c2 = 0$, lo_limit =0 and up_limit lies in $[0, \pi/2]$.
  $F(\phi, k) = \int_0^\phi dt \frac{1}{\sqrt{(1-k^2 \sin^2(t))}}$

- $integral\_function = 6$

  Evaluate Elliptic integral of second kind $E(\phi, k)$, $c1$ represents k and lies in [-1,1], $c2 = 0$, lo_limit =0 and up_limit lies in $[0, \pi/2]$.
  $E(\phi, k) = \int_0^\phi dt \sqrt{(1 - k^2 \sin^2(t))}$

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

- $integral\_function = 7$

  Evaluate Beta function $\beta(m, n)$, $c1$ and $c2$ represent m and n respectively and are positive integers, $lo\_limit = 0$, $up\_limit = 1$.
  $\beta(m, n) = \frac{\Gamma n \Gamma m}{\Gamma(m+n)}$

- $integral\_function \geq 8$

  Evaluate Bessel Function of first kind $J_n(x)$, $c1$ represents n and will be an integer , $c2$ represents x and will be a real number, $lo\_limit = 0$ and $up\_limit = \pi$.

  The Bessel functions of the first kind $J_n(x)$ are defined as the solutions to the Bessel differential equation,
  $x^2 \frac{d^2}{dx^2} y + x \frac{d}{dx} y + (x^2 - n^2)y = 0$

7.3 Description

This function accepts choice, two numbers, lower limit, upper limit and number of calls to perform Monte Carlo integration.

Note that n is not required to be an integer.

8 Normalized Associated Legendre (NormAss_legendre)

It computes the Normalized Associated Legendre Polynomials suitable for use in spherical harmonics

8.1 Syntax

$y = NormAss\_legendre(l, m, x)$

8.2 Parameters

y : output
l and m : are integers. m greater than equal to 0 and l greater than equal to m.
x : is a real number lies in the range [-1,1]

8.3 Description

This function computes the Normalized Associated Legendre Polynomial for given values of l, m and x.

The normalized associated Legendre polynomials are defined as $Y_l^m(x) = (-1)^m \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} P_l^m(x)$
where $P_l^m(x)$ is the associated legendre polynomial.

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

9 Physical constants (phyconst)

It gives the value of various physical constants such speed of light c, boltzmann constant k, mass and charge of electron, proton and neutron etc. in S.I or MKS units.

9.1 Syntax

$y = phyconst(constant)$

9.2 Parameters

y : output
constant :
It is used to choose the desired physical constant and can be set from 1 to 25.

- *constant* = 1: Speed of light c.
- *constant* = 2: Vacuum Permeability μ_0.
- *constant* = 3: Vacuum Permittivity ϵ_0.
- *constant* = 4: Planck's Constant h.
- *constant* = 5: Reduced Planck's Constant $\hbar = \frac{h}{2\pi}$.
- *constant* = 6: Avogadro's Number N_a.
- *constant* = 7: Boltzmann Constant k.
- *constant* = 8: Molar Gas constant R_0.
- *constant* = 9: Stefan Boltzmann Constant σ.
- *constant* = 10: Gravitational Constant G.
- *constant* = 11: Value of 1 light year in metre.
- *constant* = 12: Acceleration due to gravity g.
- *constant* = 13: Magnitude of charge of electron e.
- *constant* = 14: Mass of muon m_μ.
- *constant* = 15: Mass of proton m_p.
- *constant* = 16: Mass of neutron m_n.
- *constant* = 17: Mass of electron m_e.
- *constant* = 18: Electromagnetic hyperfine structure constant α.
- *constant* = 19: Rydberg Constant R_y.

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

- *constant* = 20: Bohr's radius a_0.
- *constant* = 21: Bohr Magneton μ_B.
- *constant* = 22: Nuclear Magneton μ_N.
- *constant* = 23: Magnitude of magnetic moment of electron μ_e.
- *constant* = 24: Magnitude of magnetic moment of proton μ_p.
- *constant* = 25: Thomson cross-section σ_T.

9.3 Description

This function takes the parameter *constant* and provides the value of various physical constants that can be directly used in various calculations.

10 Radial wavefunction of hydrogen like atoms (Rnl_Hlike)

It computes the normalized radial wavefunction of hydrogen like atoms.

10.1 Syntax

$y = Rnl\_Hlike(n, l, Z, r)$

10.2 Parameters

y : output
n : Principal Quantum Number.
l : Azimuthal Quantum number.
Z : Atomic Number of Hydrogen like atom.
r : radial distance.

10.3 Description

This function computes the normalized radial wavefunction of hydrogen like atoms.

$$R_n = \frac{2Z^{3/2}}{n^2}\left(\frac{2Zr}{n}\right)^l \sqrt{\frac{(n-l-1)!}{(n+l)!}} \exp(-Zr/n) L^{2l+1}_{n-l-1}(2Zr/n).$$

11 Special functions (splfunc)

It computes the value of special functions e.g. Airy's function, Gamma function, Beta function, Bessel's function, Legendre's polynomial etc.

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

11.1 Syntax

$y = splfunc(choice, c1, c2)$

11.2 Parameters

y : output
$c1$ and $c2$: real or integer numbers depending on the type of special function to be evaluated.
choice: It is used to choose the desired special function and is always a positive integer i.e. $choice > 0$.

- $choice = 1$
  Airy's Function $Ai(x)$, $c1$ represents the x while $c2$ can be set to any value.
  $Ai(x) = \frac{1}{\pi}\int_0^\infty \cos(t^3/3 + xt)\ dt$.

- $choice = 2$
  Airy's Function $Bi(x)$, $c1$ represents the x while $c2$ can be set to any value.
  $Bi(x) = \frac{1}{\pi}\int_0^\infty (e^{-t^3/3+xt} + \sin(t^3/3 + xt))\, dt$.

- $choice = 3$
  Scaled Airy's Function $Ai(x)$, $c1$ represents the x while $c2$ can be set to any value.

  These routines compute a scaled version of the Airy function $S_A(x)Ai(x)$. For $x > 0$ the scaling factor $S_A(x)$ is $\exp(+(2/3)x^{3/2})$, and is 1 for $x < 0$.

- $choice = 4$
  Scaled Airy's Function $Bi(x)$, $c1$ represents the x while $c2$ can be set to any value.

  These routines compute a scaled version of the Airy function $S_B(x)Bi(x)$. For $x > 0$ the scaling factor $S_B(x)$ is $\exp(-(2/3)x^{3/2})$, and is 1 for $x < 0$.

- $choice = 5$
  Derivative of Airy's Function $Ai(x)$, $c1$ represents the x while $c2$ can be set to any value. $Ai'(x)$.

  These routines compute the Airy function derivative $Ai'(x)$.

- $choice = 6$
  Derivative of Airy's Function $Bi(x)$, $c1$ represents the x while $c2$ can be set to any value. $Bi'(x)$.

  These routines compute the Airy function derivative $Bi'(x)$.

- $choice = 7$
  Scaled Derivative of Airy's Function $Ai(x)$, $c1$ represents the x while $c2$ can be set to any value.

  These routines compute the scaled Airy function derivative $S_A(x)Ai'(x)$. For $x > 0$ the scaling factor $S_A(x)$ is $\exp(+(2/3)x^{3/2})$, and is 1 for $x < 0$.

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

- *choice* = 8
  Scaled Derivative of Airy's Function $Bi(x)$, $c1$ represents the x while $c2$ can be set to any value.

  These routines compute the scaled Airy function derivative $S_B(x)Bi'(x)$. For $x > 0$ the scaling factor $S_B(x)$ is $\exp(-(2/3)x^{3/2})$, and is 1 for $x < 0$.

- *choice* = 9
  n^{th} Zeros of Airy's Function $Ai(x)$, $c1$ can be set to any value while $c2$ will be an integer representing n^{th} zero.

- *choice* = 10
  n^{th} Zeros of Airy's Function $Bi(x)$, $c1$ can be set to any value while $c2$ will be an integer representing n^{th} zero.

- *choice* = 11
  n^{th} Zeros of Derivative of Airy's Function $Ai(x)$, $c1$ can be set to any value while $c2$ will be an integer representing n^{th} zero.

  These routines compute the location of the n^{th} zero of the Airy function derivative $Ai'(x)$.

- *choice* = 12
  n^{th} Zeros of Derivative of Airy's Function $Bi(x)$, $c1$ can be set to any value while $c2$ will be an integer representing n^{th} zero.

  These routines compute the location of the n^{th} zero of the Airy function derivative $Bi'(x)$.

- *choice* = 13
  Regular Cylindrical Bessel's Function $J_n(x)$, $c1$ represents x and $c2$ represents n. Here n is an integer.

  These routines compute the regular cylindrical Bessel function of order n, $J_n(x)$.

- *choice* = 14
  Irregular Cylindrical Bessel's Function $Y_n(x)$, $c1$ represents x, $x > 0$ and $c2$ represents n. Here n is greater than or equal to 0.

  These routines compute the irregular cylindrical Bessel function of order n, $Y_n(x)$, for $x > 0$ and $n \geq 0$.

- *choice* = 15
  Regular Modified Cylindrical Bessel's Function $I_n(x)$, $c1$ represents x, $x > 0$ and $c2$ represents n. Here n is greater than or equal to 0.

  These routines compute the regular modified cylindrical Bessel function of order n, $I_n(x)$, for $n \geq 0$.

- *choice* = 16
  Scaled Regular Modified Cylindrical Bessel's Function $I_n(x)$, $c1$ represents x and $c2$ represents n. Here n is an integer.

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

These routines compute the scaled regular modified cylindrical Bessel function of order n, $\exp(-|x|)I_n(x)$.

- *choice* = 17
  Irregular Modified Cylindrical Bessel's Function $K_n(x)$, $c1$ represents x, $x > 0$ and $c2$ represents n. Here n is an integer.

  These routines compute the irregular modified cylindrical Bessel function of order n, $K_n(x)$, for $x > 0$.

- *choice* = 18
  Scaled Irregular Modified Cylindrical Bessel's Function $K_n(x)$, $c1$ represents x, $x > 0$ and $c2$ represents n. Here n is an integer.

  These routines compute the scaled irregular modified cylindrical Bessel function of order n, $\exp(+|x|)K_n(x)$, for $x > 0$.

- *choice* = 19
  Regular Spherical Bessel's Function $j_l(x)$, $c1$ and $c2$ represent x and l respectively. x is real and l is an integer. x and l both are greater than or equal to zero.

  These routines compute the regular spherical Bessel function of order l, $j_l(x)$, for $l \geq 0$ and $x \geq 0$.

- *choice* = 20
  Irregular Spherical Bessel's Function $y_l(x)$, $c1$ and $c2$ represent x and l respectively. x is real and l is an integer. x and l both are greater than or equal to zero.

  These routines compute the irregular spherical Bessel function of order l, $y_l(x)$, for $l \geq 0$ and $x \geq 0$.

- *choice* = 21
  Scaled Regular Modified Spherical Bessel's Function $i_l(x)$, $c1$ and $c2$ represent x and l respectively. x is real and l is an integer. x and l both are greater than or equal to zero.

  These routines compute the scaled regular modified spherical Bessel function of order l, $\exp(-|x|)i_l(x)$, for $l \geq 0$ and $x \geq 0$.

- *choice* = 22
  Scaled Irregular Modified Spherical Bessel's Function $k_l(x)$, $c1$ and $c2$ represent x and l respectively. x is real and l is an integer. x and l both are greater than or equal to zero.

  These routines compute the scaled irregular modified spherical Bessel function of order l, $\exp(+|x|)k_l(x)$, for $x \geq 0$ and $l \geq 0$.

- *choice* = 23
  Regular Cylindrical Bessel's Function $J_n(x)$, $c1$ represents x, $x > 0$ and $c2$ represents n. Here n is a fraction.

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

These routines compute the regular cylindrical Bessel function of order n, $J_n(x)$.

- *choice* = 24
  Irregular Cylindrical Bessel's Function $Y_n(x)$, $c1$ represents x, $x > 0$ and $c2$ represents n. Here n is a fraction.

  These routines compute the irregular cylindrical Bessel function of order n, $Y_n(x)$, for $x > 0$.

- *choice* = 25
  Regular Modified Cylindrical Bessel's Function $I_n(x)$, $c1$ represents x, $x > 0$ and $c2$ represents n. Here n is a fraction, $n > 0$.

  These routines compute the regular modified cylindrical Bessel function of order n, $I_n(x)$, for $n > 0$.

- *choice* = 26
  Scaled Regular Modified Cylindrical Bessel's Function $I_n(x)$, $c1$ represents x, $x > 0$ and $c2$ represents n. Here n is a fraction, $n > 0$.

  These routines compute the scaled regular modified cylindrical Bessel function of order n, $\exp(-|x|)I_n(x)$.

- *choice* = 27
  Irregular Modified Cylindrical Bessel's Function $K_n(x)$, $c1$ represents x, $x > 0$ and $c2$ represents n. Here n is a fraction, $n > 0$.

  These routines compute the irregular modified cylindrical Bessel function of order n, $K_n(x)$, for $x > 0$.

- *choice* = 28
  Scaled Irregular Modified Cylindrical Bessel's Function $K_n(x)$, $c1$ represents x, $x > 0$ and $c2$ represents n. Here n is a fraction, $n > 0$.

  These routines compute the scaled irregular modified cylindrical Bessel function of order n, $exp(+|x|)K_n(x)$, for $x > 0$.

- *choice* = 29
  n^{th} Zero of Regular Cylindrical Bessel Function $J_0(x)$, $c1$ can be set to any value while $c2$ represents n and is a positive integer.

  These routines compute the n^{th} zero of regular cylindrical Bessel function of zeroth order, $J_0(x)$, for $n > 0$.

- *choice* = 30
  n^{th} Zero of Regular Cylindrical Bessel Function $J_1(x)$, $c1$ can be set to any value while $c2$ represents n and is a positive integer.

  These routines compute the n^{th} zero of regular cylindrical Bessel function of first order, $J_1(x)$.

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

- *choice* = 31
  m^{th} Zero of Regular Cylindrical Bessel Function $J_n(x)$, $c1$ represents m while $c2$ represents n. Here n is a positive fraction.

  These routines compute the m^{th} zero of regular cylindrical Bessel function of order n, $Jn(x)$.

- *choice* = 32
  Dawson Function or Integral $Da(x)$, c1 represents x while c2 can be set to any value.

  The Dawson integral is defined by $\exp(-(x^2)) \int_0^x dt \exp(t^2)$.

- *choice* = 33
  Debye Function or Integral of first order $D_1(x)$, $c1$ represents x and $x > 0$ while $c2$ can be set to any value.

  The Debye function $D_1(x)$ is defined by the following integral,
  $D_1(x) = \frac{1}{x} \int_0^x dt \frac{t}{e^t - 1}$.

- *choice* = 34
  Debye Function or Integral of second order $D_2(x)$, $c1$ represents x and $x > 0$ while $c2$ can be set to any value.

  The Debye function $D_2(x)$ is defined by the following integral,
  $D_2(x) = \frac{2}{x^2} \int_0^x dt \frac{t^2}{e^t - 1}$.

- *choice* = 35
  Error Function $erf(x)$, $c1$ represents x and $x > 0$ while $c2$ can be set to any value.

  These routines compute the error function $erf(x)$, where,
  $erf(x) = (2/\sqrt{\pi}) \int_0^x dt \exp(-t^2)$.

- *choice* = 36
  Complementary Error Function $erfc(x)$, $c1$ represents x and $x > 0$ while $c2$ can be set to any value.

  These routines compute the complementary error function,
  $erfc(x) = 1 - erf(x) = (2/\sqrt{\pi}) \int_x^\infty dt \exp(-t^2)$.

- *choice* = 37
  Log Complementary Error Function, $c1$ represents x and $x > 0$ while $c2$ can be set to any value.

  These routines compute the logarithm of the complementary error function $\log(erfc(x))$.

- *choice* = 38
  Normalized Normal or Gaussian Probability Density Function $N \sim (0, 1)$, $c1$ represents x while $c2$ can be set to any value.

  These routines compute the Gaussian probability density function,
  $Z(x) = (1/\sqrt{2\pi}) \exp(-x^2/2)$.

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

- *choice* = 39
  Upper tail Normalized Gaussian probability density function $N \sim (0, 1)$, $c1$ represents x while $c2$ can be set to any value.

  These routines compute the upper tail of the Gaussian probability function, $Q(x) = (1/\sqrt{2\pi}) \int_x^\infty dt \exp(-t^2/2)$.

- *choice* = 40
  Real part of the Exponential Integral $E_n(x)$, $c1$ and $c2$ represent x and n respectively. n is positive real number.

  These routines compute the exponential integral $E_n(x)$,
  $E_n(x) = \Re \int_1^\infty dt \exp(-xt)/t^n$.

- *choice* = 41
  Fermi Dirac Integral $F_{\frac{1}{2}}(x)$, $c1$ represents x and x is real number while $c2$ represents n and n is positive integer.

  The complete Fermi-Dirac integral $F_j(x)$ is given by,
  $F_j(x) = \frac{1}{\Gamma(j+1)} \int_0^\infty dt \frac{t^j}{(\exp(t-x)+1)}$.

- *choice* = 42
  Gamma Function, $c1$ represent n and can be set to an integer or fraction. $c1$ cannot be set equal to zero and $c2$ can be set to any value.

  The Gamma function is defined by the following integral,
  $\Gamma(x) = \int_0^\infty dt t^{x-1} \exp(-t)$.

- *choice* = 43
  Beta function, $c1$ and $c2$ represent m and n. Both m and n are positive integers.

  These routines compute the Beta Function,
  $\beta(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$ subject to a and b being positive integers.

- *choice* = 44
  Laguerre Polynomial of order n $L_n(x)$, $c1$ and $c2$ represent x and n respectively. n is a non-negative integer.

  The generalized Laguerre polynomials, sometimes referred to as associated Laguerre polynomials, are defined in terms of confluent hypergeometric functions as $L_n^a(x) = \frac{(a+1)_n}{n!} {}_1F_1(-n, a+1, x)$

- *choice* = 45
  Legendre's Polynomial of order n $P_n(x)$, $c1$ and $c2$ represent x and n respectively. n is a non-negative integer.

11.3 Description

This function takes the parameters *choice* and two numbers to calculate the special functions.

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

12 Distribution Functions

Beta Distribution (betadist)

Beta Distribution

a. Syntax

$y = betadist(a, b, x, c)$

b. Parameters

y : output
a, b : parameters
x : x is real number lying within the range 0 and 1 for cdf and pdf computation. while for inverse cdf x will represent P and Q and its value will lie between 0 and 1.

- $c = 1$:
  It will allow the user to generate a beta random variate. In this case the function is independent of x and hence any value of x can be set.
- $c = 2$:
  It will allow the user to compute the probability density $p(x)$ at x for a beta distribution.
- $c = 3$:
  It will allow the user to compute beta cdf $P(x)$
- $c = 4$:
  It will allow the user to compute beta cdf $Q(x)$.
- $c = 5$:
  It will allow the user to compute inverse beta cdf $P(x)$.
- $c \geq 6$:
  It will allow the user to compute inverse beta cdf $Q(x)$.

c. Description

This function is used to calculate beta random variate, probability density, cumulative distribution functions $P(x)$, $Q(x)$ and their inverse.
The distribution function is,
$p(x)dx = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1}dx$
for $0 \leq x \leq 1$.

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

Bivariate Gaussian Distribution (bigaussian)

Bivariate Gaussian Distribution: Generate a pair of correlated Gaussian variates.

a. Syntax

$[x, y] = bigaussian(mx, my, sx, sy, rho)$

b. Parameters

x, y : output
mx : mean in the x direction.
my : mean in the y direction.
sx : standard deviation in the x direction.
sy : standard deviation in the y direction.
rho : It is the correlation coefficient and lies in between -1 and +1

c. Description

This function generates a pair of correlated Gaussian variates, with mean mx and my, correlation coefficient rho and standard deviations sx and sy in the x and y directions

Bivariate Gaussian Distribution Probability Density (bigaussianpdf)

Compute probability density p(x,y) at (x,y) for a bivariate Gaussian distribution.

a. Syntax

$z = bigaussian(mx, my, sx, sy, rho, x, y)$

b. Parameters

z : output
mx : mean in the x direction (μ_x).
my : mean in the y direction (μ_y).
sx : standard deviation in the x direction (σ_x).
sy : standard deviation in the y direction (σ_y).
rho : It is the correlation coefficient and lies in between -1 and +1 (ρ).
x, y : correlated Gaussian variates at which probability density $p(x, y)$ is to be computed.

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

c. Description

This function computes the probability density $p(x, y)$ at (x, y) for a bivariate Gaussian distribution with mean μ_x,μ_y and standard deviations σ_x, σ_y and correlation coefficient ρ.
The probability distribution for bivariate Gaussian random variates is,

$p(x, y)dxdy =$
$\frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp\left(-\frac{((x-\mu_x)^2/\sigma_x^2+(y-\mu_y)^2/\sigma_y^2-2\rho(x-\mu_x)(y-\mu_y)/(\sigma_x\sigma_y))}{2(1-\rho^2)}\right) dxdy$

for x,y in the range $-\infty$ to $+\infty$. The correlation coefficient ρ should lie between 1 and -1.

Binomial Distribution (binomial)

Binomial Distribution

a. Syntax

$y = binomial(p, n, k, c)$

b. Parameters

y : output
p : probability of success
n : Total number of independent trials. It takes only positive integer values.
k : k is number of success. It can be only within the range 0 and n.

- $c = 1$:
  It will allow the user to generate a binomial random variate, the number of success in n independent trials with probability p. In this case the function is independent of k and hence any value of k can be set.
- $c = 2$:
  It will allow the user to compute the probability $p(k)$ of obtaining k from a binomial distribution with parameters p and n.
- $c = 3$:
  It will allow the user to compute binomial cdf $P(k)$
- $c \geq 4$:
  It will allow the user to compute binomial cdf $Q(k)$

c. Description

This function is used to calculate binomial random variate, probability $p(k)$ of obtaining k from a binomial distribution and cumulative distribution functions $P(k)$, $Q(k)$.
The probability distribution for binomial variates is,

$p(k) = \frac{n!}{k!(n-k)!}p^k(1-p)^{n-k}$
for $0 \leq k \leq n$.

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

Chi-squared Distribution (chisq)

Chi-squared Distribution

a. Syntax

$y = chisq(n, x, c)$

b. Parameters

y : output
n : degrees of freedom (ν)
x : x is real number lying within the range 0 to $+\infty$ for cdf and pdf computation. while for inverse cdf x will represent P and Q and its value will lie between 0 and 1.

- $c = 1$:
  It will allow the user to generate chi-squared random variate with ν degrees of freedom. In this case the function is independent of x and hence any value of x can be set.
- $c = 2$:
  It will allow the user to compute the probability density $p(x)$ at x for chi-squared distribution with ν degrees of freedom.
- $c = 3$:
  It will allow the user to compute chi-squared cdf $P(x)$
- $c = 4$:
  It will allow the user to compute chi-squared cdf $Q(x)$.
- $c = 5$:
  It will allow the user to compute inverse chi-squared cdf $P(x)$.
- $c \geq 6$:
  It will allow the user to compute inverse chi-squared cdf $Q(x)$.

c. Description

This function is used to calculate chi-squared random variate, probability density, cumulative distribution functions $P(x)$, $Q(x)$ and their inverse.
The distribution function is,
$p(x)dx = \frac{1}{2\Gamma(\nu/2)}(x/2)^{\nu/2-1}\exp(-x/2)dx$ where ν represents degrees of freedom and $x \geq 0$.

Exponential Distribution (exponential)

Exponential Distribution

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

a. Syntax

$y = exponential(m, x, c)$

b. Parameters

y : output
m : mean and is greater than 0 (μ)
x : x is real number lying within the range 0 to $+\infty$ for cdf and pdf computation. while for inverse cdf x will represent P and Q and its value will lie between 0 and 1.

- $c = 1$:
  It will allow the user to generate an exponential random variate with mean μ. In this case the function is independent of x and hence any value of x can be set.
- $c = 2$:
  It will allow the user to compute the probability density $p(x)$ at x for a exponential distribution with mean μ.
- $c = 3$:
  It will allow the user to compute exponential cdf $P(x)$
- $c = 4$:
  It will allow the user to compute exponential cdf $Q(x)$.
- $c = 5$:
  It will allow the user to compute inverse exponential cdf $P(x)$.
- $c \geq 6$:
  It will allow the user to compute inverse exponential cdf $Q(x)$.

c. Description

This function is used to calculate exponential random variate, probability density, cumulative distribution functions $P(x)$, $Q(x)$ and their inverse.
The distribution is,
$p(x)dx = \frac{1}{\mu}\exp(-x/\mu)dx$
where μ is the mean and $x \geq 0$.

Exponential Power Distribution (exppow)

Exponential Power Distribution

a. Syntax

$y = exppow(a, b, x, c)$

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

b. Parameters

y : output
a : scale parameter
b : exponent. For b=1 the distribution reduces to Laplace distribution and for b=2 the distribution reduces to gaussian distribution but with $a = 1.414s$. where 's' is standard deviation
x : x is real number lying within the range 0 to $+\infty$ for cdf and pdf computation.

- $c = 1$:
  It will allow the user to generate a exponential power random variate with scale parameter a and exponent b. In this case the function is independent of x and hence any value of x can be set.
- $c = 2$:
  It will allow the user to compute the probability density $p(x)$ at x for an exponential power distribution scale with parameter a and exponent b.
- $c = 3$:
  It will allow the user to compute exponential power cdf $P(x)$.
- $c \geq 4$:
  It will allow the user to compute exponential power cdf $Q(x)$.

c. Description

This function is used to calculate exponential power random variate, probability density, cumulative distribution functions $P(x)$, $Q(x)$.
The distribution is,
$p(x)dx = \frac{1}{2a\Gamma(1+1/b)} \exp(-|x/a|^b)dx$
for $x \geq 0$. For $b = 1$ this reduces to the Laplace distribution. For $b = 2$ it has the same form as a Gaussian distribution, but with $a = \sqrt{2}\sigma$.

F-Distribution (fdist)

The F-Distribution

a. Syntax

$y = fdist(n1, n2, x, c)$

b. Parameters

y : output
$n1, n2$: degrees of freedom (ν_1, ν_2)
x : x is real number lying within the range 0 to $+\infty$ for cdf and pdf computation. while for inverse cdf x will represent P and Q and its value will lie between 0 and 1.

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

- $c = 1$:
  It will allow the user to generate a F-Distributed random variate with degrees of freedom ν_1 and ν_2. In this case the function is independent of x and hence any value of x can be set.

- $c = 2$:
  It will allow the user to compute the probability density $p(x)$ at x for a F-Distribution with degrees of freedom ν_1 and ν_2.

- $c = 3$:
  It will allow the user to compute F cdf $P(x)$

- $c = 4$:
  It will allow the user to compute F cdf $Q(x)$.

- $c = 5$:
  It will allow the user to compute inverse F cdf $P(x)$.

- $c \geq 6$:
  It will allow the user to compute inverse F cdf $Q(x)$.

c. Description

This function is used to calculate F-Distributed random variate, probability density, cumulative distribution functions $P(x)$, $Q(x)$ and their inverse.
The distribution function is,
$p(x)dx = \frac{\Gamma((\nu_1+\nu_2)/2)}{\Gamma(\nu_1/2)\Gamma(\nu_2/2)}\nu_1^{\nu_1/2}\nu_2^{\nu_2/2}x^{\nu_1/2-1}(\nu_2+\nu_1 x)^{-\nu_1/2-\nu_2/2}$ where ν_1 and ν_2 represent degrees of freedom and $x \geq 0$.

Flat(Uniform) distribution (flatuniform)

Flat(Uniform) distribution

a. Syntax

$y = flatuniform(a, b, x, c)$

b. Parameters

y : output
a, b : 'a' and 'b' are lower and upper limit respectively.
x : x is a real number and lies in between 'a' and 'b' for cdf and pdf computation.

- $c = 1$:
  It will allow the user to generate a flat(uniform) random variate between the lower limit 'a' and the upper limit 'b'. In this case the function is independent of x and hence any value of x can be set.

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

- $c = 2$:
  It will allow the user to compute the probability density $p(x)$ at x for flat(uniform) distribution with lower limit 'a' and upper limit 'b'.
- $c = 3$:
  It will allow the user to compute flat(uniform) cdf $P(x)$.
- $c = 4$:
  It will allow the user to compute flat(uniform) cdf $Q(x)$.
- $c = 5$:
  It will allow the user to compute inverse flat(uniform) cdf $P(x)$.
- $c \geq 6$:
  It will allow the user to compute inverse flat(uniform) cdf $Q(x)$.

c. Description

This function is used to calculate flat(uniform) random variate, probability density, cumulative distribution functions $P(x)$, $Q(x)$ and their inverse.
The distribution is,
$p(x)dx = \frac{1}{(b-a)}dx$
if $a \leq x < b$ and 0 otherwise.

Gamma Distribution (gammadist)

Gamma Distribution or Erlang distribution

a. Syntax

$y = gammadist(a, b, x, c)$

b. Parameters

y : output
a, b : parameter of distribution. If 'a' is an integer then the distribution will be known as Erlang distribution.
x : x is a real number and lies in between 0 and $+\infty$ for cdf and pdf computation.

- $c = 1$:
  It will allow the user to generate a gamma random variate with parameter a and b. In this case the function is independent of x and hence any value of x can be set.
- $c = 2$:
  It will allow the user to compute the probability density $p(x)$ at x for gamma distribution with parameter a and exponent b.

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

- $c = 3$:
  It will allow the user to compute gamma cdf $P(x)$.

- $c = 4$:
  It will allow the user to compute gamma cdf $Q(x)$.

- $c = 5$:
  It will allow the user to compute inverse gamma cdf $P(x)$.

- $c \geq 6$:
  It will allow the user to compute inverse gamma cdf $Q(x)$.

c. Description

This function is used to calculate gamma random variate, probability density, cumulative distribution functions $P(x)$, $Q(x)$ and their inverse.
The distribution function is,
$p(x)dx = \frac{1}{\Gamma(a)b^a} x^{a-1} e^{-x/b} dx$
for $x > 0$.

Gaussian Distribution (gaussian)

Gaussian Distribution

a. Syntax

$y = gaussian(m, s, x, c)$

b. Parameters

y : output
m : mean (μ)
s : standard deviation (σ)
x : x is real number lying within the range $-\infty$ to $+\infty$ for cdf and pdf computation. while for inverse cdf x will represent P and Q and its value will lie between 0 and 1.

- $c = 1$:
  It will allow the user to generate a gaussian random variate with mean μ and standard deviation σ. In this case the function is independent of x and hence any value of x can be set.

- $c = 2$:
  It will allow the user to compute the probability density $p(x)$ at x for a gaussian distribution with mean μ and standard deviation σ.

- $c = 3$:
  It will allow the user to compute gaussian cdf $P(x)$

 rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

- $c = 4$:
  It will allow the user to compute gaussian cdf $Q(x)$.

- $c = 5$:
  It will allow the user to compute inverse gaussian cdf $P(x)$.

- $c \geq 6$:
  It will allow the user to compute inverse gaussian cdf $Q(x)$

c. Description

This function is used to calculate gaussian random variate, probability density, cumulative distribution functions $P(x)$, $Q(x)$ and their inverse.
The probability distribution for random variates is,
$p(x)dx = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-(x-\mu)^2/2\sigma^2)dx$
for x in the range $-\infty$ to $+\infty$.

Gaussian Tail Distribution (gaussiantail)

Gaussian Tail Distribution

a. Syntax

$y = gaussiantail(m, s, x, a, c)$

b. Parameters

y : output
m : mean (μ)
s : standard deviation (σ)
x : x is real number lying within the range $-\infty$ to $+\infty$ for pdf computation.
a : lower limit and it must be positive.

- $c = 1$:
  It will allow the user to generate a random variate from the upper tail of a Gaussian distribution with mean μ and standard deviation σ. The values returned are larger than the lower limit a. In this case the function is independent of x and hence any value of x can be set.

- $c \geq 2$:
  It will allow the user to compute the probability density $p(x)$ at x for a gaussiantail distribution with mean μ and standard deviation σ.

c. Description

This function is used to generate the gaussian random variate from the upper tail gaussian distribution and probability density $p(x)$ at x of a gaussian tail

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

distribution with mean μ and standard deviation σ.
The probability distribution for Gaussian tail random variates is,
$p(x)dx = \frac{1}{N(a;\sigma)\sqrt{2\pi\sigma^2}} \exp(-x^2/2\sigma^2)dx$
for $x > a$ where $N(a;\sigma)$ is the normalization constant,
$N(a;\sigma) = \frac{1}{2}\text{erfc}\left(\frac{a}{\sqrt{2\sigma^2}}\right)$.

Geometric distribution (geometric)

Geometric distribution

a. Syntax

$y = geometric(p, k, c)$

b. Parameters

y : output
p : probability of success.
k : k is an integer greater than equal to 1.

- $c = 1$:
  It will allow the user to generate a random integer from the geometric distribution, the number of independent trials with probability p until the first success.. In this case the function is independent of k and hence any value of k can be set.
- $c = 2$:
  It will allow the user to compute the probability $p(k)$ of obtaining k from a geometric distribution with parameter p.
- $c = 3$:
  It will allow the user to compute geometric cdf $P(k)$
- $c \geq 4$:
  It will allow the user to compute geometric cdf $Q(k)$

c. Description

This function generates geometric random variates, probability p(k) of obtaining k from a geometric distribution and cdf P(k), Q(k)
The probability distribution for geometric variates is,
$p(k) = p(1-p)^{k-1}$
for $k \geq 1$.

Hypergeometric Distribution (hypergeom)

Hypergeometric Distribution

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

a. Syntax

$y = hypergeom(n1, n2, t, k, c)$

b. Parameters

y : output
$n1$: number of elements of type-1 in the population
$n2$: number of elements of type-2 in the population
t : number of samples and $t \leq (n1 + n2)$
k : number of elements of type-1 in t samples. The domain of k is $max(0, t - n2)$....,$min(t, n1)$.

- $c = 1$:
  It will allow the user to generate a hypergeometric random variate. In this case the function is independent of k and hence any value of k can be set.
- $c = 2$:
  It will allow the user to compute the probability $p(k)$ of obtaining k from a hypergeometric distribution with parameters $n1$, $n2$, t.
- $c = 3$:
  It will allow the user to compute hypergeometric cdf $P(k)$
- $c \geq 4$:
  It will allow the user to compute hypergeometric cdf $Q(k)$

c. Description

This function is used to calculate hypergeometric random variate, probability $p(k)$ of obtaining k from a hypergeometric distribution and cumulative distribution functions $P(k)$, $Q(k)$.
The probability distribution for hypergeometric random variates is,
$p(k) = C(n_1, k)C(n_2, t - k)/C(n_1 + n_2, t)$
where $C(a, b) = a!/(b!(a-b)!)$ and $t \leq n_1 + n_2$. The domain of k is $\max(0, t - n_2)$, ..., $\min(t, n_1)$

Lognormal Distribution (lognormal)

Lognormal Distribution

a. Syntax

$y = lognormal(m, s, x, c)$

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

b. Parameters

y : output
m : mean (ζ)
s : standard deviation (σ)
x : x is real number greater than 0 for cdf and pdf computation. while for inverse cdf x will represent P and Q and its value will lie between 0 and 1.

- $c = 1$:
  It will allow the user to generate a lognormal random variate with mean ζ and standard deviation σ. In this case the function is independent of x and hence any value of x can be set.
- $c = 2$:
  It will allow the user to compute the probability density $p(x)$ at x for a lognormal distribution with mean ζ and standard deviation σ.
- $c = 3$:
  It will allow the user to compute lognormal cdf $P(x)$
- $c = 4$:
  It will allow the user to compute lognormal cdf $Q(x)$.
- $c = 5$:
  It will allow the user to compute inverse lognormal cdf $P(x)$.
- $c \geq 6$:
  It will allow the user to compute inverse lognormal cdf $Q(x)$.

c. Description

This function is used to calculate lognormal random variate, probability density, cumulative distribution functions $P(x)$, $Q(x)$ and their inverse.
The distribution function is,
$p(x)dx = \frac{1}{x\sqrt{2\pi\sigma^2}} \exp(-(\ln(x) - \zeta)^2/2\sigma^2)dx$ for $x > 0$. where σ is the standard deviation and ζ is the mean.

Logarithmic Distribution (logth)

Logarithmic Distribution

a. Syntax

$y = logth(p, k, c)$

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

b. Parameters

y : output
p : probability of success
k : k is an integer greater than equal to 1.

- $c = 1$:
  It will allow the user to generate a logarithmic random variate with probability of success p. In this case the function is independent of k and hence any value of k can be set.
- $c \geq 2$:
  It will allow the user to compute the probability $p(k)$ of obtaining k from a logarithmic distribution with probability of success p.

c. Description

This function is used to calculate logarithmic random variate and probability $p(k)$ of obtaining k from a logarithmic distribution with parameter p.
The probability distribution for logarithmic random variates is,
$p(k) = \frac{-1}{\log(1-p)}\left(\frac{p^k}{k}\right)$
for $k \geq 1$.

Negative Binomial Distribution (negbinomial)

Negative Binomial Distribution

a. Syntax

$y = negbinomial(p, n, k, c)$

b. Parameters

y : output
p : probability of success
n : number of successes in independent trials. It may be a positive real or integer number.
k : k is number of failures occurring before n successes in independent trials. It will be a positive integer.

- $c = 1$:
  It will generate random integer from the negative binomial distribution, the number of failures occurring before n successes in independent trials with probability p of success. In this case the function is independent of k and hence any value of k can be set.

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

- $c = 2$:
  It will allow the user to compute the probability $p(k)$ of obtaining k from a negative binomial distribution with parameters p and n.
- $c = 3$:
  It will allow the user to compute negative binomial cdf $P(k)$
- $c \geq 4$:
  It will allow the user to compute negative binomial cdf $Q(k)$

c. Description

This function is used to calculate negative binomial random variate, probability $p(k)$ of obtaining k from a negative binomial distribution and cumulative distribution functions $P(k)$, $Q(k)$.
The probability distribution for negative binomial variates is,
$p(k) = \frac{\Gamma(n+k)}{\Gamma(k+1)\Gamma(n)} p^n (1-p)^k$

Pascal Distribution (pascal)

Pascal Distribution. The Pascal distribution is simply a negative binomial distribution with an integer value of n.

a. Syntax

$y = pascal(p, n, k, c)$

b. Parameters

y : output
p : probability of success
n : number of successes in independent trials. It will be a positive integer number.
k : k is number of failures occurring before n successes in independent trials. It will be a positive integer.

- $c = 1$:
  It will generate random integer from the pascal distribution, the number of failures occurring before n successes in independent trials with probability p of success. In this case the function is independent of k and hence any value of k can be set.
- $c = 2$:
  It will allow the user to compute the probability $p(k)$ of obtaining k from a pascal distribution with parameters p and n.
- $c = 3$:
  It will allow the user to compute pascal cdf $P(k)$

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

- $c \geq 4$:
  It will allow the user to compute pascal cdf $Q(k)$

c. Description

This function is used to calculate pascal random variate, probability $p(k)$ of obtaining k from a pascal distribution and cumulative distribution functions $P(k)$, $Q(k)$.
The Pascal distribution is simply a negative binomial distribution with an integer value of n.
$p(k) = \frac{(n+k-1)!}{k!(n-1)!} p^n (1-p)^k$
for $k \geq 0$.

Poisson Distribution (poisson)

Poisson Distribution

a. Syntax

$y = poisson(m, k, c)$

b. Parameters

y : output
m : mean (μ)
k : k is positive integer greater than 0.

- $c = 1$:
  It will allow the user to generate a poisson random variate with mean μ. In this case the function is independent of k and hence any value of k can be set.
- $c = 2$:
  It will allow the user to compute the probability $p(k)$ of obtaining k from a poisson distribution with mean μ.
- $c = 3$:
  It will allow the user to compute poisson cdf $P(k)$
- $c \geq 4$:
  It will allow the user to compute poisson cdf $Q(k)$

c. Description

This function is used to calculate poisson random variate, probability $p(k)$ of obtaining k from a poisson distribution and cumulative distribution functions

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

$P(k)$, $Q(k)$
The probability distribution for Poisson variates is,
$p(k) = \frac{\mu^k}{k!} \exp(-\mu)$
where μ is mean and $k \geq 0$.

Student's t Distribution (tdist)

Student's t Distribution

a. Syntax

$y = tdist(n, x, c)$

b. Parameters

y : output
n : degrees of freedom (ν)
x : x is real number lying within the range $-\infty$ to $+\infty$ for cdf and pdf computation. while for inverse cdf x will represent P and Q and its value will lie between 0 and 1.

- $c = 1$:
  It will allow the user to generate an student's t random variate with ν degrees of freedom. In this case the function is independent of x and hence any value of x can be set.
- $c = 2$:
  It will allow the user to compute the probability density $p(x)$ at x for a student's t distribution with ν degrees of freedom.
- $c = 3$:
  It will allow the user to compute student's t cdf $P(x)$
- $c = 4$:
  It will allow the user to compute student's t cdf $Q(x)$.
- $c = 5$:
  It will allow the user to compute inverse student's t cdf $P(x)$.
- $c \geq 6$:
  It will allow the user to compute inverse student's t cdf Q(x).

c. Description

This function is used to calculate student's t random variate, probability density, cumulative distribution functions $P(x)$, $Q(x)$ and their inverse.
The t-distribution arises in statistics. If Y_1 has a normal distribution and Y_2 has a chi-squared distribution with ν degrees of freedom then the ratio,

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

$X = \frac{Y_1}{\sqrt{Y_2/\nu}}$
has a t-distribution $t(x;\nu)$ with ν degrees of freedom.

13 Linear System of Equations

Linear System solution (linsys)

Solution of linear system of equations $Ax = b$ using LU decomposition, QR decomposition and Householder solver.

a. Syntax

$y = linsys(A, b, c)$

b. Parameters

y : output Array containing the solution of linear system of equations.
b : Array of constant term present in the equations on RHS.
A : Array containing the coefficients of unknowns in the equations. The number of elements in A must be equal to the square of elements in array b. e.g. if number of elements in b is n then number of elements in A must be equal to n^2.

- $c = 1$:
  Solve system of linear equations using LU decomposition method.
  A general $M - by - N$ matrix A has an LU decomposition

  $PA = LU$

  where P is an $M - by - M$ permutation matrix, L is $M - by - \min(M, N)$ and U is $\min(M, N) - by - N$.

- $c = 2$:
  Solve system of linear equations using QR decomposition method.
  A general rectangular $M - by - N$ matrix A has a QR decomposition into the product of an orthogonal $M - by - M$ square matrix Q (where $Q^TQ = I$) and an $M - by - N$ right-triangular matrix R,

  $A = QR$

- $c \geq 3$:
  Solve system of linear equations using Householder solver method.

c. Description

This function is used to solve linear system of equations $Ax = b$ using LU decomposition, QR decomposition and Householder solver.

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

Least Squares solution (lsqod)

This function finds the least squares solution to the overdetermined system $Ay = b$ where the matrix A has more rows than columns or number of equations is more than the number of unknowns.

a. Syntax

$[y, res] = lsqod(A, b, c)$

b. Parameters

y : output Array containing the least squares solution to the overdetermined system.
res : residual error.
b : Array of constant terms present in the equations on RHS.
A : Array containing the coefficients of unknowns in the equations. The number of elements in A must be equal to the product of number of unknowns and number of elements in b.
c : Number of unknowns.

c. Description

This function finds the least squares solution to the overdetermined system $Ay = b$ where the matrix A has more rows than columns or number of equations is more than the number of unknowns. This function uses QR decomposition method to find the least squares solution.
The least squares solution minimizes the Euclidean norm of the residual, $||Ax - b||$.

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

14 Frequently Asked Questions

a. How to add the toolbox to Scilab after extracting it from the .zip file ?

Open Scilab. Click on File -> Browse for New. A new window will open. Look for the toolbox directory (sci_gsl), choose and click OK. Once the above step is done, execute the following codes in the Scilab console: exec builder.sce Click on 'Create Anyway' when the Popup box shows up. exec loader.sce During the execution of these codes, ensure that you don't encounter any errors.

b. How do I remove the toolbox?

Execute the following codes in the Scilab console after going to the sci_gsl toolbox directory (refer Q1).
exec unloader.sce
exec cleaner.sce
atomsRemove(["sci_gsl"])
ulink(0)
After running the above codes, restart Scilab.

c. Is the toolbox available for Windows/Mac OS?

No. At present, it works only in Ubuntu 18.04 and Ubuntu 20.04.

d. Where can I check for the details about the functions inside the toolbox?

You can either execute help in the Scilab console and view the details about the functions present in the toolbox or refer to the README file available inside the toolbox directory.

e. Can montecarlo() perform Monte Carlo Integration for any type of function?

No. There are 9 special functions defined inside montecarlo() for which the Monte Carlo Integration can be performed. Go to help in Scilab after installing the toolbox to view the list of functions available for Monte Carlo Integration. You can also refer to the README file available in the toolbox directory for further details.

f. Which version of Scilab has been used to test the toolbox?

Scilab 6.0.2

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com

***

rajangoyal68@gmail.com
drrashmivaida@gmail.com
kunderakshi@gmail.com
manjulaprakash2002@gmail.com