Pade Approximations
By
Namir C. Shammas

Contents

 Version 1.00.03

INTRODUCTION

Pade polynomials are polynomial ratios defined as:

$$y = P_{m,n}(x) = \frac{Q_m(x)}{D_n(x)} \quad (1)$$

Where $Q_m(x)$ is defined as:

$$Q_m(x) = a_0 + \sum_{i=1}^{m} a_i\, x^i \quad (2)$$

And $D_n(x)$ is defined as:

$$D_n(x) = 1 + \sum_{i=1}^{n} b_i\, x^i \quad (3)$$

The Pade polynomials have more flexibility that ordinary legacy polynomials. The trick is to find the optimum values for m and n. This study uses swarm optimization to determine these optimum values within the range of (2, 7). The multiple regression model used to fit Pade polynomials is:

$$y = = a_0 + \sum_{i=1}^{m} a_i\, x^i - \sum_{i=1}^{n} b_i\, y\, x^i \quad (4)$$

When you use the Pade polynomials to calculate new values, use the following form:

$$y = = [a_0 + \sum_{i=1}^{m} a_i\, x^i] \,/ [1 - \sum_{i=1}^{n} b_i\, y\, x^i] \quad (5)$$

Unlike the Shammas polynomial fitting, working with Pade polynomials focusing on simply determining the optimum values of the orders m and n for $Q_m(x)$ and $D_n(x)$, respectively. As these two pollynomials are regular polynomials there are no Shammas polynomial parameters to calculate the powers for the various terms. Thus, the process of approximating various functions using Pade polynomials becomes much simpler. Another advantage for this simplicity is that this report is much shorter than the one for Shammas polynomials approximations—there is one approximation per function.

This study looks at Pade polynomials used to approximate common functions that include:

- Trigonometric functions and their inverses.
- Hyperbolic functions.
- Logarithmic functions.
- Exponential functions.
- Inverse student-t functions.
- The common logarithm of the gamma function.
- Tthe digamma function.
- The trigamma function.

The digamma function is defined as:

$$\psi(x) = \Gamma'(x) \;/\; \Gamma(x) = \frac{d \ln(\Gamma(x))}{dx} \tag{6}$$

The following Matlab function implements the code for the digamma function:

```
function y = digamma(x)
%DIGAMMA Summary of this function goes here
%   Detailed explanation goes here
  h = 0.001;
  fp = gammaln(x+h);
  fm = gammaln(x-h);
  y = (fp -fm)/2/h;
end
```

The above implementation of the digamma function was suggested by Albert Chan, a member of the hp museum web site, in a post he wrote on that site. The above code gives slightly more accurate results than the expression (gamma(x+h)-gamma(x-h)/(2*h)/gamma(x).

The trigamma function is defined as:

$$\psi_1(x) = \frac{d^2}{dx^2} \ln(\Gamma(x)) \tag{7}$$

The following Matlab function implements the code for the trigamma function:

```
function y = trigamma(x)
%DIGAMMA Summary of this function goes here
%   Detailed explanation goes here
  h = 0.001;
  fp = log(gamma(x+h));
  fm = log(gamma(x-h));
  f0 = log(gamma(x));
  y = (fp -2*f0 + fm)/h/h;
end
```

The approximations that I obtain are typically for a defined and suitable interval. It is your responsibility to implement expanded versions of the approximation functions that take wider ranges of arguments and map them onto the interval used. For example, given that my approximation for $\log_{10}(x)$ uses the range (1, 10), to calculate $\log_{10}(235)$ use:

$\log_{10}(235) =$

$\log_{10}(2.35 * 100) =$

$\log_{10}(2.35) + \log_{10}(100) =$

$\log_{10}(2.35) + 2$

The argument of the $\log_{10}(x)$ function in the last line falls in the interval (1, 10).

MATLAB Code

The algorithm in this study uses particle swarm optimization to obtain the best orders for polynomials $Q_m(x)$ and $D_n(x)$ fitting a set of (x, y) data.

This study uses the function PadePoly () to perform various Pade polynomial curve fitting:

```
function PadePoly(fx,xRange,Lb,Ub,runNum,sFxName,diaryFilename)
%SHAMPOLY2 Summary of this function goes here
%   Detailed explanation goes here
clc
global bDeleteIfExists
global bUseDiary
global xdata
global ydata
global orderA
global orderB
warning('off','all')
if isempty(sFxName)
  sFxName = getFuncName(fx);
end
xdata = xRange';
ydata = xdata;
for i=1:length(xdata)
  ydata(i)=fx(xdata(i));
end
xmin = min(xdata);
xmax = max(xdata);
ymin = min(ydata);
ymax = max(ydata);
xdata = (xdata - xmin)/(xmax - xmin) + 1;
ydata = (ydata - ymin)/(ymax - ymin) + 1;
```



```
fprintf('Fitting %s in range (%f, %f)\n', sFxName, xmin ,xmax);

options = optimoptions('particleswarm', 'Display', 'iter');
[x,psAICc] = particleswarm(@optimFunc,2,Lb,Ub,options);
orderA = round(x(1));
orderB = round(x(2));
if bUseDiary
  diaryFilename = strrep(diaryFilename, ".txt", strcat("_",
num2str(orderA),"x",num2str(orderB),"_run", num2str(runNum),".txt"));
  if exist(diaryFilename, 'file')==2
    if bDeleteIfExists
      delete(diaryFilename);
    else
      return;
    end
  end
end
X = [];
for i=1:orderA
  xs = xdata.^i;
  X = [X;xs'];
end
for i=1:orderB
  xs = ydata.*xdata.^i;
  X = [X;xs'];
end
X = X';
lm = fitlm(X,ydata);
if bUseDiary
  diary(diaryFilename)
end
fprintf('Fitting %s in range (%f, %f)\n', sFxName, xmin ,xmax);
format long
disp(lm);
anva = anova(lm,'summary');
disp(anva);
format short
fprintf('orderA = %f, orderB = %f\n', orderA, orderB);
fprintf("Xmin = %f and Xmax = %f\n", xmin, xmax);
fprintf("Ymin = %f and Ymax = %f\n", ymin, ymax);
sMdl1 = "y = [Intercept";
j = 1;
for i=1:orderA
  if i<2
    sMdl1 = strcat(sMdl1, " + cx", num2str(j),"*x");
  else
    sMdl1 = strcat(sMdl1, " + cx", num2str(j),"*x^",num2str(i));
  end
  j = j + 1;
end
sMdl1 = strcat(sMdl1, "] /");
sMdl2 = "[ 1";
for i=1:orderB
  if i<2
    sMdl2 = strcat(sMdl2, " + cx", num2str(j),"*y*x");
  else
    sMdl2 = strcat(sMdl2, " + cx", num2str(j),"*y*x^",num2str(i));
```

```
  end
  j = j + 1;
end
sMdl2 = strcat(sMdl2, "]");
fprintf("Model is %s\n\t\t%s\n", sMdl1, sMdl2);
fprintf('Fitting %s in range (%f, %f)\n', sFxName, xmin ,xmax);
n = length(xdata);
sumsqr = 0;
for i=1:n
  yc = lm.Coefficients{1,1};
  for j=1:orderA
    yc = yc + lm.Coefficients{j+1,1} * xdata(i)^j;
  end
  for j=1:orderB
    yc = yc + lm.Coefficients{j+orderA+1,1} * ydata(i)*xdata(i)^j;
  end
  sumsqr = sumsqr + (ydata(i) - yc)^2;
end

k = orderA + orderB + 1;
fprintf('MSS of errors squared = %e\n', sqrt(sumsqr)/n);
AIC = lm.ModelCriterion.AIC;
AICc = AIC + 2*k*(1 + (k+1)/(n-k-1));
fprintf('Particle swarm AICc =  %e\n', psAICc);
fprintf('AIC =  %e\n', AIC);
fprintf('AICc =  %e\n', AICc);

if bUseDiary
  diary off
end
end

function AICc = optimFunc(x)
  global xdata
  global ydata
  global orderA
  global orderB

  orderA = round(x(1));
  orderB = round(x(2));
  X = [];
  for i=1:orderA
    xs = xdata.^i;
    X = [X;xs'];
  end
  for i=1:orderB
    xs = ydata.*xdata.^i;
    X = [X;xs'];
  end
  X = X';
  lm = fitlm(X,ydata);

  n = length(xdata);
  k = orderA + orderB + 1;
  AIC = lm.ModelCriterion.AIC;
  AICc = AIC + 2*k*(1 + (k+1)/(n-k-1));
  if isinf(AICc), AICc = -1e+99; end
```

```
end

function sFx = getFuncName(fx)
  sFx = func2str(fx);
  if sFx(1:2)=="@("
    i = strfind(sFx,")");
    sFx = sFx(i(1)+1:end);
  elseif sFx(1)=="@"
    sFx = strcat(sFx(2:end),".m");
  else
    % return sFx as is
  end
end
```

The parameters of function PadePoly () are:

- The parameter fx is the handle (or inline function) for the function being approximated. An example is @(x)cos(x) which also shows the ***recommended format*** for the argument of parameter fx.
- The parameter xRange is the array that specifies the minimum value, increment value, and maximum value for the range of approximation.
- The parameter Lb is is the array of lower limits for the orders of polynomials $Q_m(x)$ and $D_n(x)$. An example is [2 2].
- The parameter Ub is is the array of upper limits for the orders of polynomials Q_m (x) and $D_n(x)$. An example is [7 7].
- The parameter runNum specifies the run number. The arguments for this parameter have nothing to do with the calculations and serve in fine tuning the name of the diary files, when used.
- The optional parameter sFxName is the name of the approximated function. An example is “cos(x)”.
- The parameter diaryFilename is the name of the diary file. An example is “cos_1.txt”.

The above listing performs the following tasks:

1. Initialize the data for the curve fitting. The function uses the global variables xdata and ydata to store the data for the polynomial fitting.
2. Normalizes the values for xdata and ydata to fall in the range (1, 2). The function stores the minimum and maximum values for the two variables.
3. Set the optimization options and then call the Matlab function particleswarm(). The function call returns the optimized values of orderA and orderB and the *corrected* Akaike information criterion (AICc). The arguments for this function call are:

   a. The handle to the local function optimFunc() that calculates the root mean sum of errors squared.
   b. The number of optimized variables which is 2.
   c. The lower and upper bounds arrays, Lb and Ub, respectively,
   d. The optimization parameters for function particleswarm().
4. Retrieve the optimum values and perform a Pade polynomial.
5. Display the results of the regression and its associated ANOVA table.
6. Display the Pade polynomial powers.
7. Display the minimum and maximum values for xdata and ydata.
8. Display the model of $Q_m(x)/D_n(x)$ in terms of the regression coefficients. This output appens a c to the names of the generic variables listed. For example, cx1 refers to the regression coefficient associated with the generic variable x1.
9. Display the range of the approximated function.
10. Calculate and display the value of the mean square root of the sum of squared errors. This statistic serves as a check that the Pade polynomial performs well in checking the training data.
11. Calculate and display the *corrected* Akaike information criterion. This statistic is calculated using:

AIC = n * ln(SSE/n) + 2*k (8)

AICc = AIC + 2*k*(k+1)/(n-k-1) (9)

Where n is the number of observations, k is the total number of regression coefficients (including the intercept), and SSE is the sum of squared errors. The program obtains the value of AIC using lm.ModelCriterion.AIC. The program uses equation (3) to calculate the value for AICc.

12. Close the diary file, if one is used.

The function optimFunc() obtains the array x containing the current values of A and B, and the best Pade polynomial order. The function calculates the transformed variables needed to perform a curve fit for a Pade polynomial. This task calls the Matlab function fitlm(). The optimFunc() function returns the AICc as its result. I am using this statistic since the optimization is dealing with different Pade polynomial orders and thus a varying number of polynomial coefficients.

The function getFuncName() returns a string-type function name given a handle of a function. The best way to take advantage of this function is to supply arguments like @(x)cos(x). Such arguments allow the function to discard the part that declares the variable(s) and return the part that comes after the first closed parenthesis (e.g. cos(x)). If you supply an argument like @fx1 which refers to the file fx1.m that defines the function fx1() then the function getFuncName() returns *fx1.m*. This string value indicates that you a rereferencing a separate Matlab file that implements the code for fx1.

Under the current calculation scheme, note that, the functionPade () does not explicitly iterate over different values of orders of polynomials $Q_m(x)$ and $D_n(x)$. It delegates that task to the Matlab function particleswarm().

The following Matlab script goAll () performs the various Pade polynomial fittings for the various tested functions:

```
% Version 1.0.0 8/14/2020
global bUseDiary
global bDeleteIfExists

bUseDiary = true;
bDeleteIfExists = false; % or false
runNum = 1;
bShutdown = false;

tic;

lstA = [2 7];
lstB = [2 7];
Lb = [lstA(1) lstB(1)];
Ub = [lstA(2) lstB(2)];

PadePoly(@(x)acos(x),[0:.01:1],Lb,Ub,runNum,"arccos(x)","arccos.txt")
PadePoly(@(x)asin(x),[0:.01:1],Lb,Ub,runNum,"arcsin(x)","arcsin.txt")
PadePoly(@(x)atan(x),[0:.01:1],Lb,Ub,runNum,"arctan(x)","arctan.txt")
PadePoly(@(x)sin(x),[0:.01:1],Lb,Ub,runNum,"sin(x)","sin.txt")
PadePoly(@(x)cos(x),[0:.01:1],Lb,Ub,runNum,"cos(x)","cos.txt")
PadePoly(@(x)tan(x),[0:.01:1],Lb,Ub,runNum,"tan(x)","tan.txt")
PadePoly(@(x)sinh(x),[0:.01:5],Lb,Ub,runNum,"sinh(x)","sinh.txt")
PadePoly(@(x)cosh(x),[0:.01:5],Lb,Ub,runNum,"cosh(x)","cosh.txt")
PadePoly(@(x)tanh(x),[0:.01:3],Lb,Ub,runNum,"tanh(x)","tanh.txt")
PadePoly(@(x)erf(x),[0:.01:2.1],Lb,Ub,runNum,"erf(x)","erf.txt")
PadePoly(@(x)exp(x),[0:.01:2],Lb,Ub,runNum,"exp(x)","exp.txt")
PadePoly(@(x)log(x),[1:.01:10],Lb,Ub,runNum,"ln(x)","ln.txt")
PadePoly(@(x)log10(x),[1:.01:10],Lb,Ub,runNum,"log(x)","log.txt")
PadePoly(@(x)10.^x,[0:.01:1],Lb,Ub,runNum,"10^x","pwr10.txt")
PadePoly(@(x)tinv(0.95,x),[2:100],Lb,Ub,runNum,"tinv(0.95,x)","tinv1.txt")
PadePoly(@(x)tinv(0.975,x),[2:100],Lb,Ub,runNum,"tinv(0.975,x)","tinv2.txt")
PadePoly(@(x)log10(gamma(x)),[2:100],Lb,Ub,runNum,"log10Gamma(x)","log10Gamma
.txt")
PadePoly(@(x)digamma(x),[1:100],Lb,Ub,runNum,"digamma(x)","digamma.txt")
PadePoly(@(x)trigamma(x),[1:100],Lb,Ub,runNum,"tigamma(x)","trigamma.txt")

toc;
```



```
for i=1:7
  beep;
  pause(3)
end

if bShutdown
  system('shutdown -s');
else
  fprintf("\n\nDone!\n\n");
end
```

The above listing has the following global and operational variabls:

- The global variable bUseDiary is a Boolean flag used to tell the function PadePoly() if you want to copy the screen output to diary text files.
- The global variable bDeleteIfExists is a Boolean flag used to tell the function PadePoly() whether you want to delete diary files if they exist.
- The Boolean variable bShutdown tells the Matlab script whether to shut down the computer when done

You can edit the script in goAll.m to perform Pade polynomial fitting for other functions you are interested in or for other Pade polynomial orders.

General Comments on Results

The next sections show the results of fitting various common functions each with a variety of Pade polynomials. There is a summary table for the fitted functions showing the Pade polynomial orders, the F statistic, the AICc statistic, the root mean sum of errors squared.

Remember that the function PadePoly() performs data transformation on the x and y values, mapping them both in the range of (1, 2). When using any of the Pade approximations shown below, perform the following:

1. Transform you x value(s) using the xmin and xmax values associated with the function approximation you are using and employing (x – xmin)/(xmax – xmin) + 1.
2. Calculate the y value using the Pade polynomial.
3. Revere map the calculate y value, using the ymin and ymax values, and applying the expression (y – 1)*(ymax – ymin) + ymin.

Results for the Arc Cosine

```
Fitting arccos(x) in range (0.000000, 1.000000)

Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10

Estimated Coefficients:
                         Estimate          SE    tStat    pValue
                   ____________________    __    _____    ______

    (Intercept)        2.74411876998652    0      Inf       0
    x1                -4.75440799756697    0     -Inf       0
    x2                 3.43983951559669    0      Inf       0
    x3                -1.54894562520587    0     -Inf       0
    x4                0.575961766772712    0      Inf       0
    x5                -0.17299703478946    0     -Inf       0
    x6               0.0319923903691364    0      Inf       0
    x7             -0.00257872204993489    0     -Inf       0
    x8                 1.37783118927832    0      Inf       0
    x9               -0.629725985821115    0     -Inf       0
    x10              0.0954032517914815    0      Inf       0


Number of observations: 101, Error degrees of freedom: 90
R-squared: 1,  Adjusted R-Squared: 1
F-statistic vs. constant model: Inf, p-value = 0
                     SumSq          DF          MeanSq           F     pValue
                ________________    ___    __________________    ___    ______

    Total       6.03050467942528    100    0.0603050467942528
    Model       6.03050467942528     10     0.603050467942528    Inf       0
    Residual                   0     90                     0

orderA = 7.000000, orderB = 3.000000
Xmin = 0.000000 and Xmax = 1.000000
Ymin = 0.000000 and Ymax = 1.570796
Model is y = [Intercept + cx1*x + cx2*x^2 + cx3*x^3 + cx4*x^4 + cx5*x^5 + cx6*x^6 + cx7*x^7] /
                 [ 1 + cx8*y*x + cx9*y*x^2 + cx10*y*x^3]
Fitting arccos(x) in range (0.000000, 1.000000)
MSS of errors squared = 9.165132e-10
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

Results for the Arc Sine

```
Fitting arcsin(x) in range (0.000000, 1.000000)

Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11

Estimated Coefficients:
                        Estimate          SE    tStat    pValue
                   __________________    __    _____    ______

    (Intercept)     0.234940453381004    0      Inf       0
    x1               0.74104922101633    0      Inf       0
    x2              -2.25026340524683    0     -Inf       0
    x3               1.94299333200437    0      Inf       0
```

x4	-0.699592168359401	0	-Inf	0
x5	0.091672916443102	0	Inf	0
x6	1.48857616520442	0	Inf	0
x7	-0.150354831868603	0	-Inf	0
x8	-0.831112075749417	0	-Inf	0
x9	0.556941620367831	0	Inf	0
x10	-0.136018141698462	0	-Inf	0
x11	0.0111669207831689	0	Inf	0

```
Number of observations: 101, Error degrees of freedom: 89
R-squared: 1,  Adjusted R-Squared: 1
F-statistic vs. constant model: Inf, p-value = 0
```

	SumSq	DF	MeanSq	F	pValue
Total	6.03050467942538	100	0.0603050467942538		
Model	6.03050467942538	11	0.54822769812958	Inf	0
Residual	0	89	0		

```
orderA = 5.000000, orderB = 6.000000
Xmin = 0.000000 and Xmax = 1.000000
Ymin = 0.000000 and Ymax = 1.570796
Model is y = [Intercept + cx1*x + cx2*x^2 + cx3*x^3 + cx4*x^4 + cx5*x^5] /
                 [ 1 + cx6*y*x + cx7*y*x^2 + cx8*y*x^3 + cx9*y*x^4 + cx10*y*x^5 + cx11*y*x^6]
Fitting arcsin(x) in range (0.000000, 1.000000)
MSS of errors squared = 1.936336e-10
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

Results for the Arc Tangent

```
Fitting arctan(x) in range (0.000000, 1.000000)

Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11

Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
(Intercept)	-0.0135261932408763	0	-Inf	0
x1	0.732530003859316	0	Inf	0
x2	-0.586835180957421	0	-Inf	0
x3	0.797757928337211	0	Inf	0
x4	-0.393396378726248	0	-Inf	0
x5	0.248522085670495	0	Inf	0
x6	0.964967258572369	0	Inf	0
x7	-0.810042964004574	0	-Inf	0
x8	0.137378866156981	0	Inf	0
x9	-0.00917906911705593	0	-Inf	0
x10	-0.0671577417687822	0	-Inf	0
x11	-0.00101861480487907	0	-Inf	0

```
Number of observations: 101, Error degrees of freedom: 89
R-squared: 1,  Adjusted R-Squared: 1
F-statistic vs. constant model: Inf, p-value = 0
```

	SumSq	DF	MeanSq	F	pValue

```
    Total       8.80077198802885    100    0.0880077198802885
    Model       8.80077198802885     11     0.800070180729895    Inf       0
    Residual                   0     89                     0

orderA = 5.000000, orderB = 6.000000
Xmin = 0.000000 and Xmax = 1.000000
Ymin = 0.000000 and Ymax = 0.785398
Model is y = [Intercept + cx1*x + cx2*x^2 + cx3*x^3 + cx4*x^4 + cx5*x^5] /
                 [ 1 + cx6*y*x + cx7*y*x^2 + cx8*y*x^3 + cx9*y*x^4 + cx10*y*x^5 + cx11*y*x^6]
Fitting arctan(x) in range (0.000000, 1.000000)
MSS of errors squared = 1.083466e-12
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

Results for Cosine

```
Fitting cos(x) in range (0.000000, 1.000000)

Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7

Estimated Coefficients:
                           Estimate          SE    tStat    pValue
                   ____________________    __    _____    ______

    (Intercept)        0.99744263261147    0      Inf       0
    x1                 1.53975473802235    0      Inf       0
    x2                -1.06223383917815    0     -Inf       0
    x3                0.113655014741225    0      Inf       0
    x4                0.304952088808806    0      Inf       0
    x5               -0.117680100338148    0     -Inf       0
    x6               0.0213925509106876    0      Inf       0
    x7             -0.00297381627419662    0     -Inf       0


Number of observations: 101, Error degrees of freedom: 93
R-squared: 1,  Adjusted R-Squared: 1
F-statistic vs. constant model: Inf, p-value = 0
                     SumSq          DF          MeanSq           F     pValue
                ________________    ___    __________________    ___    ______

    Total       9.38559236862651    100    0.0938559236862651
    Model       9.38559236862651      7      1.34079890980379    Inf       0
    Residual                   0     93                     0

orderA = 3.000000, orderB = 4.000000
Xmin = 0.000000 and Xmax = 1.000000
Ymin = 0.540302 and Ymax = 1.000000
Model is y = [Intercept + cx1*x + cx2*x^2 + cx3*x^3] /
                 [ 1 + cx4*y*x + cx5*y*x^2 + cx6*y*x^3 + cx7*y*x^4]
Fitting cos(x) in range (0.000000, 1.000000)
MSS of errors squared = 2.504558e-10
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

Results for the Hyperbolic Cosine Function


```
Fitting cosh(x) in range (0.000000, 5.000000)

Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9

Estimated Coefficients:
                        Estimate         SE    tStat    pValue
                   __________________    __    _____    ______

    (Intercept)      1.55934916282121    0      Inf       0
    x1              -3.10369563975385    0     -Inf       0
    x2               4.66652693020443    0      Inf       0
    x3              -4.87494386825909    0     -Inf       0
    x4               3.21698131564907    0      Inf       0
    x5              -1.30015037085423    0     -Inf       0
    x6               0.29725294182685    0      Inf       0
    x7             -0.0288874680927875   0     -Inf       0
    x8              0.688132938573364    0      Inf       0
    x9             -0.120566000484616    0     -Inf       0


Number of observations: 501, Error degrees of freedom: 491
R-squared: 1,  Adjusted R-Squared: 1
F-statistic vs. constant model: Inf, p-value = 0
                     SumSq         DF          MeanSq          F     pValue
                ______________    ___    _________________    ___    ______

    Total       31.218732615625   500    0.0624374652312499
    Model       31.218732615625     9      3.46874806840277   Inf       0
    Residual                  0   491                     0

orderA = 7.000000, orderB = 2.000000
Xmin = 0.000000 and Xmax = 5.000000
Ymin = 1.000000 and Ymax = 74.209949
Model is y = [Intercept + cx1*x + cx2*x^2 + cx3*x^3 + cx4*x^4 + cx5*x^5 + cx6*x^6 + cx7*x^7] /
                 [ 1 + cx8*y*x + cx9*y*x^2]
Fitting cosh(x) in range (0.000000, 5.000000)
MSS of errors squared = 5.717733e-10
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

RESULTS FOR THE ERROR FUNCTION

```
Fitting erf(x) in range (0.000000, 1.000000)

Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11

Estimated Coefficients:
                        Estimate                SE                   tStat                  pValue
                   __________________    __________________    __________________    ____________________

    (Intercept)    0.0634927794926562     0.163551992466092     0.388211592749748       0.698274432642597
    x1             -0.206186311473873     0.954407294219535    -0.216035976173549       0.829180986134864
    x2              0.165176348223269      2.31652059597425      0.07130363896195       0.943227707138312
    x3                0.8791079054853      2.98952919533215     0.294062324883108       0.769016767974869
    x4              -1.22188727848586      2.13725789505277     -0.57170792598976       0.568165050489232
    x5              0.652873854026182     0.865274673274095     0.754527867498752       0.451424283164583
    x6             -0.149260114136752     0.183761788372226    -0.812247831602575       0.417620013842709
    x7             0.0140281514647448    0.0161279336076203     0.869804638711845        0.38545511140998
```


x8	1.96685368256505	0.0452586936750921	43.4580303330208	1.60125995554417e-103
x9	-1.82884980961066	0.0379019688152698	-48.2521058081251	8.74396904566243e-112
x10	0.839678341953981	0.0204449658452734	41.0701758226758	3.98958209220962e-99
x11	-0.175027545943141	0.00915353159077918	-19.1213133649372	6.03867144018144e-47

Number of observations: 211, Error degrees of freedom: 199
Root Mean Squared Error: 4.78e-08
R-squared: 1, Adjusted R-Squared: 1
F-statistic vs. constant model: 7.29e+14, p-value = 0

	SumSq	DF	MeanSq	F	pValue
Total	18.3371112829991	210	0.0873195775380909		
Model	18.3371112829986	11	1.66701011663624	729493008730985	0
Residual	4.54747350886464e-13	199	2.28516256726866e-15		

```
orderA = 7.000000, orderB = 4.000000
Xmin = 0.000000 and Xmax = 2.100000
Ymin = 0.000000 and Ymax = 0.997021
Model is y = [Intercept + cx1*x + cx2*x^2 + cx3*x^3 + cx4*x^4 + cx5*x^5 + cx6*x^6 + cx7*x^7] /
                 [ 1 + cx8*y*x + cx9*y*x^2 + cx10*y*x^3 + cx11*y*x^4]
Fitting erf(x) in range (0.000000, 1.000000)
MSS of errors squared = 7.212629e-11
Particle swarm AICc =  -6.688183e+03
AIC =  -6.713759e+03
AICc =  -6.688183e+03
```

RESULTS FOR THE EXPONENTIAL FUNCTION

Fitting exp(x) in range (0.000000, 2.000000)

Linear regression model:
y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	0.864659858935923	0	Inf	0
x1	-0.66662892088373	0	-Inf	0
x2	0.224882971306153	0	Inf	0
x3	-0.020997224679647	0	-Inf	0
x4	0.820002283472213	0	Inf	0
x5	-0.251379087975406	0	-Inf	0
x6	0.0292917287174262	0	Inf	0
x7	0.000203802681796752	0	Inf	0
x8	6.61006958148164e-05	0	Inf	0
x9	-0.000114334503638729	0	-Inf	0
x10	1.28222327971386e-05	0	Inf	0

Number of observations: 201, Error degrees of freedom: 190
R-squared: 1, Adjusted R-Squared: 1
F-statistic vs. constant model: Inf, p-value = 0

	SumSq	DF	MeanSq	F	pValue
Total	15.9274939123249	200	0.0796374695616245		
Model	15.9274939123249	10	1.59274939123249	Inf	0
Residual	0	190	0		


```
orderA = 3.000000, orderB = 7.000000
Xmin = 0.000000 and Xmax = 2.000000
Ymin = 1.000000 and Ymax = 7.389056
Model is y = [Intercept + cx1*x + cx2*x^2 + cx3*x^3] /
                 [ 1 + cx4*y*x + cx5*y*x^2 + cx6*y*x^3 + cx7*y*x^4 + cx8*y*x^5 + cx9*y*x^6 + cx10*y*x^7]
Fitting exp(x) in range (0.000000, 2.000000)
MSS of errors squared = 5.899982e-15
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

RESULTS FOR THE NATURAL LOGRITHM

```
Fitting ln(x) in range (1.000000, 10.000000)

Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11 + x12

Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
(Intercept)	2.12103478819	0	Inf	0
x1	-7.32802068756791	0	-Inf	0
x2	0	0	NaN	NaN
x3	24.8126032129289	0	Inf	0
x4	-29.5997694083641	0	-Inf	0
x5	8.19029041078807	0	Inf	0
x6	1.82492367231733	0	Inf	0
x7	5.14829823747902	0	Inf	0
x8	-6.73819013764748	0	-Inf	0
x9	-2.69042586262795	0	-Inf	0
x10	9.9311271484996	0	Inf	0
x11	-4.17669832038885	0	-Inf	0
x12	-0.49517305508618	0	-Inf	0

```
Number of observations: 901, Error degrees of freedom: 889
R-squared: 1,  Adjusted R-Squared: 1
F-statistic vs. constant model: Inf, p-value = 0
```

	SumSq	DF	MeanSq	F	pValue
Total	58.921158778179	900	0.0654679541979767		
Model	58.921158778179	11	5.35646897983446	Inf	0
Residual	0	889	0		

```
orderA = 6.000000, orderB = 6.000000
Xmin = 1.000000 and Xmax = 10.000000
Ymin = 0.000000 and Ymax = 2.302585
Model is y = [Intercept + cx1*x + cx2*x^2 + cx3*x^3 + cx4*x^4 + cx5*x^5 + cx6*x^6] /
                 [ 1 + cx7*y*x + cx8*y*x^2 + cx9*y*x^3 + cx10*y*x^4 + cx11*y*x^5 + cx12*y*x^6]
Fitting ln(x) in range (1.000000, 10.000000)
MSS of errors squared = 8.896563e-12
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

Results for the Common Logarithm

```
Fitting log(x) in range (1.000000, 10.000000)

Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10

Estimated Coefficients:
                        Estimate         SE    tStat    pValue
                   __________________    __    _____    ______

    (Intercept)      1.45540511589676    0      Inf       0   
    x1             -0.513439118636382    0     -Inf       0   
    x2              -13.3013394674701    0     -Inf       0   
    x3               22.3461715375533    0      Inf       0   
    x4              -8.05269520917601    0     -Inf       0   
    x5              -2.07766207174536    0     -Inf       0   
    x6               2.87893865791043    0      Inf       0   
    x7              0.915541424555934    0      Inf       0   
    x8               -7.4356545028703    0     -Inf       0   
    x9                4.2111499152529    0      Inf       0   
    x10             0.573583704223171    0      Inf       0   


Number of observations: 901, Error degrees of freedom: 890
R-squared: 1,  Adjusted R-Squared: 1
F-statistic vs. constant model: Inf, p-value = 0
                     SumSq          DF          MeanSq           F     pValue
                ________________    ___    __________________    ___    ______

    Total       58.9211587781763    900    0.0654679541979736                 
    Model       58.9211587781763     10      5.89211587781763    Inf       0  
    Residual                   0    890                     0                 

orderA = 5.000000, orderB = 5.000000
Xmin = 1.000000 and Xmax = 10.000000
Ymin = 0.000000 and Ymax = 1.000000
Model is y = [Intercept + cx1*x + cx2*x^2 + cx3*x^3 + cx4*x^4 + cx5*x^5] /
                 [ 1 + cx6*y*x + cx7*y*x^2 + cx8*y*x^3 + cx9*y*x^4 + cx10*y*x^5]
Fitting log(x) in range (1.000000, 10.000000)
MSS of errors squared = 7.292823e-11
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

Results for the Common Exponent (10^x)

```
Fitting 10^(x) in range (0.000000, 1.000000)

Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7

Estimated Coefficients:
                        Estimate          SE    tStat    pValue
                   ___________________    __    _____    ______

    (Intercept)      0.900308834713545    0      Inf       0   
    x1              -0.613792590815386    0     -Inf       0   
    x2               0.175326759951948    0      Inf       0   
    x3             -0.0110643895286522    0     -Inf       0   
```

```
    x4             0.00339167046058943    0      Inf       0
    x5               0.708790850944777    0      Inf       0
    x6              -0.179054051949656    0     -Inf       0
    x7              0.0160929194398677    0      Inf       0


Number of observations: 101, Error degrees of freedom: 93
R-squared: 1,  Adjusted R-Squared: 1
F-statistic vs. constant model: Inf, p-value = 0
                     SumSq          DF          MeanSq           F     pValue
                ________________    ___    __________________    ___    ______

    Total       7.96302009058562    100    0.0796302009058562
    Model       7.96302009058562      7      1.13757429865509    Inf       0
    Residual                   0     93                     0

orderA = 4.000000, orderB = 3.000000
Xmin = 0.000000 and Xmax = 1.000000
Ymin = 1.000000 and Ymax = 10.000000
Model is y = [Intercept + cx1*x + cx2*x^2 + cx3*x^3 + cx4*x^4] /
                 [ 1 + cx5*y*x + cx6*y*x^2 + cx7*y*x^3]
Fitting 10^(x) in range (0.000000, 1.000000)
MSS of errors squared = 1.120822e-10
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

RESULTS FOR THE SINE FUNCTION

```
Fitting sin(x) in range (0.000000, 1.000000)

Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8

Estimated Coefficients:
                         Estimate                 SE                  tStat                 pValue
                   _____________________    ___________________    ___________________    ____________________

    (Intercept)    -5.68503965954338e-05     0.0016865508408706    -0.0337080835144509       0.973182958306001
    x1                 0.642453006092019    0.00935439914415206        68.679238098756    7.64477851861237e-81
    x2                 0.448766635814435      0.147771268161347       3.03690048409437      0.0031096796874584
    x3                -0.131145401196643     0.0918230027193288      -1.42824126104337       0.156608517160931
    x4               -0.0245544620181456     0.0227439458853017        -1.0796043106141       0.283140647017255
    x5               0.00749756488864991     0.0247118183343463      0.303399967782962       0.762270117870453
    x6             -0.000378998279974461    0.00320252606731495     -0.118343542568639       0.906053464928963
    x7                0.0782238325451951      0.195784268503804       0.39954094955119       0.690420698083317
    x8               -0.0208053274733243     0.0370350562153575     -0.561773886674887       0.575636126169296


Number of observations: 101, Error degrees of freedom: 92
Root Mean Squared Error: 1.76e-08
R-squared: 1,  Adjusted R-Squared: 1
F-statistic vs. constant model: 3.61e+15, p-value = 0
                       SumSq           DF           MeanSq                   F               pValue
                ___________________    ___    ____________________    ____________________    ______

    Total          8.91856438137271    100      0.0891856438137271
    Model          8.91856438137268      8        1.11482054767159    3.60863200846283e+15       0
    Residual    2.8421709430404e-14     92    3.08931624243522e-16

orderA = 6.000000, orderB = 2.000000
```

```
Xmin = 0.000000 and Xmax = 1.000000
Ymin = 0.000000 and Ymax = 0.841471
Model is y = [Intercept + cx1*x + cx2*x^2 + cx3*x^3 + cx4*x^4 + cx5*x^5 + cx6*x^6] /
                 [ 1 + cx7*y*x + cx8*y*x^2]
Fitting sin(x) in range (0.000000, 1.000000)
MSS of errors squared = 9.278072e-13
Particle swarm AICc =  -3.392877e+03
AIC =  -3.412855e+03
AICc =  -3.392877e+03
```

Results for the Hyperbolic Sine Function

```
Fitting sinh(x) in range (0.000000, 5.000000)

Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11

Estimated Coefficients:
                        Estimate         SE    tStat    pValue
                   __________________    __    _____    ______

    (Intercept)    -0.475800690571069    0     -Inf       0
    x1               8.64195038336725    0      Inf       0
    x2              -7.52961153097691    0     -Inf       0
    x3               1.33379714643916    0      Inf       0
    x4              0.455609617918177    0      Inf       0
    x5             -0.110999743150955    0     -Inf       0
    x6              -3.36452847184026    0     -Inf       0
    x7             -0.658320806193701    0     -Inf       0
    x8               5.72031085649003    0      Inf       0
    x9              -3.99540212994497    0     -Inf       0
    x10              1.09169726869538    0      Inf       0
    x11            -0.108701903440452    0     -Inf       0


Number of observations: 501, Error degrees of freedom: 489
R-squared: 1,  Adjusted R-Squared: 1
F-statistic vs. constant model: Inf, p-value = 0
                     SumSq          DF          MeanSq           F     pValue
                ________________    ___    __________________    ___    ______

    Total       30.8337526139307    500    0.0616675052278614
    Model       30.8337526139307     11      2.80306841944825    Inf       0
    Residual                   0    489                     0

orderA = 5.000000, orderB = 6.000000
Xmin = 0.000000 and Xmax = 5.000000
Ymin = 0.000000 and Ymax = 74.203211
Model is y = [Intercept + cx1*x + cx2*x^2 + cx3*x^3 + cx4*x^4 + cx5*x^5] /
                 [ 1 + cx6*y*x + cx7*y*x^2 + cx8*y*x^3 + cx9*y*x^4 + cx10*y*x^5 + cx11*y*x^6]
Fitting sinh(x) in range (0.000000, 5.000000)
MSS of errors squared = 5.054090e-11
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

Results for the Tangent

```
Fitting tan(x) in range (0.000000, 1.000000)
```

```
Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9

Estimated Coefficients:
                         Estimate           SE    tStat    pValue
                   _____________________    __    _____    ______

    (Intercept)    -4.20574311922127e-05    0     -Inf       0
    x1                  2.19984561284616    0      Inf       0
    x2                -0.213680098049683    0     -Inf       0
    x3                -0.239732086406133    0     -Inf       0
    x4                0.0225835353341068    0      Inf       0
    x5               0.00140998002526337    0      Inf       0
    x6                 -1.46088400329214    0     -Inf       0
    x7                 0.594323732991001    0      Inf       0
    x8                 0.126387134317526    0      Inf       0
    x9               -0.0302117503378271    0     -Inf       0


Number of observations: 101, Error degrees of freedom: 91
R-squared: 1,  Adjusted R-Squared: 1
F-statistic vs. constant model: Inf, p-value = 0
                     SumSq          DF          MeanSq           F     pValue
                ________________    ___    __________________    ___    ______

    Total       7.61910551705131    100    0.0761910551705131
    Model       7.61910551705131      9     0.846567279672368    Inf       0
    Residual                   0     91                     0

orderA = 5.000000, orderB = 4.000000
Xmin = 0.000000 and Xmax = 1.000000
Ymin = 0.000000 and Ymax = 1.557408
Model is y = [Intercept + cx1*x + cx2*x^2 + cx3*x^3 + cx4*x^4 + cx5*x^5] /
                 [ 1 + cx6*y*x + cx7*y*x^2 + cx8*y*x^3 + cx9*y*x^4]
Fitting tan(x) in range (0.000000, 1.000000)
MSS of errors squared = 1.149152e-13
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

Results for the Hyperbolic Tangent Function

```
Fitting tanh(x) in range (0.000000, 3.000000)

Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10

Estimated Coefficients:
                        Estimate          SE    tStat    pValue
                   ___________________    __    _____    ______

    (Intercept)      0.167791028100871    0      Inf       0
    x1              -0.935540124306837    0     -Inf       0
    x2                2.37070403152777    0      Inf       0
    x3               -2.97078649938341    0     -Inf       0
    x4                2.29365874096859    0      Inf       0
    x5              -0.880228670220663    0     -Inf       0
    x6               0.172387240734481    0      Inf       0
    x7             -0.0138806590578725    0     -Inf       0
    x8                1.61926511122068    0      Inf       0
    x9              -0.862043277098288    0     -Inf       0
```

x10	0.0386730872018001	0	Inf	0

Number of observations: 301, Error degrees of freedom: 290
R-squared: 1, Adjusted R-Squared: 1
F-statistic vs. constant model: Inf, p-value = 0

	SumSq	DF	MeanSq	F	pValue
Total	23.2802959868209	300	0.0776009866227363		
Model	23.2802959868209	10	2.32802959868209	Inf	0
Residual	0	290	0		

```
orderA = 7.000000, orderB = 3.000000
Xmin = 0.000000 and Xmax = 3.000000
Ymin = 0.000000 and Ymax = 0.995055
Model is y = [Intercept + cx1*x + cx2*x^2 + cx3*x^3 + cx4*x^4 + cx5*x^5 + cx6*x^6 + cx7*x^7] /
                 [ 1 + cx8*y*x + cx9*y*x^2 + cx10*y*x^3]
Fitting tanh(x) in range (0.000000, 3.000000)
MSS of errors squared = 1.524299e-10
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

Results for the Two-Sided T-Inverse Distribution

Fitting tinv(0.975,x) in range (2.000000, 100.000000)

Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	0.979119303945661	0	Inf	0
x1	-1.878994097881	0	-Inf	0
x2	0.00955133125518624	0	Inf	0
x3	1.29765857899649	0	Inf	0
x4	0	0	NaN	NaN
x5	-0.407364182564784	0	-Inf	0
x6	-7.57017719835083e-09	0	-Inf	0
x7	1.90822588885485	0	Inf	0
x8	0	0	NaN	NaN
x9	-1.31549494995822	0	-Inf	0
x10	-0.00434262168361492	0	-Inf	0
x11	0.411626219696284	0	Inf	0

Number of observations: 99, Error degrees of freedom: 89
R-squared: 1, Adjusted R-Squared: 1
F-statistic vs. constant model: Inf, p-value = 0

	SumSq	DF	MeanSq	F	pValue
Total	1.4314772743268	98	0.0146069109625184		
Model	1.4314772743268	9	0.159053030480756	Inf	0
Residual	0	89	0		

orderA = 6.000000, orderB = 5.000000
Xmin = 2.000000 and Xmax = 100.000000

```
Ymin = 1.983972 and Ymax = 4.302653
Model is y = [Intercept + cx1*x + cx2*x^2 + cx3*x^3 + cx4*x^4 + cx5*x^5 + cx6*x^6] /
                 [ 1 + cx7*y*x + cx8*y*x^2 + cx9*y*x^3 + cx10*y*x^4 + cx11*y*x^5]
Fitting tinv(0.975,x) in range (2.000000, 100.000000)
MSS of errors squared = 4.680093e-14
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

Results for the One-Sided T-Inverse Distribution

```
Fitting tinv(0.95,x) in range (2.000000, 100.000000)

Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8

Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
(Intercept)	0.975312316906565	0	Inf	0
x1	-3.00234264235221	0	-Inf	0
x2	3.08073968338658	0	Inf	0
x3	-1.05372700285936	0	-Inf	0
x4	3.06540517416161	0	Inf	0
x5	-3.1321477062934	0	-Inf	0
x6	1.06675137217174	0	Inf	0
x7	-1.99493718593477e-08	0	-Inf	0
x8	2.3685685960035e-09	0	Inf	0

```
Number of observations: 99, Error degrees of freedom: 90
R-squared: 1,  Adjusted R-Squared: 1
F-statistic vs. constant model: Inf, p-value = 0
```

	SumSq	DF	MeanSq	F	pValue
Total	1.51844154817134	98	0.0154943015119524		
Model	1.51844154817134	8	0.189805193521417	Inf	0
Residual	0	90	0		

```
orderA = 3.000000, orderB = 5.000000
Xmin = 2.000000 and Xmax = 100.000000
Ymin = 1.660234 and Ymax = 2.919986
Model is y = [Intercept + cx1*x + cx2*x^2 + cx3*x^3] /
                 [ 1 + cx4*y*x + cx5*y*x^2 + cx6*y*x^3 + cx7*y*x^4 + cx8*y*x^5]
Fitting tinv(0.95,x) in range ((2.000000, 100.000000)
MSS of errors squared = 6.748744e-14
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

Results for the Common Logarithm of the Gamma Function

```
Fitting log10Gamma(x) in range (2.000000, 100.000000)

Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10
```

```
Estimated Coefficients:
                           Estimate          SE    tStat    pValue
                     __________________    __    _____    ______

    (Intercept)    -0.0810158671459593    0     -Inf       0
    x1              -0.118627827538114    0     -Inf       0
    x2                1.32026727678648    0      Inf       0
    x3               -4.09275990701905    0     -Inf       0
    x4                5.46139474348187    0      Inf       0
    x5               -2.50113320772895    0     -Inf       0
    x6                3.66856164077954    0      Inf       0
    x7               -3.10237937749939    0     -Inf       0
    x8               -1.19225524927864    0     -Inf       0
    x9                1.66170491724495    0      Inf       0
    x10            -0.0237571328294603    0     -Inf       0


Number of observations: 99, Error degrees of freedom: 88
R-squared: 1,  Adjusted R-Squared: 1
F-statistic vs. constant model: Inf, p-value = 0
                       SumSq          DF          MeanSq           F     pValue
                  ________________    __    __________________    ___    ______

    Total       9.07288178684988    98    0.0925804263964273                  
    Model       9.07288178684988    10     0.907288178684988    Inf       0   
    Residual                   0    88                     0                  

orderA = 5.000000, orderB = 5.000000
Xmin = 2.000000 and Xmax = 100.000000
Ymin = 0.000000 and Ymax = 155.970004
Model is y = [Intercept + cx1*x + cx2*x^2 + cx3*x^3 + cx4*x^4 + cx5*x^5] /
                 [ 1 + cx6*y*x + cx7*y*x^2 + cx8*y*x^3 + cx9*y*x^4 + cx10*y*x^5]
Fitting log10Gamma(x) in range ((2.000000, 100.000000)
MSS of errors squared = 7.841470e-10
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

Results for the Digamma Function

```
Fitting digamma(x) in range (1.000000, 100.000000)

Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11 + x12 + x13

Estimated Coefficients:
                           Estimate          SE    tStat    pValue
                     __________________    __    _____    ______

    (Intercept)       3.55657619628626    0      Inf       0
    x1               -29.7038549951656    0     -Inf       0
    x2                81.0509857744355    0      Inf       0
    x3               -90.6479862422103    0     -Inf       0
    x4                28.5446256329591    0      Inf       0
    x5                17.8511502495401    0      Inf       0
    x6               -10.6516195856764    0     -Inf       0
    x7                 11.554860392583    0      Inf       0
    x8               -36.8291014514652    0     -Inf       0
    x9                47.4044612398808    0      Inf       0
    x10              -21.2339184432746    0     -Inf       0
```

```
    x11            -4.24151166138219    0     -Inf       0
    x12             4.36669670323192    0      Inf       0
    x13           -0.0213638097402183   0     -Inf       0


Number of observations: 100, Error degrees of freedom: 86
R-squared: 1,  Adjusted R-Squared: 1
F-statistic vs. constant model: Inf, p-value = 0
                     SumSq          DF         MeanSq           F     pValue
                ________________    __    __________________    ___   ______

    Total       3.53709271459914    99    0.0357282092383752
    Model       3.53709271459914    13     0.272084054969165    Inf     0
    Residual                   0    86                     0

orderA = 6.000000, orderB = 7.000000
Xmin = 1.000000 and Xmax = 100.000000
Ymin = -0.577216 and Ymax = 4.600162
Model is y = [Intercept + cx1*x + cx2*x^2 + cx3*x^3 + cx4*x^4 + cx5*x^5 + cx6*x^6] /
                 [ 1 + cx7*y*x + cx8*y*x^2 + cx9*y*x^3 + cx10*y*x^4 + cx11*y*x^5 + cx12*y*x^6 + cx13*y*x^7]
Fitting digamma(x) in range (1.000000, 100.000000)
MSS of errors squared = 3.376550e-11
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

RESULTS FOR THE TRIGAMMA FUNCTION

```
Fitting tigamma(x) in range (1.000000, 100.000000)

Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11 + x12

Estimated Coefficients:
                        Estimate           SE    tStat    pValue
                   _____________________    __    _____    ______

    (Intercept)         0.98764836765494    0      Inf       0
    x1                 -4.03311154322681    0     -Inf       0
    x2                   6.4684691231006    0      Inf       0
    x3                 -5.08078932084451    0     -Inf       0
    x4                  1.95025824266778    0      Inf       0
    x5                -0.292476260537502    0     -Inf       0
    x6               2.42316514620585e-06   0      Inf       0
    x7              -2.12674752882649e-07   0     -Inf       0
    x8                  4.07722845370951    0      Inf       0
    x9                 -6.52993901381203    0     -Inf       0
    x10                  5.1225557671214    0      Inf       0
    x11                -1.96411913323873    0     -Inf       0
    x12                0.294273516564833    0      Inf       0


Number of observations: 100, Error degrees of freedom: 87
R-squared: 1,  Adjusted R-Squared: 1
F-statistic vs. constant model: Inf, p-value = 0
                     SumSq          DF         MeanSq           F     pValue
                ________________    __    __________________    ___   ______

    Total       1.20244526290484    99    0.0121459117465136
    Model       1.20244526290484    12     0.100203771908737    Inf     0
    Residual                   0    87                     0
```

```
orderA = 7.000000, orderB = 5.000000
Xmin = 1.000000 and Xmax = 100.000000
Ymin = 0.010050 and Ymax = 1.644935
Model is y = [Intercept + cx1*x + cx2*x^2 + cx3*x^3 + cx4*x^4 + cx5*x^5 + cx6*x^6 + cx7*x^7] /
                 [ 1 + cx8*y*x + cx9*y*x^2 + cx10*y*x^3 + cx11*y*x^4 + cx12*y*x^5]
Fitting tigamma(x) in range (1.000000, 100.000000)
MSS of errors squared = 5.253660e-12
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

SUMMARY

The following table shows the summary of the above results. Notice that the regression coefficients for polynomials $Q_m(x)$ and $D_n(x)$ appear as one long vertical list. The output that starts with *Model is y = [Intercept +* tells you which coefficients belong to $Q_m(x)$ and which ones belong to $D_n(x)$.

Function	*Order Q(x)*	*Order D(x)*	*F*	*AICc*	*MSSE*
arccos(x)	7	3	Infinity	-Infinity	9.165132E-10
arcsin(x)	5	6	Infinity	-Infinity	1.936336E-10
arctan(x)	5	6	Infinity	-Infinity	1.083466E-12
cosh(x)	7	2	Infinity	-Infinity	5.717733E-10
cos(x)	3	4	Infinity	-Infinity	2.504558E-10
digamma(x)	6	7	Infinity	-Infinity	3.376550e-11
erf(x)	7	4	7.29493E+14	-6.688183E+03	7.212629E-11
exp(x)	3	7	Infinity	-Infinity	5.899982E-15
ln(x)	6	6	Infinity	-Infinity	8.896563E-12
log10Gamma(x)	5	5	Infinity	-Infinity	7.841470E-10
log(x)	5	5	Infinity	-Infinity	7.292823E-11
10^x	4	3	Infinity	-Infinity	1.120822E-10
sinh(x)	5	6	Infinity	-Infinity	5.054090E-11
sin(x)	6	2	3.60863200846283e+15	-3.392877E+03	9.278072E-13
tanh(x)	7	3	Infinity	-Infinity	1.524299E-10
tan(x)	5	4	Infinity	-Infinity	1.149152E-13
tinv(0.95,x)	3	5	Infinity	-Infinity	6.748744E-14
tinv(0.975,x)	6	5	Infinity	-Infinity	4.680093E-14
tigamma(x)	7	5	Infinity	-Infinity	5.253660E-12

CONCLUSIONS

The results of the Pade approximations show that the Pade polynomials perform an excellent job in approximating various functions. All the coefficients of determination obtained for the optimum Pade polynomials are 1 (or extremely close to it). The optimum orders for polynomials $Q_m(x)$ and $D_n(x)$ vary for different functions. This variation shows that the particle swarm optimization succeeds in getting optimum orders for polynomials $Q_m(x)$ and $D_n(x)$ and is not always attracted to the order 7—the maximum order allowed.

DOCUMENT HISTORY

Date	***Version***	***Comments***
9/5/2020	1.00.00	Initial release.
9/6/2020	1.00.01	Corrections for equation 1 and ranges in the output.
9/8/2020	1.00.02	Minor equation edits.
9/16/2020	1.00.03	Edited equation 5.

