Hybrid Quadratic Fourier-Shammas Series Approximations
By
Namir C. Shammas

Copyright © 2020 by Namir Clement Shammas Version 1.00.00

INTRODUCTION

This study picks up where the study *Fourier-Shammas Series Approximations* left off. The conclusion of the latter study suggested that adding linear and quadratic terms could effectively enhance the Fourier-Shammas series. This study does just

Copyright © 2020 by Namir Clement Shammas

that, going over all of the function approximations from the last study with the proposed enhancement.

The Fourier Series

Fourier polynomials are polynomial ratios defined as:

$$y = a_0 + \sum_{i=1}^{n} a_i \sin(i \cdot \pi \cdot x) + \sum_{i=1}^{n} b_i \cos(i \cdot \pi \cdot x) \quad (1)$$

The multiplier i in the sine and cosine terms is an integer.

The Shammas Sequence

In the HHC 2008 conference in Corvallis, Oregon, I introduced the Shammas polynomials as polynomials with non-integer powers. I explained that the powers of such polynomial change using some math expressions that involves the polynomial term sequence number, possible transformations of the term sequence number, and constants. The sequences that I use in this study are:

Power = A + B*i (2)

Power = A + B/i (3)

Power = A+B*sqrt(i) (4)

Power = A + B * $[\log_{10}(i)]^4$ (5)

Where i is the number of term (1, 2, 3, and so on). A, B, and C are parameters that allow the above power expression to progress in smaller and non-integer increments.

Copyright © 2020 by Namir Clement Shammas

👉 The sequences of regular polynomial powers and Fourier series are integers that are generally defined as:

gx(i, A) = A*i

Where A is typically 1 or 2 for polynomials and π for Fourier series. The Shammas sequences support progressions of smaller values that are not necessarily integers. The *general* forms of the Shammas sequences are:

gx(i, A, B) < i, for i > 0 and gx(i, A, B) < gx(i+|Δi|, A, B)

gx(i, A, B, C) < i, for i > 0 and gx(i, A, B, C) < gx(i+|Δi|, A, B, C)

Where A is a constant that appears as the first term. The second term is constant B multiplied by a function of the sequence number i (as shown above). The third term, if used, is constant C multiplied by another function of i, or better yet i–1. These rules hold true except for A+B/i (or any other form that divides by i) where the sequence values decrease as i increases. Thus gx() can be:

gx(i, A, B) = A + B*g(i)

gx(i, A, B, C) = A + B*g_1(i) + C*g_2(i)

gx(i, A, B, C) = A + B*g_1(i) + C*g_2(i–1)

Where g(), g_1(), and g_2() are functions such as the square root, natural logarithm, common logarithm, reciprocal, and so on. The simplest Shammas sequence is:

gx(i, A, B) = A + B* √i with A = 0 and B = 1.
= √i

The Fourier-Shammas Series

The Fourier-Shammas series combine the Fourier series and Shammas sequence concepts. In a previous study (titled *Fourier-Shammas Approximations*) I used the following multiple regression general form:

$$y = a_0 + \sum_{i=1}^{n} a_i \sin(S_i * gx_i(i, A_i, B_i) * x + Os_i) \quad (6)$$

Equation 6 shows the Fourier-Shammas Sine Series.

Copyright © 2020 by Namir Clement Shammas

$$y = a_0 + \sum_{i=1}^{n} a_i \cos(C_i * gx_i(i, A_i, B_i) * x + Oc_i) \quad (7)$$

Equation 7 shows the Fourier-Shammas Cosine Series.

$$y = a_0 + \sum_{i=1}^{n} a_i \sin(S_i * gx_i(i, As_i, Bs_i) * x + Os_i) + \sum_{i=1}^{m} a_i \cos(C_i * gx_i(i, Ac_i, Bc_i) * x + Oc_i) \quad (8)$$

Equation 8 shows the Fourier-Shammas Sine/Cosine series.

The Hybrid Quadratic Fourier-Shammas (HQFS for short) series. in this study, adds a linear term and a quadratic term to the various Fourier-Shammas series. This enhancement yields the following equations:

$$y = a_0 + \sum_{i=1}^{n} a_i \sin(S_i * gx_i(i, A_i, B_i) * x + Os_i) + b_1*x + b_2*x^2 \quad (9)$$

Equation 9 shows the HQFS Sine Series.

$$y = a_0 + \sum_{i=1}^{n} a_i \cos(C_i * gx_i(i, A_i, B_i) * x + Oc_i) + b_1*x + b_2*x^2 \quad (10)$$

Equation 10 shows the HQFS Cosine Series.

$$y = a_0 + \sum_{i=1}^{n} a_i \sin(S_i * gx_i(i, As_i, Bs_i) * x + Os_i) + \sum_{i=1}^{m} a_i \cos(C_i * gx_i(i, Ac_i, Bc_i) * x + Oc_i) + b_1*x + b_2*x^2 \quad (11)$$

Equation 11 shows the HQFS Sine/Cosine Series.

The linear and quadratic terms add more robustness to the models. In these enhanced models, the linear and quadratic terms provide a rough approximation that is refined by the sine and/or cosine terms.

The next table lists the HQFS series and their orders.

Part	***Models***
Sine series for n=3 to 7	y = a(0) + a(1)*sin(C1*x*gx(1,A1,B1)+Os1)) + … + a(n)*sin(Cn*x*gx(n,An,Bn)+Os(n)) + a(n+1)*x + a(n+2)*x^2
Cosine Series for n=3 to 7	y = a(0) + a(1)*cos(C1*x*gx(1,A1,B1)+Os(1)) + … + a(n)*cos(Cn*x*gx(n,An,Bn)+Os(n)) + a(n+1)*x + a(n+2)*x^2
Alternating Sine and cosine Series for n=3 to 7	y = a(0) + a(1)*sin(C1*x*gx(1,A1,B1)+Os(1)) + a(2)*cos(C2*x*gx(2,A2,B2)+Os(2)) + a(3)*sin(C3*x*gx(n,A3,B3)+Os(3)) +... +

Copyright © 2020 by Namir Clement Shammas

Part	*Models*
	a(n)*sin(Cn*x*gx(n,An,Bn)+Os(n)) + a(n+1)*x + a(n+2)*x^2
Alternating Cosine and sine Series for n = 7 ONLY	y = a(0) + a(1)*cos(C1*x*gx(1,A1,B1)+Os(1)) + a(2)*sin(C2*x*gx(2,A2,B2)+Os(2)) + a(3)*cos(C3*x*gx(n,A3,B3)+Os(3)) +... + a(7)*cos(Cn*x*gx(7,A7,B7)+Os(7)) + a(8)*x + a(9)*x^2

In the above tables, the coefficients a(0) through a(n+2) are calculated as multiple regression coefficients. The rest of the coefficients (Cs, As, Bs, and the offset values Os are different for each term and are calculated based on optimization). The addition of an offset term inside each sine and cosine plus having different multipliers C(i) and gx parameters A(i) and B(i) gives the models in the above table more flexibility.

This study looks at HQFS series used to approximate common functions that include:

- Trigonometric functions and their inverses.
- Hyperbolic functions.
- Logarithmic functions.
- Exponential functions.
- Bessel functions $J_0(x)$ to $J_5(x)$.
- The sine and cosine integrals.
- The Fresnel sine and cosine.
- Inverse student-t functions.
- The common logarithm of the gamma function.
- The digamma function.
- The trigamma function.

The digamma function is defined as:

$$\psi(x) = \Gamma'(x) \,/\, \Gamma(x) = \frac{d \ln(\Gamma(x))}{dx} \qquad (12)$$

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;
```

Copyright © 2020 by Namir Clement Shammas

```
  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:

$$\psi_1(x) = \frac{d^2}{dx^2} \ln(\Gamma(x)) \tag{13}$$

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 = gammaln(x+h);
  fm = gammaln(x-h);
  f0 = gammaln(x);
  y = (fp -2*f0 + fm)/h/h;
end
```

There are several choices that we can use to determine the best HQFS series parameters C, A, B and O. The best one is to use an optimization function to determine the best values for parameters C, A, B, and Os, for a given HQFS series order. The study uses the Matlab particle swarm optimization function to select the best values for Cs, As, Bs, and Os.

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 if 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:

$\log_{10}(235)$ =

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

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

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

Copyright © 2020 by Namir Clement Shammas Version 1.00.00

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

MATLAB CODE

The Sine Series

The algorithm in this study uses particle swarm optimization to obtain the best values for parameters Cs, As, Bs, and Os in a prespecified range of Fourier-Shammas series orders. The order of the series used is set in the caller routine using the global variable order.

This study uses the function FourierShammasSeries2() to perform various HQFS series curve fitting:

```
function
FourierShammasSeries2(fx,gx,xRange,Lb,Ub,runNum,sFxName,diaryFilename)
% FourierShammasSeries2 implements the Fourier-Shammas Sine Series
% Model:
% y = a(0) + a(1)*sin(S(1)*gx(1,A1,B1)*x+Os1) +...+
a(n)*sin(S(n)*gx(n,An,Bn)*x+Osn) +
%      b(1)*x + b(2)*x^2
clc
global bDeleteIfExists
global bUseDiary
global xdata
global ydata
global order
global ggx

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
ggx = gx;

fprintf('Fitting %s in range (%f, %f)\n', sFxName, min(xdata),max(xdata));

options = optimoptions('particleswarm', 'Display', 'iter');
[x,psAICc] = particleswarm(@optimFunc,length(Lb),Lb,Ub,options);
if isinf(psAICc), psAICc = -1e+99; end
S = zeros(order,1);
As = zeros(order,1);
Bs = zeros(order,1);
Os = zeros(order,1);
S = x(1:order);
As = x(order+1:2*order);
Bs = x(2*order+1:3*order);
Os = x(3*order+1:4*order);
```

Copyright © 2020 by Namir Clement Shammas

```
if bUseDiary
  diaryFilename = strrep(diaryFilename, ".txt", strcat("_",
num2str(order),"_sin_run", num2str(runNum),".txt"));
  if exist(diaryFilename, 'file')==2
    if bDeleteIfExists
      delete(diaryFilename);
    else
      return;
    end
  end
end
X = [];
for i=1:order
  g = S(i)*gx(i,As(i),Bs(i));
  xs = sin(xdata.*g + Os(i));
  X = [X;xs'];
end
xs = xdata;
X = [X;xs'];
xs = xdata.^2;
X = [X;xs'];
X = X';
lm = fitlm(X,ydata);
if bUseDiary
  diary(diaryFilename)
end
fprintf('Fitting %s in range (%f, %f)\n', sFxName, min(xdata),max(xdata));
sp = getFuncName(gx);
fprintf('Fourier Shammas Series factor is %s\n', sp);
format long
disp(lm);
anva = anova(lm,'summary');
disp(anva);
format short
fprintf('Model is ----------------------------------\n')
fprintf("y = %e +\n", lm.Coefficients{1,1});
k=2;
for i=1:order
  g = S(i)*gx(i,As(i),Bs(i));
  if g >= 0 && Os(i) > 0
    fprintf("    %e * sin(%e * x + %f) + \n", lm.Coefficients{k,1}, g,
Os(i));
  elseif g >= 0 && Os(i) < 0
    fprintf("    %e * sin(%e * x - %f) + \n", lm.Coefficients{k,1}, g,
abs(Os(i)));
  elseif g >= 0 && Os(i) == 0
    fprintf("    %e * sin(%e * x) + \n", lm.Coefficients{k,1}, g);
  elseif g < 0 && Os(i) > 0
    fprintf("    %e * sin((%e) * x + %f) + \n", lm.Coefficients{k,1}, g,
Os(i));
  elseif g < 0 && Os(i) == 0
    fprintf("    %e * sin((%e) * x) + \n", lm.Coefficients{k,1}, g);
  else % both g and Os(i) are negative
    fprintf("    %e * sin((%e) * x - %f) + \n", lm.Coefficients{k,1}, g,
abs(Os(i)));
  end
  k = k + 1;
```

Copyright © 2020 by Namir Clement Shammas

```
end
fprintf("    (%e)*x + (%e)*x^2", lm.Coefficients{order+2,1},
lm.Coefficients{order+3,1});
fprintf("\n");

lstFactor = [];
lstOffset = [];
for i=1:order
  g = S(i)*gx(i,As(i),Bs(i));
  lstFactor = [lstFactor,g];
  lstOffset = [lstOffset, Os(i)];
end
fprintf('List of factors: [');
for i=1:order-1
 fprintf('%f, ', lstFactor(i));
end
fprintf('%f]\n', lstFactor(order));
fprintf('List of offsets: [');
for i=1:order-1
 fprintf('%f, ', lstOffset(i));
end
fprintf('%f]\n', lstOffset(order));
fprintf('Fitting %s in range (%f, %f)\n', sFxName, min(xdata),max(xdata));
n = length(xdata);
sumsqr = 0;
for i=1:n
  yc = lm.Coefficients{1,1};
  for j=2:order+1 % length(lm.Coefficients{:,1})
    yc = yc + lm.Coefficients{j,1} * sin(xdata(i)*lstFactor(j-1)+Os(j-1));
  end
  yc = yc + lm.Coefficients{order+2,1}*xdata(i) +
lm.Coefficients{order+3,1}*xdata(i)^2;
  sumsqr = sumsqr + (ydata(i) - yc)^2;
end
k = order + 3;
fprintf('MSS of errors squared = %e\n', sqrt(sumsqr)/n);
fprintf("R-Squared = %12.8f\n", lm.Rsquared.Ordinary);
fprintf("R-Squared Adjusted = %12.8f\n", lm.Rsquared.Adjusted);
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 order
  global ggx

  S = zeros(order,1);
  As = zeros(order,1);
```

```
  Bs = zeros(order,1);
  Os = zeros(order,1);
  S = x(1:order);
  As = x(order+1:2*order);
  Bs = x(2*order+1:3*order);
  Os = x(3*order+1:4*order);
  X = [];
  for i=1:order
      g = S(i)*ggx(i,As(i),Bs(i));
    xs = sin(xdata*g + Os(i));
    X = [X;xs'];
  end
  xs = xdata;
  X = [X;xs'];
  xs = xdata.^2;
  X = [X;xs'];
  X = X';
  lm = fitlm(X,ydata);

  n = length(xdata);
  k = order + 3;
  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 FourierShammasSeries2() 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 gx is the handle (or inline function) for the function that calculates the powers of the Shammas polynomial. An example is @(i,A,B)A+B*sqrt(i) which also shows the ***recommended format*** for the argument of parameter gx.
- 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 the array of lower limits for the parameters S, As, Bs, and Os.
- The parameter Ub is the array of upper limits for the parameters S, As, Bs, and Os.
- 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. Pass the order of the HQFS series using the global variable order specified by the caller routine.
2. Initialize the data for the curve fitting. The function uses the global variables xdata and ydata to store the data for the series fitting.
3. Store the handle of function gx in the global handle ggx.
4. Set the optimization options and then call the Matlab function particleswarm(). The function call returns the optimized values of S, As, Bs, Os, and the optimum value for *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 equal to the number of elements in parameter Lb.
    c. The lower and upper bounds arrays, Lb and Ub, respectively,
    d. The optimization parameters for function particleswarm().
5. Retrieve the optimum values and perform a HQFS series fit for the best values of S, As, Bs, and Os.
6. Assemble the data for the multiple regression of the model. Notice that the above code includes the linear and quadratic terms in addition to the various sine terms.
7. Perform the curve fitting of the nest model by calling function fitlm and passing it the arguments for the data matrix X and vector ydata.
8. Display the results of the regression and its associated ANOVA table.

9. Display the HQFS model. This is the form that you can use in predicting other values.
10. Calculate and display the list of Fourier-Shammas series factor.
11. Display the range of the approximated function.
12. Calculate and display the value of the mean square root of the sum of squared errors.
13. Display the coefficient of determination and its adjusted value. The latter statistic serves as a measure of goodness of model fitting.
14. Calculate and display the *corrected* Akaike information criterion. This statistic is calculated using:

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

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

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.

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

The function optimFunc() obtains the array x containing the current values of the parameters of the HQFS series. The function calculates the transformed variables needed to perform a curve fit for a HQFS series. 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 Shammas polynomial orders and thus a varying number of polynomial coefficients. One last thing to keep in mind. The optimization function uses an implementation of the Particle Swarm Optimization algorithm. This method uses random numbers to search for the optimum values. As such, the results can vary between different runs.

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) and @(x,A,B)A+B*sqrt(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) and A+B*sqrt(x) for the above examples). 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

Copyright © 2020 by Namir Clement Shammas

indicates that you a rereferencing a separate Matlab file that implements the code for fx1.

	Programming Notes
	The above listing uses the variables S, As, Bs, and Os to represent the variables C, A, B, and Os that I showed in an earlier table of equations. I have renamed C as S and used the trailing letter s to signal that we are dealing with sine terms. The next listing which deals with cosine terms uses the variables C, Ac, Bc, and Oc to represent the variables C, A, B, and Os that I showed in an earlier table of equations. I use the trailing letter c to signal that we are dealing with cosine terms. The upcoming listing which deals with sine and cosine terms uses the variables S, As, Bs, and Os for sine terms and C, Ac, Bc, and Oc cosine terms. The conclusion to draw from the above comments that I have simplified the mathematical represetations of the series in the tables and equations that I presented earlier. In coding the various trigonometric series, I used a different naming scheme for the variables related to the sine and to the cosine terms. This scheme makes is easier to implement and read the code, especially one that involves both sine and cosine calculations.

The Cosine Series

The next listing shows a version of the FourierShamSeries2() coded to model HQFS Cosine series.

```
function
FourierShammasSeries2(fx,gx,xRange,Lb,Ub,runNum,sFxName,diaryFilename)
%FourierShammasSeries2 Summary of this function goes here
% Model:
% y = a(0) + a(1)*cos(c(1)*gx(1,A1,B1))*X + ... + a(n)*cos(c(N)*gx(n,AN,BN))
+
%     a(n+1)*x + a(n+2)*x^2
clc
global bDeleteIfExists
global bUseDiary
global xdata
global ydata
global order
global ggx

warning('off','all')
if isempty(sFxName)
```

Copyright © 2020 by Namir Clement Shammas

```
  sFxName = getFuncName(fx);
end
xdata = xRange';
ydata = xdata;
for i=1:length(xdata)
  ydata(i)=fx(xdata(i));
end
ggx = gx;

fprintf('Fitting %s in range (%f, %f)\n', sFxName, min(xdata),max(xdata));

options = optimoptions('particleswarm', 'Display', 'iter');
[x,psAICc] = particleswarm(@optimFunc,length(Lb),Lb,Ub,options);
if isinf(psAICc), psAICc = -1e+99; end
C = zeros(order,1);
Ac = zeros(order,1);
Bc = zeros(order,1);
Oc = zeros(order,1);
C = x(1:order);
Ac = x(order+1:2*order);
Bc = x(2*order+1:3*order);
Oc = x(3*order+1:4*order);

if bUseDiary
  diaryFilename = strrep(diaryFilename, ".txt", strcat("_",
num2str(order),"_cos_run", num2str(runNum),".txt"));
  if exist(diaryFilename, 'file')==2
    if bDeleteIfExists
      delete(diaryFilename);
    else
      return;
    end
  end
end
X = [];
for i=1:order
  g = C(i)*gx(i,Ac(i),Bc(i));
  xs = cos(xdata.*g + Oc(i));
  X = [X;xs'];
end
xs = xdata;
X = [X;xs'];
xs = xdata.^2;
X = [X;xs'];
X = X';
lm = fitlm(X,ydata);
if bUseDiary
  diary(diaryFilename)
end
fprintf('Fitting %s in range (%f, %f)\n', sFxName, min(xdata),max(xdata));
sp = getFuncName(gx);
fprintf('Fourier Shammas Series factor is %s\n', sp);
format long
disp(lm);
anva = anova(lm,'summary');
disp(anva);
format short
```

```
fprintf('Model is ----------------------------------\n')
fprintf("y = %e +\n", lm.Coefficients{1,1});
k=2;
for i=1:order
  g = C(i)*gx(i,Ac(i),Bc(i));
  if g >= 0 && Oc(i) > 0
    fprintf("    %e * cos(%e * x + %f) + \n", lm.Coefficients{k,1}, g,
Oc(i));
  elseif g >= 0 && Oc(i) < 0
    fprintf("    %e * cos(%e * x - %f) + \n", lm.Coefficients{k,1}, g,
abs(Oc(i)));
  elseif g >= 0 && Oc(i) == 0
    fprintf("    %e * cos(%e * x) + \n", lm.Coefficients{k,1}, g);
  elseif g < 0 && Oc(i) > 0
    fprintf("    %e * cos((%e) * x + %f) + \n", lm.Coefficients{k,1}, g,
Oc(i));
  elseif g < 0 && Oc(i) == 0
    fprintf("    %e * cos((%e) * x) + \n", lm.Coefficients{k,1}, g);
  else % both g and Oc(i) are negative
    fprintf("    %e * cos((%e) * x - %f) + \n", lm.Coefficients{k,1}, g,
abs(Oc(i)));
  end
  k = k + 1;
end
fprintf("    (%e)*x + (%e)*x^2", lm.Coefficients{order+2,1},
lm.Coefficients{order+3,1});
fprintf("\n");

lstFactor = [];
lstOffset = [];
for i=1:order
  g = C(i)*gx(i,Ac(i),Bc(i));
  lstFactor = [lstFactor,g];
  lstOffset = [lstOffset, Oc(i)];
end
fprintf('List of factors: [');
for i=1:order-1
 fprintf('%f, ', lstFactor(i));
end
fprintf('%f]\n', lstFactor(order));
fprintf('List of offsets: [');
for i=1:order-1
 fprintf('%f, ', lstOffset(i));
end
fprintf('%f]\n', lstOffset(order));
fprintf('Fitting %s in range (%f, %f)\n', sFxName, min(xdata),max(xdata));
n = length(xdata);
sumsqr = 0;
for i=1:n
  yc = lm.Coefficients{1,1};
  for j=2:order+1 % length(lm.Coefficients{:,1})
    yc = yc + lm.Coefficients{j,1} * cos(xdata(i)*lstFactor(j-1)+Oc(j-1));
  end
  yc = yc + lm.Coefficients{order+2,1}*xdata(i) +
lm.Coefficients{order+3,1}*xdata(i)^2;
  sumsqr = sumsqr + (ydata(i) - yc)^2;
end
```

```
k = order + 3;
fprintf('MSS of errors squared = %e\n', sqrt(sumsqr)/n);
fprintf("R-Squared = %12.8f\n", lm.Rsquared.Ordinary);
fprintf("R-Squared Adjusted = %12.8f\n", lm.Rsquared.Adjusted);
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 order
  global ggx

  C = zeros(order,1);
  Ac = zeros(order,1);
  Bc = zeros(order,1);
  Oc = zeros(order,1);
  C = x(1:order);
  Ac = x(order+1:2*order);
  Bc = x(2*order+1:3*order);
  Oc = x(3*order+1:4*order);
  X = [];
  for i=1:order
      g = C(i)*ggx(i,Ac(i),Bc(i));
    xs = cos(xdata*g + Oc(i));
    X = [X;xs'];
  end
  xs = xdata;
  X = [X;xs'];
  xs = xdata.^2;
  X = [X;xs'];
  X = X';
  lm = fitlm(X,ydata);

  n = length(xdata);
  k = order + 3;
  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 above listing is very similar to the one before it. It simply replaces reference of sine function with the cosine function and renames the variables S, As, Bs, and Os.

The Alternating Sine/Cosine Series

The next listing shows the version of FourierShammasSeries2() for the HQFS alternating sine/cosine series.

```
function
FourierShammasSeries2(fx,gx,xRange,Lb,Ub,runNum,sFxName,diaryFilename)
%FourierShammasSeries2 implements the Fourier-Shammas Sine/Cosine series
% Model:
% y = a(0) + a(1)*sin(S1*gx(1,A1,B1)+Os1) + b(2)*cos(C1*gx(2,A2,B2)*x+Oc1) +
... +
%            a(n)*sin(Sn*gx(n,Asn,Bsn)+Osn) + a(n+1)*x + a(n+2)*x^3
clc
global bDeleteIfExists
global bUseDiary
global xdata
global ydata
global order
global ggx

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
ggx = gx;

fprintf('Fitting %s in range (%f, %f)\n', sFxName, min(xdata),max(xdata));

options = optimoptions('particleswarm', 'Display', 'iter');
[x,psAICc] = particleswarm(@optimFunc,length(Lb),Lb,Ub,options);
if isinf(psAICc), psAICc = -1e+99; end
S = zeros(order,1);
As = zeros(order,1);
Bs = zeros(order,1);
Os = zeros(order,1);
S = x(1:order);
As = x(order+1:2*order);
Bs = x(2*order+1:3*order);
Os = x(3*order+1:4*order);

if bUseDiary
  diaryFilename = strrep(diaryFilename, ".txt", strcat("_",
num2str(order),"_run", num2str(runNum),".txt"));
  if exist(diaryFilename, 'file')==2
```

```
    if bDeleteIfExists
      delete(diaryFilename);
    else
      return;
    end
  end
end
X = [];
flag = 1;
for i=1:order
  g = S(i)*gx(i,As(i),Bs(i));
  if flag > 0
    xs = sin(xdata.*g + Os(i));
  else
      xs = cos(xdata.*g + Os(i));
  end
  flag = 1 - flag;
  X = [X;xs'];
end
xs = xdata;
X = [X;xs'];
xs = xdata.^2;
X = [X;xs'];
X = X';
lm = fitlm(X,ydata);
if bUseDiary
  diary(diaryFilename)
end
fprintf('Fitting %s in range (%f, %f)\n', sFxName, min(xdata),max(xdata));
sp = getFuncName(gx);
fprintf('Fourier Shammas Series factor is %s\n', sp);
format long
disp(lm);
anva = anova(lm,'summary');
disp(anva);
format short
fprintf('Model is ----------------------------------\n')
fprintf("y = %e +\n", lm.Coefficients{1,1});
k=2;
flag = 1;
for i=1:order
  g = S(i)*gx(i,As(i),Bs(i));
  if flag > 0
        if g >= 0 && Os(i) > 0
          fprintf("    %e * sin(%e * x + %f) + \n", lm.Coefficients{k,1}, g,
Os(i));
        elseif g >= 0 && Os(i) < 0
          fprintf("    %e * sin(%e * x - %f) + \n", lm.Coefficients{k,1}, g,
abs(Os(i)));
        elseif g >= 0 && Os(i) == 0
          fprintf("    %e * sin(%e * x) + \n", lm.Coefficients{k,1}, g);
        elseif g < 0 && Os(i) > 0
          fprintf("    %e * sin((%e) * x + %f) + \n", lm.Coefficients{k,1},
g, Os(i));
        elseif g < 0 && Os(i) == 0
          fprintf("    %e * sin((%e) * x) + \n", lm.Coefficients{k,1}, g);
        else % both g and Os(i) are negative
```

```
          fprintf("    %e * sin((%e) * x - %f) + \n", lm.Coefficients{k,1},
g, abs(Os(i)));
        end
  else
        if g >= 0 && Os(i) > 0
          fprintf("    %e * cos(%e * x + %f) + \n", lm.Coefficients{k,1}, g,
Os(i));
        elseif g >= 0 && Os(i) < 0
          fprintf("    %e * cos(%e * x - %f) + \n", lm.Coefficients{k,1}, g,
abs(Os(i)));
        elseif g >= 0 && Os(i) == 0
          fprintf("    %e * cos(%e * x) + \n", lm.Coefficients{k,1}, g);
        elseif g < 0 && Os(i) > 0
          fprintf("    %e * cos((%e) * x + %f) + \n", lm.Coefficients{k,1},
g, Os(i));
        elseif g < 0 && Os(i) == 0
          fprintf("    %e * cos((%e) * x) + \n", lm.Coefficients{k,1}, g);
        else % both g and Os(i) are negative
          fprintf("    %e * cos((%e) * x - %f) + \n", lm.Coefficients{k,1},
g, abs(Os(i)));
        end
  end
  flag = 1 - flag;
end
fprintf("    (%e)*x + (%e)*x^2", lm.Coefficients{order+2,1},
lm.Coefficients{order+3,1});
fprintf("\n");

lstFactor = [];
lstOffset = [];
for i=1:order
  g = S(i)*gx(i,As(i),Bs(i));
  lstFactor = [lstFactor,g];
  lstOffset = [lstOffset, Os(i)];
end
fprintf('List of factors: [');
for i=1:order-1
 fprintf('%f, ', lstFactor(i));
end
fprintf('%f]\n', lstFactor(order));
fprintf('List of offsets: [');
for i=1:order-1
 fprintf('%f, ', lstOffset(i));
end
fprintf('%f]\n', lstOffset(order));
fprintf('Fitting %s in range (%f, %f)\n', sFxName, min(xdata),max(xdata));
n = length(xdata);
sumsqr = 0;
for i=1:n
  flag = 1;
  yc = lm.Coefficients{1,1};
  for j=2:order+1 % length(lm.Coefficients{:,1})
      if flag > 0
        yc = yc + lm.Coefficients{j,1} * sin(xdata(i)*lstFactor(j-1)+Os(j-1));
    else
        yc = yc + lm.Coefficients{j,1} * cos(xdata(i)*lstFactor(j-1)+Os(j-1));
    end
```

```
    flag = 1 - flag;
  end
  yc = yc + lm.Coefficients{order+2,1}*xdata(i) +
lm.Coefficients{order+3,1}*xdata(i)^2;
  sumsqr = sumsqr + (ydata(i) - yc)^2;
end
k = order + 3;
fprintf('MSS of errors squared = %e\n', sqrt(sumsqr)/n);
fprintf("R-Squared = %12.8f\n", lm.Rsquared.Ordinary);
fprintf("R-Squared Adjusted = %12.8f\n", lm.Rsquared.Adjusted);
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 order
  global ggx

  S = zeros(order,1);
  As = zeros(order,1);
  Bs = zeros(order,1);
  Os = zeros(order,1);
  S = x(1:order);
  As = x(order+1:2*order);
  Bs = x(2*order+1:3*order);
  Os = x(3*order+1:4*order);
  X = [];
  flag = 1;
  for i=1:order
      g = S(i)*ggx(i,As(i),Bs(i));
    if flag > 0
      xs = sin(xdata*g + Os(i));
    else
      xs = cos(xdata*g + Os(i));
    end
    flag = 1 - flag;
    X = [X;xs'];
  end
  xs = xdata;
  X = [X;xs'];
  xs = xdata.^2;
  X = [X;xs'];
  X = X';
  lm = fitlm(X,ydata);

  n = length(xdata);
  k = order + 3;
  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 above listing resembles the two listings before it. The code uses the variable flag to alternate between working with the sine and cosine functions.

The Driver Script

The following Matlab script goAll performs the various Fourier-Shammas series fittings for the various tested functions. All series use this script:

```
% Version 1.0.0 10/5/2020
global bUseDiary
global bDeleteIfExists
global order

bUseDiary = true;
bDeleteIfExists = false; % or false
selIdx = 0; % note a zero value will execute all the models
runNum = 1;
bShutdown = false;
order = 3;

tic;

% build lower limit values
lstC = zeros(order,1);
for i=1:order
  if i>1
    lstC(i) = (i-1)*pi;
  else
    lstC(1) = 0.1;
  end
end
lstAs = zeros(order,1);
lstBs = 0.1 + zeros(order,1);
lstOs = -2 + zeros(order,1);
% Lb = [0.1 pi  2*pi 0 0 0 0.1 0.1 0.1 -2 -2 -2];
Lb = [lstC; lstAs; lstBs; lstOs]';
% build upper limit values
lstC = zeros(order,1);
for i=1:order
  lstC(i) = i*pi;
```

```
end
lstAs = 1 + zeros(order,1);
lstBs = 3 + zeros(order,1);
lstOs = 2 + zeros(order,1);
% Ub = [pi 2*pi 3*pi 1 1 1 3 3 3 2 2 2];
Ub = [lstC; lstAs; lstBs; lstOs]';

if selIdx==0 || selIdx==1
  gx = @(i,A,B)A+B*i;

FourierShammasSeries2(@(x)acos(x),gx,[0:.01:1],Lb,Ub,runNum,"arccos(x)","arcc
os_1.txt")

FourierShammasSeries2(@(x)asin(x),gx,[0:.01:1],Lb,Ub,runNum,"arcsin(x)","arcs
in_1.txt")

FourierShammasSeries2(@(x)atan(x),gx,[0:.01:1],Lb,Ub,runNum,"arctan(x)","arct
an_1.txt")

FourierShammasSeries2(@(x)tan(x),gx,[0:.01:1],Lb,Ub,runNum,"tan(x)","tan_1.tx
t")

FourierShammasSeries2(@(x)sinh(x),gx,[0:.01:5],Lb,Ub,runNum,"sinh(x)","sinh_1
.txt")

FourierShammasSeries2(@(x)cosh(x),gx,[0:.01:5],Lb,Ub,runNum,"cosh(x)","cosh_1
.txt")

FourierShammasSeries2(@(x)tanh(x),gx,[0:.01:3],Lb,Ub,runNum,"tanh(x)","tanh_1
.txt")

FourierShammasSeries2(@(x)erf(x),gx,[0:.01:2.1],Lb,Ub,runNum,"erf(x)","erf_1.
txt")

FourierShammasSeries2(@(x)exp(x),gx,[0:.01:2],Lb,Ub,runNum,"exp(x)","exp_1.tx
t")

FourierShammasSeries2(@(x)log(x),gx,[1:.01:10],Lb,Ub,runNum,"ln(x)","ln_1.txt
")

FourierShammasSeries2(@(x)log10(x),gx,[1:.01:10],Lb,Ub,runNum,"log(x)","log_1
.txt")

FourierShammasSeries2(@(x)10.^x,gx,[0:.01:1],Lb,Ub,runNum,"10^x","pwr10_1.txt
")

FourierShammasSeries2(@(x)tinv(0.95,x),gx,[2:0.1:100],Lb,Ub,runNum,"tinv(0.95
,x)","tinv1_1.txt")

FourierShammasSeries2(@(x)tinv(0.975,x),gx,[2:0.1:100],Lb,Ub,runNum,"tinv(0.9
75,x)","tinv2_1.txt")

FourierShammasSeries2(@(x)log10(gamma(x)),gx,[2:0.1:100],Lb,Ub,runNum,"log10G
amma(x)","log10Gamma_1.txt")

FourierShammasSeries2(@(x)digamma(x),gx,[2:0.1:100],Lb,Ub,runNum,"digamma(x)"
,"digamma_1.txt")
```

```
FourierShammasSeries2(@(x)trigamma(x),gx,[1:0.1:100],Lb,Ub,runNum,"trigamma(x
)","trigamma_1.txt")

FourierShammasSeries2(@(x)besselj(0,x),gx,[2:0.1:30],Lb,Ub,runNum,"J0(x)","J0
x_1.txt")

FourierShammasSeries2(@(x)besselj(1,x),gx,[0:0.1:30],Lb,Ub,runNum,"J1(x)","J1
x_1.txt")

FourierShammasSeries2(@(x)besselj(2,x),gx,[0:0.1:30],Lb,Ub,runNum,"J2(x)","J2
x_1.txt")

FourierShammasSeries2(@(x)besselj(3,x),gx,[0:0.1:30],Lb,Ub,runNum,"J3(x)","J3
x_1.txt")

FourierShammasSeries2(@(x)besselj(4,x),gx,[0:0.1:30],Lb,Ub,runNum,"J4(x)","J4
x_1.txt")

FourierShammasSeries2(@(x)besselj(5,x),gx,[0:0.1:30],Lb,Ub,runNum,"J5(x)","J5
x_1.txt")

FourierShammasSeries2(@(x)integral(@(z)sin(z.^2),0,x),gx,[1:0.01:5],Lb,Ub,run
Num,"FresnelSine(x)","FresnelSine_1.txt")

FourierShammasSeries2(@(x)integral(@(z)cos(z.^2),0,x),gx,[0.5:0.01:5],Lb,Ub,r
unNum,"FresnelCosine(x)","FresnelCosine_1.txt")

FourierShammasSeries2(@(x)integral(@(z)sin(z)./z,0,x),gx,[1.3:0.1:20],Lb,Ub,r
unNum,"Si(x)","Si_1.txt")
  FourierShammasSeries2(@(x)0.57721566+log(x)-integral(@(z)(1-
cos(z))./z,0,x),gx,[0.5:0.1:20],Lb,Ub,runNum,"Ci(x)","CI_1.txt")

FourierShammasSeries2(@(x)asinh(x),gx,[0:0.1:100],Lb,Ub,runNum,"asinh(x)","as
inh_1.txt")

FourierShammasSeries2(@(x)acosh(x),gx,[1:0.1:100],Lb,Ub,runNum,"acosh(x)","ac
osh_1.txt")

FourierShammasSeries2(@(x)atanh(x),gx,[0:0.001:0.999],Lb,Ub,runNum,"atanh(x)"
,"atanh_1.txt")
end

if selIdx==0 || selIdx==2
  gx = @(i,A,B)A+B/i;

FourierShammasSeries2(@(x)acos(x),gx,[0:.01:1],Lb,Ub,runNum,"arccos(x)","arcc
os_2.txt")

FourierShammasSeries2(@(x)asin(x),gx,[0:.01:1],Lb,Ub,runNum,"arcsin(x)","arcs
in_2.txt")

FourierShammasSeries2(@(x)atan(x),gx,[0:.01:1],Lb,Ub,runNum,"arctan(x)","arct
an_2.txt")

FourierShammasSeries2(@(x)tan(x),gx,[0:.01:1],Lb,Ub,runNum,"tan(x)","tan_2.tx
t")
```

```
FourierShammasSeries2(@(x)sinh(x),gx,[0:.01:5],Lb,Ub,runNum,"sinh(x)","sinh_2
.txt")

FourierShammasSeries2(@(x)cosh(x),gx,[0:.01:5],Lb,Ub,runNum,"cosh(x)","cosh_2
.txt")

FourierShammasSeries2(@(x)tanh(x),gx,[0:.01:3],Lb,Ub,runNum,"tanh(x)","tanh_2
.txt")

FourierShammasSeries2(@(x)erf(x),gx,[0:.01:2.1],Lb,Ub,runNum,"erf(x)","erf_2.
txt")

FourierShammasSeries2(@(x)exp(x),gx,[0:.01:2],Lb,Ub,runNum,"exp(x)","exp_2.tx
t")

FourierShammasSeries2(@(x)log(x),gx,[1:.01:10],Lb,Ub,runNum,"ln(x)","ln_2.txt
")

FourierShammasSeries2(@(x)log10(x),gx,[1:.01:10],Lb,Ub,runNum,"log(x)","log_2
.txt")

FourierShammasSeries2(@(x)10.^x,gx,[0:.01:1],Lb,Ub,runNum,"10^x","pwr10_2.txt
")

FourierShammasSeries2(@(x)tinv(0.95,x),gx,[2:0.1:100],Lb,Ub,runNum,"tinv(0.95
,x)","tinv1_2.txt")

FourierShammasSeries2(@(x)tinv(0.975,x),gx,[2:0.1:100],Lb,Ub,runNum,"tinv(0.9
75,x)","tinv2_2.txt")

FourierShammasSeries2(@(x)log10(gamma(x)),gx,[2:0.1:100],Lb,Ub,runNum,"log10G
amma(x)","log10Gamma_2.txt")

FourierShammasSeries2(@(x)digamma(x),gx,[2:0.1:100],Lb,Ub,runNum,"digamma(x)"
,"diamma_2.txt")

FourierShammasSeries2(@(x)trigamma(x),gx,[1:0.1:100],Lb,Ub,runNum,"trigamma(x
)","trigamma_2.txt")

FourierShammasSeries2(@(x)trigamma(x),gx,[1:0.1:100],Lb,Ub,runNum,"trigamma(x
)","trigamma_2.txt")

FourierShammasSeries2(@(x)besselj(0,x),gx,[2:0.1:30],Lb,Ub,runNum,"J0(x)","J0
x_2.txt")

FourierShammasSeries2(@(x)besselj(1,x),gx,[0:0.1:30],Lb,Ub,runNum,"J1(x)","J1
x_2.txt")

FourierShammasSeries2(@(x)besselj(2,x),gx,[0:0.1:30],Lb,Ub,runNum,"J2(x)","J2
x_2.txt")

FourierShammasSeries2(@(x)besselj(3,x),gx,[0:0.1:30],Lb,Ub,runNum,"J3(x)","J3
x_2.txt")

FourierShammasSeries2(@(x)besselj(4,x),gx,[0:0.1:30],Lb,Ub,runNum,"J4(x)","J4
x_2.txt")
```

Copyright © 2020 by Namir Clement Shammas

```
FourierShammasSeries2(@(x)besselj(5,x),gx,[0:0.1:30],Lb,Ub,runNum,"J5(x)","J5
x_2.txt")

FourierShammasSeries2(@(x)integral(@(z)sin(z.^2),0,x),gx,[1:0.01:5],Lb,Ub,run
Num,"FresnelSine(x)","FresnelSine_2.txt")

FourierShammasSeries2(@(x)integral(@(z)cos(z.^2),0,x),gx,[0.5:0.01:5],Lb,Ub,r
unNum,"FresnelCosine(x)","FresnelCosine_2.txt")

FourierShammasSeries2(@(x)integral(@(z)sin(z)./z,0,x),gx,[1.3:0.1:20],Lb,Ub,r
unNum,"Si(x)","Si_2.txt")
  FourierShammasSeries2(@(x)0.57721566+log(x)-integral(@(z)(1-
cos(z))./z,0,x),gx,[0.5:0.1:20],Lb,Ub,runNum,"Ci(x)","Ci_2.txt")

FourierShammasSeries2(@(x)asinh(x),gx,[0:0.1:100],Lb,Ub,runNum,"asinh(x)","as
inh_2.txt")

FourierShammasSeries2(@(x)acosh(x),gx,[1:0.1:100],Lb,Ub,runNum,"acosh(x)","ac
osh_2.txt")

FourierShammasSeries2(@(x)atanh(x),gx,[0:0.001:0.999],Lb,Ub,runNum,"atanh(x)"
,"atanh_2.txt")
end

if selIdx==0 || selIdx==3
  gx = @(i,A,B)A+B*sqrt(i);

FourierShammasSeries2(@(x)acos(x),gx,[0:.01:1],Lb,Ub,runNum,"arccos(x)","arcc
os_3.txt")

FourierShammasSeries2(@(x)asin(x),gx,[0:.01:1],Lb,Ub,runNum,"arcsin(x)","arcs
in_3.txt")

FourierShammasSeries2(@(x)atan(x),gx,[0:.01:1],Lb,Ub,runNum,"arctan(x)","arct
an_3.txt")

FourierShammasSeries2(@(x)tan(x),gx,[0:.01:1],Lb,Ub,runNum,"tan(x)","tan_3.tx
t")

FourierShammasSeries2(@(x)sinh(x),gx,[0:.01:5],Lb,Ub,runNum,"sinh(x)","sinh_3
.txt")

FourierShammasSeries2(@(x)cosh(x),gx,[0:.01:5],Lb,Ub,runNum,"cosh(x)","cosh_3
.txt")

FourierShammasSeries2(@(x)tanh(x),gx,[0:.01:3],Lb,Ub,runNum,"tanh(x)","tanh_3
.txt")

FourierShammasSeries2(@(x)erf(x),gx,[0:.01:2.1],Lb,Ub,runNum,"erf(x)","erf_3.
txt")

FourierShammasSeries2(@(x)exp(x),gx,[0:.01:2],Lb,Ub,runNum,"exp(x)","exp_3.tx
t")

FourierShammasSeries2(@(x)log(x),gx,[1:.01:10],Lb,Ub,runNum,"ln(x)","ln_3.txt
")
```

Copyright © 2020 by Namir Clement Shammas

```
FourierShammasSeries2(@(x)log10(x),gx,[1:.01:10],Lb,Ub,runNum,"log(x)","log_3
.txt")

FourierShammasSeries2(@(x)10.^x,gx,[0:.01:1],Lb,Ub,runNum,"10^x","pwr10_3.txt
")

FourierShammasSeries2(@(x)tinv(0.95,x),gx,[2:0.1:100],Lb,Ub,runNum,"tinv(0.95
,x)","tinv1_3.txt")

FourierShammasSeries2(@(x)tinv(0.975,x),gx,[2:0.1:100],Lb,Ub,runNum,"tinv(0.9
75,x)","tinv2_3.txt")

FourierShammasSeries2(@(x)log10(gamma(x)),gx,[2:0.1:100],Lb,Ub,runNum,"log10G
amma(x)","log10Gamma_3.txt")

FourierShammasSeries2(@(x)digamma(x),gx,[2:0.1:100],Lb,Ub,runNum,"digamma(x)"
,"digamma_3.txt")

FourierShammasSeries2(@(x)trigamma(x),gx,[1:0.1:100],Lb,Ub,runNum,"trigamma(x
)","trigamma_3.txt")

FourierShammasSeries2(@(x)besselj(0,x),gx,[2:0.1:30],Lb,Ub,runNum,"J0(x)","J0
x_3.txt")

FourierShammasSeries2(@(x)besselj(1,x),gx,[0:0.1:30],Lb,Ub,runNum,"J1(x)","J1
x_3.txt")

FourierShammasSeries2(@(x)besselj(2,x),gx,[0:0.1:30],Lb,Ub,runNum,"J2(x)","J2
x_3.txt")

FourierShammasSeries2(@(x)besselj(3,x),gx,[0:0.1:30],Lb,Ub,runNum,"J3(x)","J3
x_3.txt")

FourierShammasSeries2(@(x)besselj(4,x),gx,[0:0.1:30],Lb,Ub,runNum,"J4(x)","J4
x_3.txt")

FourierShammasSeries2(@(x)besselj(5,x),gx,[0:0.1:30],Lb,Ub,runNum,"J5(x)","J5
x_3.txt")

FourierShammasSeries2(@(x)integral(@(z)sin(z.^2),0,x),gx,[1:0.01:5],Lb,Ub,run
Num,"FresnelSine(x)","FresnelSine_3.txt")

FourierShammasSeries2(@(x)integral(@(z)cos(z.^2),0,x),gx,[0.5:0.01:5],Lb,Ub,r
unNum,"FresnelCosine(x)","FresnelCosine_3.txt")

FourierShammasSeries2(@(x)integral(@(z)sin(z)./z,0,x),gx,[1.3:0.1:20],Lb,Ub,r
unNum,"Si(x)","Si_3.txt")
  FourierShammasSeries2(@(x)0.57721566+log(x)-integral(@(z)(1-
cos(z))./z,0,x),gx,[0.5:0.1:20],Lb,Ub,runNum,"Ci(x)","Ci_3.txt")

FourierShammasSeries2(@(x)asinh(x),gx,[0:0.1:100],Lb,Ub,runNum,"asinh(x)","as
inh_3.txt")

FourierShammasSeries2(@(x)acosh(x),gx,[1:0.1:100],Lb,Ub,runNum,"acosh(x)","ac
osh_3.txt")
```

Copyright © 2020 by Namir Clement Shammas

```
FourierShammasSeries2(@(x)atanh(x),gx,[0:0.001:0.999],Lb,Ub,runNum,"atanh(x)"
,"atanh_3.txt")
end

if selIdx==0 || selIdx==4
  gx = @(i,A,B)A+B*log(i)^4;

FourierShammasSeries2(@(x)acos(x),gx,[0:.01:1],Lb,Ub,runNum,"arccos(x)","arcc
os_4.txt")

FourierShammasSeries2(@(x)asin(x),gx,[0:.01:1],Lb,Ub,runNum,"arcsin(x)","arcs
in_4.txt")

FourierShammasSeries2(@(x)atan(x),gx,[0:.01:1],Lb,Ub,runNum,"arctan(x)","arct
an_4.txt")

FourierShammasSeries2(@(x)tan(x),gx,[0:.01:1],Lb,Ub,runNum,"tan(x)","tan_4.tx
t")

FourierShammasSeries2(@(x)sinh(x),gx,[0:.01:5],Lb,Ub,runNum,"sinh(x)","sinh_4
.txt")

FourierShammasSeries2(@(x)cosh(x),gx,[0:.01:5],Lb,Ub,runNum,"cosh(x)","cosh_4
.txt")

FourierShammasSeries2(@(x)tanh(x),gx,[0:.01:3],Lb,Ub,runNum,"tanh(x)","tanh_4
.txt")

FourierShammasSeries2(@(x)erf(x),gx,[0:.01:2.1],Lb,Ub,runNum,"erf(x)","erf_4.
txt")

FourierShammasSeries2(@(x)exp(x),gx,[0:.01:2],Lb,Ub,runNum,"exp(x)","exp_4.tx
t")

FourierShammasSeries2(@(x)log(x),gx,[1:.01:10],Lb,Ub,runNum,"ln(x)","ln_4.txt
")

FourierShammasSeries2(@(x)log10(x),gx,[1:.01:10],Lb,Ub,runNum,"log(x)","log_4
.txt")

FourierShammasSeries2(@(x)10.^x,gx,[0:.01:1],Lb,Ub,runNum,"10^x","pwr10_4.txt
")

FourierShammasSeries2(@(x)tinv(0.95,x),gx,[2:0.1:100],Lb,Ub,runNum,"tinv(0.95
,x)","tinv1_4.txt")

FourierShammasSeries2(@(x)tinv(0.975,x),gx,[2:0.1:100],Lb,Ub,runNum,"tinv(0.9
75,x)","tinv2_4.txt")

FourierShammasSeries2(@(x)log10(gamma(x)),gx,[2:0.1:100],Lb,Ub,runNum,"log10G
amma(x)","log10Gamma_4.txt")

FourierShammasSeries2(@(x)digamma(x),gx,[2:0.1:100],Lb,Ub,runNum,"digamma(x)"
,"digamma_4.txt")
```

```
FourierShammasSeries2(@(x)trigamma(x),gx,[1:0.1:100],Lb,Ub,runNum,"tigamma(x)
","trigamma_4.txt")

FourierShammasSeries2(@(x)besselj(0,x),gx,[2:0.1:30],Lb,Ub,runNum,"J0(x)","J0
x_4.txt")

FourierShammasSeries2(@(x)besselj(1,x),gx,[0:0.1:30],Lb,Ub,runNum,"J1(x)","J1
x_4.txt")

FourierShammasSeries2(@(x)besselj(2,x),gx,[0:0.1:30],Lb,Ub,runNum,"J2(x)","J2
x_4.txt")

FourierShammasSeries2(@(x)besselj(3,x),gx,[0:0.1:30],Lb,Ub,runNum,"J3(x)","J3
x_4.txt")

FourierShammasSeries2(@(x)besselj(4,x),gx,[0:0.1:30],Lb,Ub,runNum,"J4(x)","J4
x_4.txt")

FourierShammasSeries2(@(x)besselj(5,x),gx,[0:0.1:30],Lb,Ub,runNum,"J5(x)","J5
x_4.txt")

FourierShammasSeries2(@(x)integral(@(z)sin(z.^2),0,x),gx,[1:0.01:5],Lb,Ub,run
Num,"FresnelSine(x)","FresnelSine_4.txt")

FourierShammasSeries2(@(x)integral(@(z)cos(z.^2),0,x),gx,[0.5:0.01:5],Lb,Ub,r
unNum,"FresnelCosine(x)","FresnelCosine_4.txt")

FourierShammasSeries2(@(x)integral(@(z)sin(z)./z,0,x),gx,[1.3:0.1:20],Lb,Ub,r
unNum,"Si(x)","Si_4.txt")
  FourierShammasSeries2(@(x)0.57721566+log(x)-integral(@(z)(1-
cos(z))./z,0,x),gx,[0.5:0.1:20],Lb,Ub,runNum,"Ci(x)","CI_4.txt")

FourierShammasSeries2(@(x)asinh(x),gx,[0:0.1:100],Lb,Ub,runNum,"asinh(x)","as
inh_4.txt")

FourierShammasSeries2(@(x)acosh(x),gx,[1:0.1:100],Lb,Ub,runNum,"acosh(x)","ac
osh_4.txt")

FourierShammasSeries2(@(x)atanh(x),gx,[0:0.001:0.999],Lb,Ub,runNum,"atanh(x)"
,"atanh_4.txt")
end


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 order which is key in selecting the order of the HQFS series being used. The version of FoutrierShammasSeries2() called by the above script, determines which HQFS series is evaluated.
- The global variable bUseDiary is a Boolean flag used to tell the function FourierShammasSeries2() whether you want to copy the screen output to diary text files.
- The global variable bDeleteIfExists is a Boolean flag used to tell the function FourierShammasSeries2() whether you want to delete diary files if they exist.
- The variable selIdx allows you to select calculations for one of the seven groups (when set to the targeted group number) or all of the groups (when set to 0). I am using this scheme to reduce calculation time which can be done by working with a specific set of approximations.
- The Boolean variable bShutdown tells the Matlab script whether to shut down the computer when done.

General Comments on Results

To prevent the page count for this report from getting out of hand, I will start with presenting the main summaries that indicate the best type of HQFS series. This will be followed by the summary for each of the three sets of series (each set has 5 versions). Finally, I will present sample output text to give you an idea of how to use the curve fitting models. The code ZIP file associated with this study has folders for the 16 Fourier-Shammas series along with the Matlab files used to generate these output text files. The ZIP file will have 16 folders, each with 120 output text files.

Summary Tables

The following table gives a summary for the performance of the various HQFS series. The table shows the count of the number of models with the Adjusted R^2 of 1 and the count of the same statistic with values between 0.9999 and 1 (excluded). The last column calculates a weighted value using:

Wt = count_R2_Adj_1 + 0.8 * count_R2_Adj_0.9999

My goal is to pick the series and order with the maximum weighted sum. The results answer the following two questions:

Copyright © 2020 by Namir Clement Shammas

- What is the best HQFS series?
- What is the best order? Could lower orders excel in offering better model fittings?!!

Series	***Order***	***# Rsqr Adj = 1***	***# Rsqr Adj > 0.9999 And < 1***	***Weighted Sum***
Sine	3	18	26	38.8
Sine	4	26	19	41.2
Sine	5	27	18	41.4
Sine	6	27	18	41.4
Sine	7	28	19	43.2
Cosine	3	17	27	38.6
Cosine	4	25	20	41.0
Cosine	5	26	20	42.0
Cosine	6	26	21	42.8
Cosine	7	29	19	44.2
Sine/Cosine	3	18	26	38.8
Sine/Cosine	4	25	19	40.2
Sine/Cosine	5	28	17	41.6
Sine/Cosine	6	26	20	42.0
Sine/Cosine	7	30	17	43.6
Cosine/Sine	7	25	22	42.6

The above table shows that the best HQFS series is the cosine series with order 7. The second rank goes to the 7th order since/cosine series. The third rank goes to the 7th order sine series.

The statistics in the above table (and other statistics, like the mean square root of errors squared and AICc) are tabulated in the next subsections, given for each series and order.

The tables in the next subsections provide you with summaries for the performances of the various Hybrid Quadratic Fourier-Shammas series and also offer you catalogs for the detailed results found in the output text files that you download in the accompanying ZIP file. Each subsection tells you what folder to look in and the general formats for the output text files. I am resorting to this approach to avoid ending up with a study that has a huge and intimidating page count. The ZIP file also contains the file *Hybrid Fourier-Shammas Series Maps.pdf*

Copyright © 2020 by Namir Clement Shammas

which has tables that guide you to easily select output text files to view. The map file has tables for the various series with the following columns:

- Filename: indicates the output file name.
- Function: indicates the approximated function.
- The gx(i,A,B) expression.
- The Rsquare adjusted value.

Sine Series of Order 3

The next table shows a summary of results for the Sine series of the order 3:

$Y = a_0 + a_1 * \sin(S_1 * gx(1,A_1,B_1) + Os_1) + a_2 * \sin(S_2 * gx(2,A_2,B_2) + Os_2) +$

$a_3 * \sin(S_3 * gx(3,A_3,B_3) + Os_1) + a_4*x + a_5*x^2$

The output text files for this series are located in the following folder:

Fourier-Shammas Series Approximations Quadratic Fit with 3 Sine

The files are named using the following general format:

*fxName_n*_3_sin_run1.txt

Where fxName is the function name and n is the gx series number which varies between 1 and 4. Each output files specifies the function being approximated and the gx series being used. For example, the file for the inverse hyperboplic cosine (acosh) using the first gx series is named acosh_1_3_sin_run1.txt.

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
10^x	A+B*i	1.961221E-06	1	-1.877026E+03
10^x	A+B/i	3.024769E-07	1	-2.254740E+03
10^x	A+B*sqrt(i)	1.173058E-06	1	-1.980845E+03
10^x	A+B*log(i)^4	1.829026E-06	1	-1.891124E+03
acosh(x)	A+B*i	2.019493E-03	0.99499394	-2.625082E+03
acosh(x)	A+B/i	2.747443E-03	0.9907345	-2.014974E+03
acosh(x)	A+B*sqrt(i)	3.399799E-03	0.98581211	-1.592718E+03
acosh(x)	A+B*log(i)^4	2.328876E-03	0.99334261	-2.342569E+03
arccos(x)	A+B*i	9.108337E-04	0.99940131	-6.365860E+02
arccos(x)	A+B/i	8.601062E-04	0.99946614	-6.481615E+02
arccos(x)	A+B*sqrt(i)	8.578461E-04	0.99946894	-6.486930E+02
arccos(x)	A+B*log(i)^4	7.800961E-04	0.99956084	-6.678845E+02
arcsin(x)	A+B*i	8.161637E-04	0.99951929	-6.587545E+02
arcsin(x)	A+B/i	8.518640E-04	0.99947632	-6.501065E+02
arcsin(x)	A+B*sqrt(i)	8.051124E-04	0.99953222	-6.615084E+02
arcsin(x)	A+B*log(i)^4	8.264813E-04	0.99950706	-6.562169E+02
arctan(x)	A+B*i	6.830721E-09	1	-3.020321E+03
arctan(x)	A+B/i	1.357283E-08	1	-2.881623E+03
arctan(x)	A+B*sqrt(i)	1.973737E-09	1	-3.271093E+03
arctan(x)	A+B*log(i)^4	4.252460E-09	1	-3.116059E+03
asinh(x)	A+B*i	2.380010E-03	0.99382433	-2.312919E+03
asinh(x)	A+B/i	2.969241E-03	0.99038792	-1.870074E+03
asinh(x)	A+B*sqrt(i)	3.809782E-03	0.98417562	-1.371044E+03
asinh(x)	A+B*log(i)^4	2.258294E-03	0.99443984	-2.418015E+03
atanh(x)	A+B*i	1.626600E-03	0.99204005	-3.072810E+03
atanh(x)	A+B/i	1.582238E-03	0.9924683	-3.128113E+03
atanh(x)	A+B*sqrt(i)	1.598525E-03	0.99231245	-3.107631E+03
atanh(x)	A+B*log(i)^4	1.572855E-03	0.99255737	-3.140008E+03
Ci(x)	A+B*i	2.176824E-03	0.95335631	-7.877375E+02
Ci(x)	A+B/i	3.618318E-03	0.8711276	-5.885456E+02
Ci(x)	A+B*sqrt(i)	2.480202E-03	0.93944913	-7.365920E+02
Ci(x)	A+B*log(i)^4	2.920459E-03	0.91604459	-6.725388E+02
cosh(x)	A+B*i	2.248860E-04	0.99999992	-3.856251E+03
cosh(x)	A+B/i	4.051606E-04	0.99999975	-3.266385E+03
cosh(x)	A+B*sqrt(i)	1.483489E-04	0.99999997	-4.273109E+03
cosh(x)	A+B*log(i)^4	1.227690E-04	0.99999998	-4.462751E+03
digamma(x)	A+B/i	1.912118E-03	0.99506418	-2.715491E+03
digamma(x)	A+B*i	1.088495E-03	0.99840051	-3.820913E+03
digamma(x)	A+B*sqrt(i)	2.221458E-03	0.99333798	-2.421285E+03
digamma(x)	A+B*log(i)^4	1.166317E-03	0.99816362	-3.685427E+03
erf(x)	A+B*i	6.247409E-07	1	-4.276216E+03
erf(x)	A+B/i	7.224226E-07	1	-4.214911E+03
erf(x)	A+B*sqrt(i)	2.708977E-07	1	-4.628838E+03
erf(x)	A+B*log(i)^4	4.255181E-07	1	-4.438277E+03
exp(x)	A+B*i	1.939552E-07	1	-4.552356E+03
exp(x)	A+B/i	3.447958E-07	1	-4.321075E+03
exp(x)	A+B*sqrt(i)	6.864568E-08	1	-4.969901E+03
exp(x)	A+B*log(i)^4	3.351072E-07	1	-4.332548E+03
FresnelCosine(x)	A+B*i	1.919001E-03	0.93002619	-1.582524E+03

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
FresnelCosine(x)	A+B/i	1.919001E-03	0.93002616	-1.582524E+03
FresnelCosine(x)	A+B*sqrt(i)	1.798004E-03	0.93857196	-1.641269E+03
FresnelCosine(x)	A+B*log(i)^4	1.978211E-03	0.92564156	-1.555114E+03
FresnelSine(x)	A+B*i	2.040370E-03	0.90795058	-1.402308E+03
FresnelSine(x)	A+B/i	2.361851E-03	0.87665878	-1.284964E+03
FresnelSine(x)	A+B*sqrt(i)	1.775433E-03	0.93030338	-1.513856E+03
FresnelSine(x)	A+B*log(i)^4	1.790235E-03	0.92913637	-1.507197E+03
J0(x)	A+B*i	1.361825E-03	0.98122849	-1.302471E+03
J0(x)	A+B/i	1.227280E-03	0.98475443	-1.360933E+03
J0(x)	A+B*sqrt(i)	1.450870E-03	0.97869343	-1.266875E+03
J0(x)	A+B*log(i)^4	1.258256E-03	0.98397512	-1.346924E+03
J1(x)	A+B*i	1.320024E-03	0.98714122	-1.394996E+03
J1(x)	A+B/i	8.148714E-04	0.9950998	-1.685386E+03
J1(x)	A+B*sqrt(i)	1.975979E-03	0.9711862	-1.152141E+03
J1(x)	A+B*log(i)^4	4.682625E-03	0.83818651	-6.327387E+02
J2(x)	A+B*i	8.464409E-04	0.99366596	-1.662504E+03
J2(x)	A+B/i	9.288968E-04	0.9923718	-1.606544E+03
J2(x)	A+B*sqrt(i)	2.573317E-03	0.94145713	-9.931339E+02
J2(x)	A+B*log(i)^4	8.876620E-04	0.99303401	-1.633879E+03
J3(x)	A+B*i	1.623573E-03	0.97356151	-1.270395E+03
J3(x)	A+B/i	1.486093E-03	0.97784941	-1.323659E+03
J3(x)	A+B*sqrt(i)	2.531496E-03	0.93572416	-1.002998E+03
J3(x)	A+B*log(i)^4	1.557208E-03	0.97567872	-1.295519E+03
J4(x)	A+B*i	1.547241E-03	0.97370957	-1.299385E+03
J4(x)	A+B/i	1.847565E-03	0.96251294	-1.192593E+03
J4(x)	A+B*sqrt(i)	2.718452E-03	0.91884315	-9.601042E+02
J4(x)	A+B*log(i)^4	1.203996E-03	0.98408041	-1.450383E+03
J5(x)	A+B*i	3.557082E-03	0.84493538	-7.982396E+02
J5(x)	A+B/i	3.555954E-03	0.84503374	-7.984306E+02
J5(x)	A+B*sqrt(i)	3.607403E-03	0.84051706	-7.897830E+02
J5(x)	A+B*log(i)^4	1.857666E-03	0.95770781	-1.189311E+03
ln(x)	A+B*i	5.863275E-05	0.99999102	-8.848100E+03
ln(x)	A+B/i	3.339986E-05	0.99999708	-9.862160E+03
ln(x)	A+B*sqrt(i)	7.482010E-05	0.99998537	-8.408785E+03
ln(x)	A+B*log(i)^4	4.856519E-05	0.99999384	-9.187572E+03
log(x)	A+B*i	3.567956E-05	0.99998236	-9.743181E+03
log(x)	A+B/i	2.409133E-05	0.99999196	-1.045087E+04
log(x)	A+B*sqrt(i)	2.516628E-05	0.99999123	-1.037221E+04
log(x)	A+B*log(i)^4	2.056740E-05	0.99999414	-1.073585E+04
log10Gamma(x)	A+B*i	1.844671E-03	0.99999847	-2.785948E+03
log10Gamma(x)	A+B/i	3.660128E-03	0.99999398	-1.441592E+03
log10Gamma(x)	A+B*sqrt(i)	4.535860E-03	0.99999075	-1.020710E+03
log10Gamma(x)	A+B*log(i)^4	1.894015E-03	0.99999839	-2.734156E+03
Si(x)	A+B*i	2.736585E-03	0.87859939	-6.763567E+02
Si(x)	A+B/i	1.346483E-03	0.97060967	-9.430216E+02
Si(x)	A+B*sqrt(i)	2.750956E-03	0.877321	-6.743873E+02
Si(x)	A+B*log(i)^4	1.618901E-03	0.95751421	-8.737428E+02
sinh(x)	A+B*i	2.582343E-04	0.9999999	-3.717700E+03
sinh(x)	A+B/i	1.725765E-04	0.99999996	-4.121533E+03

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
sinh(x)	A+B*sqrt(i)	1.490198E-04	0.99999997	-4.268588E+03
sinh(x)	A+B*log(i)^4	1.272009E-04	0.99999998	-4.427217E+03
tan(x)	A+B*i	2.702921E-06	1	-1.812231E+03
tan(x)	A+B/i	2.990614E-06	0.99999999	-1.791800E+03
tan(x)	A+B*sqrt(i)	3.429800E-06	0.99999999	-1.764121E+03
tan(x)	A+B*log(i)^4	2.980440E-06	0.99999999	-1.792488E+03
tanh(x)	A+B*i	3.083020E-06	0.99999996	-5.042813E+03
tanh(x)	A+B/i	4.609790E-06	0.99999992	-4.800645E+03
tanh(x)	A+B*sqrt(i)	9.731102E-07	1	-5.737020E+03
tanh(x)	A+B*log(i)^4	4.273537E-06	0.99999993	-4.846240E+03
tinv(0.95,x)	A+B*i	1.796438E-03	0.80217748	-2.837932E+03
tinv(0.95,x)	A+B/i	1.676375E-03	0.82773639	-2.973647E+03
tinv(0.95,x)	A+B*sqrt(i)	1.777720E-03	0.80627845	-2.858482E+03
tinv(0.95,x)	A+B*log(i)^4	1.286069E-03	0.89861354	-3.493662E+03
tinv(0.975,x)	A+B*i	3.354094E-03	0.77284033	-1.612906E+03
tinv(0.975,x)	A+B/i	3.230259E-03	0.78930447	-1.686715E+03
tinv(0.975,x)	A+B*sqrt(i)	3.316051E-03	0.77796415	-1.635287E+03
tinv(0.975,x)	A+B*log(i)^4	2.475632E-03	0.87624781	-2.208738E+03
trigamma(x)	A+B*i	1.810782E-03	0.79508728	-2.841294E+03
trigamma(x)	A+B/i	2.153676E-03	0.71013397	-2.497581E+03
trigamma(x)	A+B*sqrt(i)	2.307676E-03	0.66719766	-2.360694E+03
trigamma(x)	A+B*log(i)^4	1.923774E-03	0.76871635	-2.721323E+03

Sine Series of Order 4

The next table shows a summary of results for the Sine series of the order 4:

$$Y = a_0 + a_1 * \sin(S_1 * gx(1,A_1,B_1) + Os_1) + a_2 * \sin(S_2 * gx(2,A_2,B_2) + Os_2) +$$
$$a_3 * \sin(S_3 * gx(3,A_3,B_3) + Os_3) + a_4 * \sin(S_4 * gx(4,A_4,B_4) + Os_4) +$$
$$a_5*x + a_6*x^2$$

The output text files for this series are located in the following folder:

Fourier-Shammas Series Approximations Quadratic Fit with 4 Sine

The files are named using the following general format:

*fxName_n_*4_sin_run1.txt

Where fxName is the function name and n is the gx series number which varies between 1 and 4. Each output files specifies the function being approximated and the gx series being used. For example, the file for the inverse hyperboplic cosine (acosh) using the first gx series is named acosh_1_4_sin_run1.txt.

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
10^x	A+B*i	6.235307E-08	1	-2.569314E+03
10^x	A+B/i	9.102884E-08	1	-2.492885E+03
10^x	A+B*sqrt(i)	9.915323E-08	1	-2.475616E+03
10^x	A+B*log(i)^4	1.063771E-07	1	-2.461411E+03
acosh(x)	A+B*i	2.060284E-03	0.99478437	-2.581419E+03
acosh(x)	A+B/i	2.917320E-03	0.98954267	-1.892036E+03
acosh(x)	A+B*sqrt(i)	2.201120E-03	0.99404694	-2.450363E+03
acosh(x)	A+B*log(i)^4	2.177582E-03	0.99417358	-2.471672E+03
arccos(x)	A+B*i	6.782165E-04	0.99966453	-6.918438E+02
arccos(x)	A+B/i	6.987993E-04	0.99964386	-6.858046E+02
arccos(x)	A+B*sqrt(i)	6.802225E-04	0.99966254	-6.912472E+02
arccos(x)	A+B*log(i)^4	6.796077E-04	0.99966315	-6.914299E+02
arcsin(x)	A+B*i	6.723201E-04	0.99967033	-6.936077E+02
arcsin(x)	A+B/i	6.823885E-04	0.99966039	-6.906050E+02
arcsin(x)	A+B*sqrt(i)	7.401500E-04	0.99960046	-6.741918E+02
arcsin(x)	A+B*log(i)^4	6.793737E-04	0.99966338	-6.914995E+02
arctan(x)	A+B*i	1.999658E-09	1	-3.264064E+03
arctan(x)	A+B/i	1.723032E-09	1	-3.293016E+03
arctan(x)	A+B*sqrt(i)	1.646708E-09	1	-3.303353E+03
arctan(x)	A+B*log(i)^4	1.766266E-09	1	-3.289230E+03
asinh(x)	A+B*i	2.269581E-03	0.99437847	-2.404005E+03
asinh(x)	A+B/i	2.715296E-03	0.99195368	-2.045035E+03
asinh(x)	A+B*sqrt(i)	3.899329E-03	0.98340632	-1.320504E+03
asinh(x)	A+B*log(i)^4	1.335867E-03	0.99805245	-3.465095E+03
atanh(x)	A+B*i	1.365155E-03	0.99438758	-3.419230E+03
atanh(x)	A+B/i	1.372181E-03	0.99432966	-3.408963E+03
atanh(x)	A+B*sqrt(i)	1.364758E-03	0.99439085	-3.419811E+03
atanh(x)	A+B*log(i)^4	1.434412E-03	0.99380368	-3.320255E+03
Ci(x)	A+B*i	3.550330E-03	0.87526869	-5.918301E+02
Ci(x)	A+B/i	2.721772E-03	0.92669362	-6.960069E+02
Ci(x)	A+B*sqrt(i)	3.462533E-03	0.88136143	-6.016458E+02
Ci(x)	A+B*log(i)^4	2.152257E-03	0.95416191	-7.880352E+02
cosh(x)	A+B*i	2.408836E-04	0.99999991	-3.783336E+03
cosh(x)	A+B/i	4.965583E-05	1	-5.365692E+03
cosh(x)	A+B*sqrt(i)	2.159035E-05	1	-6.200227E+03
cosh(x)	A+B*log(i)^4	1.198151E-04	0.99999998	-4.483097E+03
digamma(x)	A+B/i	2.027385E-03	0.99444546	-2.596617E+03
digamma(x)	A+B*i	1.114017E-03	0.9983229	-3.771412E+03
digamma(x)	A+B*sqrt(i)	1.220005E-03	0.9979886	-3.593101E+03
digamma(x)	A+B*log(i)^4	1.166035E-03	0.99816262	-3.681873E+03
erf(x)	A+B*i	1.769104E-08	1	-5.776202E+03
erf(x)	A+B/i	3.090221E-08	1	-5.540823E+03
erf(x)	A+B*sqrt(i)	3.170499E-08	1	-5.530000E+03
erf(x)	A+B*log(i)^4	5.078563E-08	1	-5.331179E+03
exp(x)	A+B*i	4.896778E-08	1	-5.101548E+03
exp(x)	A+B/i	1.548776E-08	1	-5.564269E+03
exp(x)	A+B*sqrt(i)	1.684435E-08	1	-5.530534E+03
exp(x)	A+B*log(i)^4	3.253579E-08	1	-5.265894E+03
FresnelCosine(x)	A+B*i	1.517017E-03	0.95617283	-1.790483E+03

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
FresnelCosine(x)	A+B/i	1.026114E-03	0.97994821	-2.143136E+03
FresnelCosine(x)	A+B*sqrt(i)	1.632517E-03	0.94924514	-1.724297E+03
FresnelCosine(x)	A+B*log(i)^4	1.502303E-03	0.9570189	-1.799275E+03
FresnelSine(x)	A+B*i	1.811089E-03	0.92729179	-1.493837E+03
FresnelSine(x)	A+B/i	8.895865E-04	0.98245795	-2.064000E+03
FresnelSine(x)	A+B*sqrt(i)	2.083224E-03	0.90379983	-1.381566E+03
FresnelSine(x)	A+B*log(i)^4	1.724332E-03	0.93409087	-1.533206E+03
J0(x)	A+B*i	1.115734E-03	0.98735381	-1.410381E+03
J0(x)	A+B/i	9.687584E-04	0.99046612	-1.489765E+03
J0(x)	A+B*sqrt(i)	7.931604E-04	0.99360912	-1.602159E+03
J0(x)	A+B*log(i)^4	1.112887E-03	0.98741826	-1.411817E+03
J1(x)	A+B*i	8.292737E-04	0.99490779	-1.670743E+03
J1(x)	A+B/i	1.601043E-04	0.99981019	-2.660867E+03
J1(x)	A+B*sqrt(i)	2.174221E-03	0.96499596	-1.090490E+03
J1(x)	A+B*log(i)^4	1.374659E-03	0.98600732	-1.366485E+03
J2(x)	A+B*i	1.530817E-04	0.99979212	-2.687869E+03
J2(x)	A+B/i	1.036174E-04	0.99990476	-2.922809E+03
J2(x)	A+B*sqrt(i)	2.603775E-03	0.93985923	-9.819539E+02
J2(x)	A+B*log(i)^4	1.727455E-04	0.99973529	-2.615119E+03
J3(x)	A+B*i	1.543074E-03	0.97603698	-1.296912E+03
J3(x)	A+B/i	4.053431E-04	0.99834646	-2.101664E+03
J3(x)	A+B*sqrt(i)	1.819902E-03	0.96666781	-1.197578E+03
J3(x)	A+B*log(i)^4	1.469173E-03	0.97827729	-1.326456E+03
J4(x)	A+B*i	2.684353E-03	0.92059717	-9.636064E+02
J4(x)	A+B/i	8.194005E-04	0.99260141	-1.677953E+03
J4(x)	A+B*sqrt(i)	2.718320E-03	0.91857501	-9.560368E+02
J4(x)	A+B*log(i)^4	1.563764E-03	0.9730537	-1.288894E+03
J5(x)	A+B*i	7.127436E-04	0.99375308	-1.761903E+03
J5(x)	A+B/i	2.565527E-03	0.91906192	-9.908625E+02
J5(x)	A+B*sqrt(i)	2.267873E-03	0.93675341	-1.065102E+03
J5(x)	A+B*log(i)^4	4.080470E-03	0.79525184	-7.115055E+02
ln(x)	A+B*i	9.780564E-05	0.99997497	-7.922005E+03
ln(x)	A+B/i	1.998936E-05	0.99999895	-1.078319E+04
ln(x)	A+B*sqrt(i)	3.637212E-05	0.99999654	-9.704507E+03
ln(x)	A+B*log(i)^4	4.437654E-05	0.99999485	-9.346074E+03
log(x)	A+B*i	3.301847E-05	0.99998488	-9.878824E+03
log(x)	A+B/i	6.471828E-06	0.99999942	-1.281538E+04
log(x)	A+B*sqrt(i)	7.826332E-06	0.99999915	-1.247294E+04
log(x)	A+B*log(i)^4	3.064465E-05	0.99998697	-1.001327E+04
log10Gamma(x)	A+B*i	1.501258E-03	0.99999899	-3.186086E+03
log10Gamma(x)	A+B/i	1.887362E-03	0.9999984	-2.737030E+03
log10Gamma(x)	A+B*sqrt(i)	2.042684E-03	0.99999812	-2.581866E+03
log10Gamma(x)	A+B*log(i)^4	8.117799E-04	0.9999997	-4.392382E+03
Si(x)	A+B*i	1.834888E-03	0.94511992	-8.224961E+02
Si(x)	A+B/i	1.714056E-03	0.95210987	-8.481094E+02
Si(x)	A+B*sqrt(i)	1.862554E-03	0.9434525	-8.168691E+02
Si(x)	A+B*log(i)^4	8.141184E-04	0.98919634	-1.128046E+03
sinh(x)	A+B*i	9.221990E-05	0.99999999	-4.745394E+03
sinh(x)	A+B/i	3.035648E-05	1	-5.858782E+03

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
sinh(x)	A+B*sqrt(i)	1.565458E-04	0.99999996	-4.215163E+03
sinh(x)	A+B*log(i)^4	1.150842E-04	0.99999998	-4.523463E+03
tan(x)	A+B*i	8.046373E-07	1	-2.052684E+03
tan(x)	A+B/i	1.457051E-06	1	-1.932740E+03
tan(x)	A+B*sqrt(i)	9.105559E-07	1	-2.027703E+03
tan(x)	A+B*log(i)^4	6.987582E-07	1	-2.081183E+03
tanh(x)	A+B*i	4.479515E-07	1	-6.199962E+03
tanh(x)	A+B/i	1.103394E-06	1	-5.657283E+03
tanh(x)	A+B*sqrt(i)	9.804160E-07	1	-5.728420E+03
tanh(x)	A+B*log(i)^4	1.120743E-06	0.99999999	-5.647891E+03
tinv(0.95,x)	A+B*i	1.311200E-03	0.89450422	-3.451663E+03
tinv(0.95,x)	A+B/i	1.593182E-03	0.84425022	-3.069486E+03
tinv(0.95,x)	A+B*sqrt(i)	1.259362E-03	0.90268097	-3.530807E+03
tinv(0.95,x)	A+B*log(i)^4	1.136296E-03	0.9207718	-3.732562E+03
tinv(0.975,x)	A+B*i	2.735784E-03	0.84871704	-2.008661E+03
tinv(0.975,x)	A+B/i	2.758592E-03	0.84618412	-1.992373E+03
tinv(0.975,x)	A+B*sqrt(i)	2.416005E-03	0.88201635	-2.252544E+03
tinv(0.975,x)	A+B*log(i)^4	2.169561E-03	0.90485848	-2.463636E+03
trigamma(x)	A+B*i	1.798893E-03	0.79756374	-2.850321E+03
trigamma(x)	A+B/i	1.916149E-03	0.7703131	-2.725166E+03
trigamma(x)	A+B*sqrt(i)	1.989534E-03	0.75238287	-2.650676E+03
trigamma(x)	A+B*log(i)^4	1.676369E-03	0.82420068	-2.990134E+03

Sine Series of Order 5

The next table shows a summary of results for the Sine series of the order 5:

$$Y = a_0 + a_1 * \sin(S_1 * gx(1,A_1,B_1) + Os_1) + \ldots$$

$$+ a_5 * \sin(S_5 * gx(5,A_5,B_5) + Os_5) + a_6*x + a_7*x^2$$

The output text files for this series are located in the following folder:

Fourier-Shammas Series Approximations Quadratic Fit with 5 Sine

The files are named using the following general format:

*fxName_n*_5_sin_run1.txt

Where fxName is the function name and n is the gx series number which varies between 1 and 4. Each output files specifies the function being approximated and the gx series being used. For example, the file for the inverse hyperboplic cosine (acosh) using the first gx series is named acosh_1_5_sin_run1.txt.

Copyright © 2020 by Namir Clement Shammas

Function	*gx(i,A,B)*	*MSSE*	*RsqrAdj*	*AICc*
10^x	A+B*i	7.674431E-09	1	-Inf
10^x	A+B/i	9.305203E-09	1	-Inf
10^x	A+B*sqrt(i)	8.573768E-09	1	-Inf
10^x	A+B*log(i)^4	3.449445E-08	1	-2.684534E+03
acosh(x)	A+B*i	1.989864E-03	0.99512987	-2.646315E+03
acosh(x)	A+B/i	2.869772E-03	0.98987048	-1.920573E+03
acosh(x)	A+B*sqrt(i)	1.865686E-03	0.99571874	-2.774030E+03
acosh(x)	A+B*log(i)^4	1.730362E-03	0.99631729	-2.923271E+03
arccos(x)	A+B*i	5.620229E-04	0.99976715	-7.254436E+02
arccos(x)	A+B/i	6.263206E-04	0.99971083	-7.035630E+02
arccos(x)	A+B*sqrt(i)	5.813747E-04	0.99975084	-7.186053E+02
arccos(x)	A+B*log(i)^4	5.803930E-04	0.99975168	-7.189467E+02
arcsin(x)	A+B*i	5.622833E-04	0.99976694	-7.253500E+02
arcsin(x)	A+B/i	5.539665E-04	0.99977378	-7.283601E+02
arcsin(x)	A+B*sqrt(i)	6.049518E-04	0.99973022	-7.105751E+02
arcsin(x)	A+B*log(i)^4	5.729036E-04	0.99975805	-7.215702E+02
arctan(x)	A+B*i	4.584554E-10	1	-Inf
arctan(x)	A+B/i	8.110631E-10	1	-3.432764E+03
arctan(x)	A+B*sqrt(i)	1.109267E-09	1	-3.373392E+03
arctan(x)	A+B*log(i)^4	4.836684E-10	1	-3.539472E+03
asinh(x)	A+B*i	1.294186E-03	0.99817024	-3.524522E+03
asinh(x)	A+B/i	2.976385E-03	0.99032216	-1.857202E+03
asinh(x)	A+B*sqrt(i)	2.866506E-03	0.99102352	-1.932508E+03
asinh(x)	A+B*log(i)^4	2.255217E-03	0.99444381	-2.412683E+03
atanh(x)	A+B*i	1.176727E-03	0.99582578	-3.712259E+03
atanh(x)	A+B/i	1.146619E-03	0.99603665	-3.764098E+03
atanh(x)	A+B*sqrt(i)	1.199581E-03	0.99566207	-3.673789E+03
atanh(x)	A+B*log(i)^4	1.158811E-03	0.99595192	-3.742944E+03
Ci(x)	A+B*i	1.807766E-03	0.96748929	-8.522357E+02
Ci(x)	A+B/i	1.409478E-03	0.98023673	-9.497935E+02
Ci(x)	A+B*sqrt(i)	2.138890E-03	0.95448873	-7.863032E+02
Ci(x)	A+B*log(i)^4	9.762164E-04	0.99051943	-1.093771E+03
cosh(x)	A+B*i	8.873198E-04	0.9999988	-2.472771E+03
cosh(x)	A+B/i	2.844924E-06	1	-8.226926E+03
cosh(x)	A+B*sqrt(i)	1.690529E-05	1	-6.441271E+03
cosh(x)	A+B*log(i)^4	8.362931E-05	0.99999999	-4.839306E+03
digamma(x)	A+B/i	1.754377E-03	0.99583642	-2.876354E+03
digamma(x)	A+B*i	4.794658E-04	0.99968902	-5.421453E+03
digamma(x)	A+B*sqrt(i)	1.077368E-03	0.99842982	-3.833010E+03
digamma(x)	A+B*log(i)^4	7.771999E-04	0.99918288	-4.473758E+03
erf(x)	A+B*i	1.605078E-09	1	-6.784784E+03
erf(x)	A+B/i	2.269142E-08	1	-5.666993E+03
erf(x)	A+B*sqrt(i)	8.294195E-09	1	-6.091706E+03
erf(x)	A+B*log(i)^4	4.242239E-08	1	-5.402952E+03
exp(x)	A+B*i	3.938164E-08	1	-5.184959E+03
exp(x)	A+B/i	2.400143E-09	1	-Inf
exp(x)	A+B*sqrt(i)	3.795787E-09	1	-Inf
exp(x)	A+B*log(i)^4	2.845730E-08	1	-5.315567E+03
FresnelCosine(x)	A+B*i	1.654225E-03	0.94776869	-1.708309E+03

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
FresnelCosine(x)	A+B/i	4.270073E-04	0.99651974	-2.929876E+03
FresnelCosine(x)	A+B*sqrt(i)	1.883903E-03	0.93225788	-1.591037E+03
FresnelCosine(x)	A+B*log(i)^4	2.086535E-03	0.91690157	-1.498890E+03
FresnelSine(x)	A+B*i	4.379180E-04	0.9957382	-2.628316E+03
FresnelSine(x)	A+B/i	1.504043E-03	0.94972773	-1.638743E+03
FresnelSine(x)	A+B*sqrt(i)	1.234726E-03	0.96611958	-1.796984E+03
FresnelSine(x)	A+B*log(i)^4	1.694821E-03	0.9361655	-1.542968E+03
J0(x)	A+B*i	8.426776E-04	0.99275982	-1.564006E+03
J0(x)	A+B/i	1.087411E-03	0.98794369	-1.420712E+03
J0(x)	A+B*sqrt(i)	1.373551E-03	0.98076393	-1.289429E+03
J0(x)	A+B*log(i)^4	2.694285E-04	0.99925986	-2.204844E+03
J1(x)	A+B*i	1.278347E-03	0.98785806	-1.406102E+03
J1(x)	A+B/i	1.194684E-03	0.98939534	-1.446849E+03
J1(x)	A+B*sqrt(i)	2.262427E-03	0.96196886	-1.062439E+03
J1(x)	A+B*log(i)^4	1.494856E-03	0.98339692	-1.311913E+03
J2(x)	A+B*i	8.583691E-04	0.99344172	-1.645872E+03
J2(x)	A+B/i	9.298948E-04	0.99230321	-1.597690E+03
J2(x)	A+B*sqrt(i)	1.570586E-03	0.9780434	-1.282162E+03
J2(x)	A+B*log(i)^4	8.407969E-05	0.99993707	-3.044480E+03
J3(x)	A+B*i	5.023365E-04	0.99745178	-1.968402E+03
J3(x)	A+B/i	4.494516E-04	0.99796008	-2.035370E+03
J3(x)	A+B*sqrt(i)	3.330863E-04	0.99887963	-2.215745E+03
J3(x)	A+B*log(i)^4	1.458474E-03	0.97851949	-1.326745E+03
J4(x)	A+B*i	1.316490E-03	0.98083666	-1.388403E+03
J4(x)	A+B/i	1.062419E-04	0.9998752	-2.903640E+03
J4(x)	A+B*sqrt(i)	4.359082E-04	0.997899	-2.053789E+03
J4(x)	A+B*log(i)^4	2.891987E-04	0.99907524	-2.300800E+03
J5(x)	A+B*i	3.777198E-03	0.82395691	-7.538868E+02
J5(x)	A+B/i	1.757764E-04	0.99961876	-2.600537E+03
J5(x)	A+B*sqrt(i)	1.526605E-03	0.97124371	-1.299260E+03
J5(x)	A+B*log(i)^4	3.574508E-03	0.8423435	-7.870902E+02
ln(x)	A+B*i	7.325581E-05	0.99998595	-8.438792E+03
ln(x)	A+B/i	8.376129E-06	0.99999982	-1.234656E+04
ln(x)	A+B*sqrt(i)	4.544527E-05	0.99999459	-9.299154E+03
ln(x)	A+B*log(i)^4	3.966708E-05	0.99999588	-9.544203E+03
log(x)	A+B*i	2.368697E-05	0.99999221	-1.047331E+04
log(x)	A+B/i	3.031459E-06	0.99999987	-1.417801E+04
log(x)	A+B*sqrt(i)	1.211252E-05	0.99999796	-1.168188E+04
log(x)	A+B*log(i)^4	2.085274E-05	0.99999396	-1.070295E+04
log10Gamma(x)	A+B*i	6.747307E-04	0.99999979	-4.751153E+03
log10Gamma(x)	A+B/i	3.487980E-03	0.99999452	-1.528050E+03
log10Gamma(x)	A+B*sqrt(i)	1.277511E-03	0.99999926	-3.498699E+03
log10Gamma(x)	A+B*log(i)^4	8.891597E-04	0.99999964	-4.209713E+03
Si(x)	A+B*i	1.398378E-03	0.9679483	-9.204619E+02
Si(x)	A+B/i	1.532764E-03	0.96149188	-8.859603E+02
Si(x)	A+B*sqrt(i)	8.582529E-04	0.98792652	-1.104014E+03
Si(x)	A+B*log(i)^4	1.784538E-03	0.94780205	-8.287756E+02
sinh(x)	A+B*i	5.754543E-04	0.9999995	-2.906683E+03
sinh(x)	A+B/i	6.341362E-06	1	-7.423767E+03

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
sinh(x)	A+B*sqrt(i)	2.775936E-05	1	-5.944333E+03
sinh(x)	A+B*log(i)^4	1.676656E-04	0.99999996	-4.142338E+03
tan(x)	A+B*i	2.598626E-07	1	-2.276631E+03
tan(x)	A+B/i	1.879908E-07	1	-2.342030E+03
tan(x)	A+B*sqrt(i)	2.058464E-07	1	-2.323701E+03
tan(x)	A+B*log(i)^4	2.317584E-07	1	-2.299755E+03
tanh(x)	A+B*i	1.163149E-06	0.99999999	-5.621422E+03
tanh(x)	A+B/i	4.549376E-07	1	-6.186535E+03
tanh(x)	A+B*sqrt(i)	4.432855E-07	1	-6.202155E+03
tanh(x)	A+B*log(i)^4	1.050086E-06	1	-5.682982E+03
tinv(0.95,x)	A+B*i	1.332335E-03	0.89096396	-3.416258E+03
tinv(0.95,x)	A+B/i	1.337434E-03	0.89012785	-3.408764E+03
tinv(0.95,x)	A+B*sqrt(i)	1.290358E-03	0.8977264	-3.479068E+03
tinv(0.95,x)	A+B*log(i)^4	1.277457E-03	0.89976121	-3.498783E+03
tinv(0.975,x)	A+B*i	2.837031E-03	0.83714524	-1.933330E+03
tinv(0.975,x)	A+B/i	2.847291E-03	0.83596517	-1.926247E+03
tinv(0.975,x)	A+B*sqrt(i)	2.360721E-03	0.88723828	-2.293927E+03
tinv(0.975,x)	A+B*log(i)^4	1.758993E-03	0.93739619	-2.871198E+03
trigamma(x)	A+B*i	1.804246E-03	0.79614991	-2.840399E+03
trigamma(x)	A+B/i	2.149328E-03	0.71071563	-2.493524E+03
trigamma(x)	A+B*sqrt(i)	1.855956E-03	0.78429773	-2.784394E+03
trigamma(x)	A+B*log(i)^4	1.790606E-03	0.79922033	-2.855439E+03

Sine Series of Order 6

The next table shows a summary of results for the Sine series of the order 6:

$$Y = a_0 + a_1 * \sin(S_1 * gx(1,A_1,B_1) + Os_1) + \ldots$$

$$+ a_6 * \sin(S_6 * gx(6,A_6,B_6) + Os_6) + a_7*x + a_8*x^2$$

The output text files for this series are located in the following folder:

Fourier-Shammas Series Approximations Quadratic Fit with 6 Sine

The files are named using the following general format:

*fxName_n*_6_sin_run1.txt

Where fxName is the function name and n is the gx series number which varies between 1 and 4. Each output files specifies the function being approximated and the gx series being used. For example, the file for the inverse hyperboplic cosine (acosh) using the first gx series is named acosh_1_6_sin_run1.txt.

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
10^x	A+B*i	7.473196E-09	1	-Inf
10^x	A+B/i	5.193137E-09	1	-Inf
10^x	A+B*sqrt(i)	5.484565E-09	1	-Inf
10^x	A+B*log(i)^4	1.636504E-08	1	-2.830749E+03
acosh(x)	A+B*i	1.988203E-03	0.99513304	-2.643933E+03
acosh(x)	A+B/i	2.910062E-03	0.98957345	-1.888903E+03
acosh(x)	A+B*sqrt(i)	2.017124E-03	0.99499042	-2.615310E+03
acosh(x)	A+B*log(i)^4	1.708106E-03	0.99640776	-2.944892E+03
arccos(x)	A+B*i	5.136726E-04	0.99980338	-7.392020E+02
arccos(x)	A+B/i	5.025193E-04	0.99981182	-7.436363E+02
arccos(x)	A+B*sqrt(i)	4.920674E-04	0.99981957	-7.478820E+02
arccos(x)	A+B*log(i)^4	5.684076E-04	0.99975924	-7.187489E+02
arcsin(x)	A+B*i	4.992671E-04	0.99981425	-7.449478E+02
arcsin(x)	A+B/i	4.950963E-04	0.99981734	-7.466424E+02
arcsin(x)	A+B*sqrt(i)	4.972199E-04	0.99981577	-7.457778E+02
arcsin(x)	A+B*log(i)^4	5.416769E-04	0.99978135	-7.284791E+02
arctan(x)	A+B*i	3.909614E-10	1	-Inf
arctan(x)	A+B/i	3.904055E-10	1	-Inf
arctan(x)	A+B*sqrt(i)	3.063179E-10	1	-Inf
arctan(x)	A+B*log(i)^4	6.363704E-10	1	-3.485459E+03
asinh(x)	A+B*i	1.375260E-03	0.99793173	-3.398843E+03
asinh(x)	A+B/i	2.739510E-03	0.99179302	-2.019192E+03
asinh(x)	A+B*sqrt(i)	2.256981E-03	0.99442951	-2.407082E+03
asinh(x)	A+B*log(i)^4	1.658930E-03	0.9969905	-3.023408E+03
atanh(x)	A+B*i	1.052416E-03	0.99665776	-3.931519E+03
atanh(x)	A+B/i	1.014321E-03	0.99689535	-4.005257E+03
atanh(x)	A+B*sqrt(i)	1.015678E-03	0.99688704	-4.002583E+03
atanh(x)	A+B*log(i)^4	1.103315E-03	0.99632666	-3.837058E+03
Ci(x)	A+B*i	2.599192E-03	0.93243293	-7.056993E+02
Ci(x)	A+B/i	1.831524E-03	0.96645069	-8.429198E+02
Ci(x)	A+B*sqrt(i)	2.371610E-03	0.94374711	-7.416189E+02
Ci(x)	A+B*log(i)^4	2.899862E-03	0.91589672	-6.627900E+02
cosh(x)	A+B*i	8.822159E-04	0.99999881	-2.474477E+03
cosh(x)	A+B/i	7.909518E-07	1	-9.505470E+03
cosh(x)	A+B*sqrt(i)	1.071925E-05	1	-6.893694E+03
cosh(x)	A+B*log(i)^4	1.272270E-04	0.99999998	-4.414814E+03
digamma(x)	A+B/i	9.984227E-04	0.99865011	-3.978281E+03
digamma(x)	A+B*i	5.185153E-04	0.99963592	-5.263797E+03
digamma(x)	A+B*sqrt(i)	1.281034E-03	0.99777777	-3.489260E+03
digamma(x)	A+B*log(i)^4	6.873482E-04	0.99936023	-4.710765E+03
erf(x)	A+B*i	4.035661E-09	1	-6.391489E+03
erf(x)	A+B/i	2.830423E-09	1	-6.541099E+03
erf(x)	A+B*sqrt(i)	8.384524E-10	1	-Inf
erf(x)	A+B*log(i)^4	1.437028E-08	1	-5.855590E+03
exp(x)	A+B*i	5.253046E-08	1	-5.064953E+03
exp(x)	A+B/i	1.175380E-09	1	-Inf
exp(x)	A+B*sqrt(i)	4.459318E-09	1	-Inf
exp(x)	A+B*log(i)^4	3.559185E-08	1	-5.221447E+03
FresnelCosine(x)	A+B*i	1.653946E-03	0.94766818	-1.704379E+03

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
FresnelCosine(x)	A+B/i	7.079773E-04	0.99041126	-2.469732E+03
FresnelCosine(x)	A+B*sqrt(i)	5.665873E-04	0.99385875	-2.670681E+03
FresnelCosine(x)	A+B*log(i)^4	1.493448E-03	0.95733192	-1.796452E+03
FresnelSine(x)	A+B*i	7.776286E-04	0.98652718	-2.163700E+03
FresnelSine(x)	A+B/i	5.052909E-04	0.99431151	-2.509454E+03
FresnelSine(x)	A+B*sqrt(i)	6.955343E-04	0.98922167	-2.253178E+03
FresnelSine(x)	A+B*log(i)^4	4.085304E-04	0.99628154	-2.679934E+03
J0(x)	A+B*i	1.753286E-03	0.96854236	-1.148115E+03
J0(x)	A+B/i	6.636647E-04	0.99549269	-1.694081E+03
J0(x)	A+B*sqrt(i)	4.733648E-04	0.99770696	-1.883987E+03
J0(x)	A+B*log(i)^4	1.081115E-03	0.98803909	-1.419841E+03
J1(x)	A+B*i	2.686715E-03	0.94618314	-9.548407E+02
J1(x)	A+B/i	1.431544E-04	0.99984721	-2.719996E+03
J1(x)	A+B*sqrt(i)	9.874291E-04	0.99273079	-1.557425E+03
J1(x)	A+B*log(i)^4	2.073637E-03	0.96794168	-1.110768E+03
J2(x)	A+B*i	2.997901E-03	0.91972855	-8.888657E+02
J2(x)	A+B/i	6.649626E-05	0.99996051	-3.181596E+03
J2(x)	A+B*sqrt(i)	2.133720E-03	0.95933682	-1.093573E+03
J2(x)	A+B*log(i)^4	1.395406E-03	0.98260888	-1.349231E+03
J3(x)	A+B*i	1.690236E-03	0.97105143	-1.233838E+03
J3(x)	A+B/i	1.483636E-03	0.97769578	-1.312322E+03
J3(x)	A+B*sqrt(i)	2.871393E-04	0.99916456	-2.300977E+03
J3(x)	A+B*log(i)^4	1.104083E-03	0.98764805	-1.490202E+03
J4(x)	A+B*i	2.869107E-03	0.90866976	-9.153004E+02
J4(x)	A+B/i	1.922014E-03	0.95901412	-1.156478E+03
J4(x)	A+B*sqrt(i)	1.836631E-03	0.96257472	-1.183833E+03
J4(x)	A+B*log(i)^4	2.152907E-03	0.94857531	-1.088184E+03
J5(x)	A+B*i	2.633210E-03	0.91415105	-9.669504E+02
J5(x)	A+B/i	1.826956E-03	0.95867429	-1.187013E+03
J5(x)	A+B*sqrt(i)	8.220143E-04	0.9916339	-1.667799E+03
J5(x)	A+B*log(i)^4	2.366883E-04	0.99930639	-2.417298E+03
ln(x)	A+B*i	8.487023E-05	0.99998111	-8.169558E+03
ln(x)	A+B/i	2.777416E-06	0.99999998	-1.433168E+04
ln(x)	A+B*sqrt(i)	5.438595E-05	0.99999224	-8.971480E+03
ln(x)	A+B*log(i)^4	4.847228E-05	0.99999384	-9.178915E+03
log(x)	A+B*i	3.514251E-05	0.99998283	-9.758403E+03
log(x)	A+B/i	1.690823E-06	0.99999996	-1.522602E+04
log(x)	A+B*sqrt(i)	1.606495E-05	0.99999641	-1.116896E+04
log(x)	A+B*log(i)^4	2.365168E-05	0.99999222	-1.047195E+04
log10Gamma(x)	A+B*i	1.707254E-03	0.99999869	-2.925736E+03
log10Gamma(x)	A+B/i	1.189873E-03	0.99999936	-3.634097E+03
log10Gamma(x)	A+B*sqrt(i)	1.233317E-03	0.99999931	-3.563738E+03
log10Gamma(x)	A+B*log(i)^4	9.305317E-04	0.99999961	-4.116446E+03
Si(x)	A+B*i	1.275835E-04	0.99973171	-1.816511E+03
Si(x)	A+B/i	6.968415E-04	0.99199633	-1.178143E+03
Si(x)	A+B*sqrt(i)	8.422979E-04	0.98830628	-1.106862E+03
Si(x)	A+B*log(i)^4	5.497463E-04	0.99501866	-1.267293E+03
sinh(x)	A+B*i	6.599852E-04	0.99999935	-2.765277E+03
sinh(x)	A+B/i	9.209849E-07	1	-9.352955E+03

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
sinh(x)	A+B*sqrt(i)	2.513452E-05	1	-6.039788E+03
sinh(x)	A+B*log(i)^4	1.138614E-04	0.99999998	-4.526027E+03
tan(x)	A+B*i	1.008838E-07	1	-2.463347E+03
tan(x)	A+B/i	6.491817E-08	1	-2.552398E+03
tan(x)	A+B*sqrt(i)	9.916645E-08	1	-2.466815E+03
tan(x)	A+B*log(i)^4	2.708152E-07	1	-2.263889E+03
tanh(x)	A+B*i	1.252446E-06	0.99999999	-5.572768E+03
tanh(x)	A+B/i	6.284876E-08	1	-7.374026E+03
tanh(x)	A+B*sqrt(i)	4.532047E-08	1	-7.570863E+03
tanh(x)	A+B*log(i)^4	9.543736E-07	1	-5.736391E+03
tinv(0.95,x)	A+B*i	1.492077E-03	0.86310988	-3.190052E+03
tinv(0.95,x)	A+B/i	1.365466E-03	0.88535591	-3.364028E+03
tinv(0.95,x)	A+B*sqrt(i)	1.271368E-03	0.90061242	-3.504120E+03
tinv(0.95,x)	A+B*log(i)^4	1.366970E-03	0.88510321	-3.361868E+03
tinv(0.975,x)	A+B*i	2.784144E-03	0.84299903	-1.966212E+03
tinv(0.975,x)	A+B/i	1.346105E-03	0.96329909	-3.392047E+03
tinv(0.975,x)	A+B*sqrt(i)	2.702013E-03	0.85212527	-2.024961E+03
tinv(0.975,x)	A+B*log(i)^4	2.743615E-03	0.84753668	-1.994983E+03
trigamma(x)	A+B*i	2.085691E-03	0.72731477	-2.549056E+03
trigamma(x)	A+B/i	1.769967E-03	0.80362248	-2.874381E+03
trigamma(x)	A+B*sqrt(i)	2.059997E-03	0.73399208	-2.573625E+03
trigamma(x)	A+B*log(i)^4	1.671107E-03	0.82494681	-2.988295E+03

Sine Series of Order 7

The next table shows a summary of results for the Sine series of the order 7:

$$Y = a_0 + a_1 * \sin(S_1 * gx(1,A_1,B_1) + Os_1) + \ldots$$

$$+ a_7 * \sin(S_7 * gx(7,A_7,B_7) + Os_7) + a_8*x + a_9*x^2$$

The output text files for this series are located in the following folder:

Fourier-Shammas Series Approximations Quadratic Fit with 7 Sine

The files are named using the following general format:

*fxName_n_*7_sin_run1.txt

Where fxName is the function name and n is the gx series number which varies between 1 and 4. Each output files specifies the function being approximated and the gx series being used. For example, the file for the inverse hyperboplic cosine (acosh) using the first gx series is named acosh_1_7_sin_run1.txt.

Copyright © 2020 by Namir Clement Shammas

Function	*gx(i,A,B)*	*MSSE*	*RsqrAdj*	*AICc*
10^x	A+B*i	1.008770E-08	1	-Inf
10^x	A+B/i	1.496569E-09	1	-Inf
10^x	A+B*sqrt(i)	8.013231E-09	1	-Inf
10^x	A+B*log(i)^4	6.811226E-09	1	-Inf
acosh(x)	A+B*i	1.775624E-03	0.9961142	-2.864015E+03
acosh(x)	A+B/i	2.030536E-03	0.99491841	-2.598134E+03
acosh(x)	A+B*sqrt(i)	1.390752E-03	0.99761615	-3.348232E+03
acosh(x)	A+B*log(i)^4	1.554971E-03	0.99701995	-3.127017E+03
arccos(x)	A+B*i	3.749406E-04	0.99989409	-7.983262E+02
arccos(x)	A+B/i	4.336138E-04	0.99985835	-7.689608E+02
arccos(x)	A+B*sqrt(i)	4.159552E-04	0.99986965	-7.773593E+02
arccos(x)	A+B*log(i)^4	5.055818E-04	0.99980743	-7.379425E+02
arcsin(x)	A+B*i	4.235095E-04	0.99986488	-7.737236E+02
arcsin(x)	A+B/i	4.316592E-04	0.99985962	-7.698734E+02
arcsin(x)	A+B*sqrt(i)	3.889178E-04	0.99988605	-7.909356E+02
arcsin(x)	A+B*log(i)^4	4.439620E-04	0.99985151	-7.641967E+02
arctan(x)	A+B*i	2.694023E-10	1	-Inf
arctan(x)	A+B/i	1.165568E-10	1	-Inf
arctan(x)	A+B*sqrt(i)	2.841368E-10	1	-Inf
arctan(x)	A+B*log(i)^4	4.011116E-10	1	-Inf
asinh(x)	A+B*i	2.921091E-03	0.99065959	-1.886667E+03
asinh(x)	A+B/i	1.853810E-03	0.99623811	-2.797005E+03
asinh(x)	A+B*sqrt(i)	1.889996E-03	0.99608981	-2.758303E+03
asinh(x)	A+B*log(i)^4	2.295237E-03	0.99423326	-2.369392E+03
atanh(x)	A+B*i	1.071312E-03	0.99653317	-3.891888E+03
atanh(x)	A+B/i	9.181714E-04	0.99745348	-4.200398E+03
atanh(x)	A+B*sqrt(i)	8.759305E-04	0.99768239	-4.294593E+03
atanh(x)	A+B*log(i)^4	1.024059E-03	0.99683225	-3.982107E+03
Ci(x)	A+B*i	3.011659E-04	0.99908799	-1.546353E+03
Ci(x)	A+B/i	3.918838E-04	0.99845581	-1.443138E+03
Ci(x)	A+B*sqrt(i)	3.475662E-03	0.87853197	-5.875688E+02
Ci(x)	A+B*log(i)^4	1.727415E-03	0.96999592	-8.616391E+02
cosh(x)	A+B*i	1.582678E-04	0.99999996	-4.191979E+03
cosh(x)	A+B/i	7.007304E-08	1	-1.192997E+04
cosh(x)	A+B*sqrt(i)	3.027787E-05	1	-5.849159E+03
cosh(x)	A+B*log(i)^4	1.054395E-04	0.99999998	-4.598943E+03
digamma(x)	A+B/i	5.854786E-04	0.99953534	-5.021451E+03
digamma(x)	A+B*i	1.064809E-03	0.99846305	-3.847938E+03
digamma(x)	A+B*sqrt(i)	6.982556E-04	0.99933909	-4.675834E+03
digamma(x)	A+B*log(i)^4	7.120601E-04	0.9993127	-4.637423E+03
erf(x)	A+B*i	5.247870E-08	1	-5.304792E+03
erf(x)	A+B/i	4.979829E-10	1	-Inf
erf(x)	A+B*sqrt(i)	2.463357E-08	1	-5.623950E+03
erf(x)	A+B*log(i)^4	8.185792E-08	1	-5.116973E+03
exp(x)	A+B*i	7.362320E-09	1	-5.850534E+03
exp(x)	A+B/i	8.657293E-10	1	-Inf
exp(x)	A+B*sqrt(i)	4.953511E-09	1	-Inf
exp(x)	A+B*log(i)^4	7.615485E-09	1	-5.836986E+03
FresnelCosine(x)	A+B*i	8.218804E-04	0.98704839	-2.331078E+03

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
FresnelCosine(x)	A+B/i	3.896423E-04	0.99708902	-3.004300E+03
FresnelCosine(x)	A+B*sqrt(i)	1.735734E-04	0.99942234	-3.733682E+03
FresnelCosine(x)	A+B*log(i)^4	8.735243E-04	0.98536958	-2.276109E+03
FresnelSine(x)	A+B*i	1.569835E-03	0.94495324	-1.596210E+03
FresnelSine(x)	A+B/i	6.331479E-04	0.99104565	-2.324443E+03
FresnelSine(x)	A+B*sqrt(i)	6.498094E-04	0.99056818	-2.303612E+03
FresnelSine(x)	A+B*log(i)^4	8.327176E-05	0.99984511	-3.951376E+03
J0(x)	A+B*i	1.763919E-03	0.96804216	-1.140566E+03
J0(x)	A+B/i	2.637550E-04	0.99928547	-2.208519E+03
J0(x)	A+B*sqrt(i)	2.410215E-04	0.99940333	-2.259175E+03
J0(x)	A+B*log(i)^4	1.455494E-03	0.9782409	-1.248578E+03
J1(x)	A+B*i	3.827950E-04	0.99890378	-2.123743E+03
J1(x)	A+B/i	9.194516E-04	0.99367555	-1.596224E+03
J1(x)	A+B*sqrt(i)	7.947483E-04	0.99527476	-1.683966E+03
J1(x)	A+B*log(i)^4	1.015339E-03	0.99228765	-1.536505E+03
J2(x)	A+B*i	6.646496E-04	0.99604085	-1.791583E+03
J2(x)	A+B/i	1.448937E-03	0.9811845	-1.322429E+03
J2(x)	A+B*sqrt(i)	8.049085E-04	0.99419357	-1.676319E+03
J2(x)	A+B*log(i)^4	3.158495E-05	0.99999106	-3.625624E+03
J3(x)	A+B*i	6.764371E-04	0.9953476	-1.781000E+03
J3(x)	A+B/i	1.182632E-03	0.98577929	-1.444688E+03
J3(x)	A+B*sqrt(i)	1.905247E-03	0.96309165	-1.157613E+03
J3(x)	A+B*log(i)^4	5.104516E-04	0.9973507	-1.950490E+03
J4(x)	A+B*i	2.481932E-03	0.93142109	-9.984285E+02
J4(x)	A+B/i	1.605063E-05	0.99999713	-4.033137E+03
J4(x)	A+B*sqrt(i)	1.796875E-03	0.96405433	-1.192867E+03
J4(x)	A+B*log(i)^4	5.012494E-05	0.99997203	-3.347597E+03
J5(x)	A+B*i	7.836486E-04	0.99237049	-1.692433E+03
J5(x)	A+B/i	1.866492E-03	0.95671813	-1.169984E+03
J5(x)	A+B*sqrt(i)	1.837129E-03	0.9580692	-1.179530E+03
J5(x)	A+B*log(i)^4	1.749840E-04	0.99961959	-2.594992E+03
ln(x)	A+B*i	6.057662E-05	0.99999037	-8.773173E+03
ln(x)	A+B/i	1.992288E-06	0.99999999	-1.492633E+04
ln(x)	A+B*sqrt(i)	6.865160E-05	0.99998763	-8.547679E+03
ln(x)	A+B*log(i)^4	1.013105E-05	0.99999973	-1.199570E+04
log(x)	A+B*i	3.254350E-05	0.99998526	-9.892813E+03
log(x)	A+B/i	1.036438E-06	0.99999999	-1.610392E+04
log(x)	A+B*sqrt(i)	2.156105E-05	0.99999353	-1.063468E+04
log(x)	A+B*log(i)^4	9.501002E-06	0.99999874	-1.211140E+04
log10Gamma(x)	A+B*i	5.591009E-04	0.99999986	-5.111898E+03
log10Gamma(x)	A+B/i	3.008216E-03	0.99999591	-1.810299E+03
log10Gamma(x)	A+B*sqrt(i)	1.451839E-03	0.99999905	-3.239648E+03
log10Gamma(x)	A+B*log(i)^4	6.362580E-04	0.99999982	-4.858262E+03
Si(x)	A+B*i	1.198038E-03	0.97620992	-9.701633E+02
Si(x)	A+B/i	3.998118E-04	0.99735049	-1.382803E+03
Si(x)	A+B*sqrt(i)	4.778623E-04	0.99621505	-1.315752E+03
Si(x)	A+B*log(i)^4	6.467040E-04	0.99306789	-1.201987E+03
sinh(x)	A+B*i	6.06E-04	0.99999945	-2.846924E+03
sinh(x)	A+B/i	1.38E-07	1	-1.125174E+04

Function	***gx(i,A,B)***	***MSSE***	***RsqrAdj***	***AICc***
sinh(x)	A+B*sqrt(i)	9.50E-06	1	-7.010833E+03
sinh(x)	A+B*log(i)^4	1.37E-05	1	-6.644408E+03
tan(x)	A+B*i	4.64E-08	1	-2.615918E+03
tan(x)	A+B/i	2.08E-08	1	-2.777669E+03
tan(x)	A+B*sqrt(i)	2.63E-08	1	-2.730761E+03
tan(x)	A+B*log(i)^4	5.40E-08	1	-2.585194E+03
tanh(x)	A+B*i	7.06E-06	0.9999998	-4.527830E+03
tanh(x)	A+B/i	9.64E-09	1	-8.498460E+03
tanh(x)	A+B*sqrt(i)	9.79E-09	1	-8.488915E+03
tanh(x)	A+B*log(i)^4	8.46E-07	1	-5.805025E+03
tinv(0.95,x)	A+B*i	1.39E-03	0.88053403	-3.320580E+03
tinv(0.95,x)	A+B/i	5.73E-04	0.97977308	-5.062853E+03
tinv(0.95,x)	A+B*sqrt(i)	1.44E-03	0.87251273	-3.256830E+03
tinv(0.95,x)	A+B*log(i)^4	9.20E-04	0.94786642	-4.134047E+03
tinv(0.975,x)	A+B*i	2.06E-03	0.91419442	-2.555870E+03
tinv(0.975,x)	A+B/i	2.29E-03	0.89332417	-2.342296E+03
tinv(0.975,x)	A+B*sqrt(i)	2.09E-03	0.91154218	-2.526006E+03
tinv(0.975,x)	A+B*log(i)^4	1.65E-03	0.94462241	-2.985458E+03
trigamma(x)	A+B*i	1.72E-03	0.81495906	-2.930276E+03
trigamma(x)	A+B/i	1.76E-03	0.80519118	-2.879298E+03
trigamma(x)	A+B*sqrt(i)	1.92E-03	0.7692028	-2.711302E+03
trigamma(x)	A+B*log(i)^4	1.80E-03	0.79674062	-2.837215E+03

Cosine Series of Order 3

The next table shows a summary of results for the Sine series of the order 3:

$$Y = a_0 + a_1 * \cos(C_1 * gx(1,A_1,B_1) + Oc_1) + \ldots$$

$$+ a_3 * \cos(C_3 * gx(3,A_3,B_3) + Oc_3) + a_4*x + a_5*x^2$$

The output text files for this series are located in the following folder:

Fourier-Shammas Series Approximations Quadratic Fit with 3 Cosine

The files are named using the following general format:

*fxName_n*_3_cos_run1.txt

Where fxName is the function name and n is the gx series number which varies between 1 and 4. Each output files specifies the function being approximated and the gx series being used. For example, the file for the inverse hyperboplic cosine (acosh) using the first gx series is named acosh_1_3_cos_run1.txt.

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
10^x	A+B*i	1.587657E-06	1	-1.919711E+03
10^x	A+B/i	1.154254E-06	1	-1.984109E+03
10^x	A+B*sqrt(i)	9.438591E-07	1	-2.024758E+03
10^x	A+B*log(i)^4	3.544919E-07	1	-2.222575E+03
acosh(x)	A+B*i	2.594558E-03	0.99173699	-2.128452E+03
acosh(x)	A+B/i	2.616153E-03	0.99159887	-2.112024E+03
acosh(x)	A+B*sqrt(i)	3.333088E-03	0.98636344	-1.631995E+03
acosh(x)	A+B*log(i)^4	2.106245E-03	0.99455461	-2.541718E+03
arccos(x)	A+B*i	8.269585E-04	0.99950649	-6.561003E+02
arccos(x)	A+B/i	8.677672E-04	0.99945658	-6.463702E+02
arccos(x)	A+B*sqrt(i)	8.125344E-04	0.99952356	-6.596548E+02
arccos(x)	A+B*log(i)^4	8.554035E-04	0.99947196	-6.492693E+02
arcsin(x)	A+B*i	8.128417E-04	0.9995232	-6.595784E+02
arcsin(x)	A+B/i	8.883632E-04	0.99943048	-6.416319E+02
arcsin(x)	A+B*sqrt(i)	8.243127E-04	0.99950965	-6.567477E+02
arcsin(x)	A+B*log(i)^4	8.284588E-04	0.9995047	-6.557342E+02
arctan(x)	A+B*i	1.134080E-08	1	-2.917914E+03
arctan(x)	A+B/i	1.471258E-08	1	-2.865335E+03
arctan(x)	A+B*sqrt(i)	4.068562E-08	1	-2.659867E+03
arctan(x)	A+B*log(i)^4	2.331945E-09	1	-3.237064E+03
asinh(x)	A+B*i	2.418985E-03	0.99362041	-2.280400E+03
asinh(x)	A+B/i	3.653491E-03	0.98544734	-1.454905E+03
asinh(x)	A+B*sqrt(i)	3.858198E-03	0.98377087	-1.345762E+03
asinh(x)	A+B*log(i)^4	2.300980E-03	0.99422766	-2.380526E+03
atanh(x)	A+B*i	1.597484E-03	0.99232246	-3.108934E+03
atanh(x)	A+B/i	1.543244E-03	0.99283496	-3.178020E+03
atanh(x)	A+B*sqrt(i)	1.653870E-03	0.99177091	-3.039558E+03
atanh(x)	A+B*log(i)^4	1.583132E-03	0.99245979	-3.126983E+03
Ci(x)	A+B*i	1.929533E-03	0.96335195	-8.350082E+02
Ci(x)	A+B/i	2.281130E-03	0.94877919	-7.693902E+02
Ci(x)	A+B*sqrt(i)	2.207265E-03	0.95204266	-7.822937E+02
Ci(x)	A+B*log(i)^4	3.195442E-03	0.89949023	-6.372649E+02
cosh(x)	A+B*i	4.532693E-04	0.99999969	-3.153956E+03
cosh(x)	A+B/i	2.635207E-04	0.99999989	-3.697395E+03
cosh(x)	A+B*sqrt(i)	1.978672E-04	0.99999994	-3.984504E+03
cosh(x)	A+B*log(i)^4	1.484161E-04	0.99999997	-4.272656E+03
digamma(x)	A+B/i	1.599437E-03	0.99654646	-3.065826E+03
digamma(x)	A+B*i	1.014815E-03	0.99860972	-3.958429E+03
digamma(x)	A+B*sqrt(i)	2.206711E-03	0.99342614	-2.434354E+03
digamma(x)	A+B*log(i)^4	1.134150E-03	0.99826352	-3.740300E+03
erf(x)	A+B*i	7.758291E-08	1	-5.156504E+03
erf(x)	A+B/i	7.851585E-07	1	-4.179769E+03
erf(x)	A+B*sqrt(i)	1.674101E-07	1	-4.831944E+03
erf(x)	A+B*log(i)^4	1.924782E-07	1	-4.773060E+03
exp(x)	A+B*i	1.417857E-07	1	-4.678307E+03
exp(x)	A+B/i	2.542008E-07	1	-4.443616E+03
exp(x)	A+B*sqrt(i)	1.151334E-07	1	-4.762014E+03
exp(x)	A+B*log(i)^4	6.792643E-08	1	-4.974135E+03
FresnelCosine(x)	A+B*i	1.803265E-03	0.93821198	-1.638634E+03

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
FresnelCosine(x)	A+B/i	1.989080E-03	0.92482218	-1.550171E+03
FresnelCosine(x)	A+B*sqrt(i)	1.802792E-03	0.93824438	-1.638870E+03
FresnelCosine(x)	A+B*log(i)^4	3.417957E-03	0.77801761	-1.061855E+03
FresnelSine(x)	A+B*i	1.922779E-03	0.91825485	-1.449914E+03
FresnelSine(x)	A+B/i	1.829357E-03	0.9260054	-1.489860E+03
FresnelSine(x)	A+B*sqrt(i)	2.241987E-03	0.88886029	-1.326735E+03
FresnelSine(x)	A+B*log(i)^4	1.732079E-03	0.93366563	-1.533683E+03
J0(x)	A+B*i	1.220001E-03	0.98493472	-1.364276E+03
J0(x)	A+B/i	1.377926E-03	0.98078199	-1.295865E+03
J0(x)	A+B*sqrt(i)	1.237783E-03	0.98449236	-1.356144E+03
J0(x)	A+B*log(i)^4	1.240817E-03	0.98441624	-1.354768E+03
J1(x)	A+B*i	1.188798E-03	0.98957078	-1.458030E+03
J1(x)	A+B/i	1.948134E-03	0.97199254	-1.160685E+03
J1(x)	A+B*sqrt(i)	2.512625E-03	0.95341014	-1.007502E+03
J1(x)	A+B*log(i)^4	1.949064E-03	0.9719658	-1.160397E+03
J2(x)	A+B*i	1.746476E-03	0.97303424	-1.226467E+03
J2(x)	A+B/i	2.588819E-03	0.94074968	-9.895183E+02
J2(x)	A+B*sqrt(i)	1.914126E-03	0.9676087	-1.171287E+03
J2(x)	A+B*log(i)^4	1.033692E-03	0.99055354	-1.542194E+03
J3(x)	A+B*i	2.555309E-03	0.93450923	-9.973615E+02
J3(x)	A+B/i	2.331091E-03	0.94549812	-1.052647E+03
J3(x)	A+B*sqrt(i)	2.437271E-03	0.94041998	-1.025833E+03
J3(x)	A+B*log(i)^4	1.500872E-03	0.97740667	-1.317702E+03
J4(x)	A+B*i	2.718879E-03	0.91881762	-9.600095E+02
J4(x)	A+B/i	1.688929E-03	0.96867402	-1.246637E+03
J4(x)	A+B*sqrt(i)	1.811558E-03	0.96395988	-1.204441E+03
J4(x)	A+B*log(i)^4	1.316549E-03	0.98096486	-1.396583E+03
J5(x)	A+B*i	5.198624E-03	0.66879123	-5.698086E+02
J5(x)	A+B/i	3.706921E-03	0.8315963	-7.734004E+02
J5(x)	A+B*sqrt(i)	3.707381E-03	0.83155449	-7.733257E+02
J5(x)	A+B*log(i)^4	1.626323E-03	0.96758557	-1.269376E+03
ln(x)	A+B*i	9.517590E-05	0.99997633	-7.975151E+03
ln(x)	A+B/i	3.676746E-05	0.99999647	-9.689058E+03
ln(x)	A+B*sqrt(i)	3.740819E-05	0.99999634	-9.657926E+03
ln(x)	A+B*log(i)^4	3.736594E-05	0.99999635	-9.659962E+03
log(x)	A+B*i	3.439769E-05	0.99998361	-9.809114E+03
log(x)	A+B/i	1.765947E-05	0.99999568	-1.101054E+04
log(x)	A+B*sqrt(i)	1.493260E-05	0.99999691	-1.131278E+04
log(x)	A+B*log(i)^4	1.791047E-05	0.99999556	-1.098511E+04
log10Gamma(x)	A+B*i	1.568912E-03	0.99999889	-3.103632E+03
log10Gamma(x)	A+B/i	4.335527E-03	0.99999155	-1.109336E+03
log10Gamma(x)	A+B*sqrt(i)	4.450238E-03	0.9999911	-1.058100E+03
log10Gamma(x)	A+B*log(i)^4	1.571949E-03	0.99999889	-3.099838E+03
Si(x)	A+B*i	2.227522E-03	0.91956463	-7.537454E+02
Si(x)	A+B/i	1.801298E-03	0.94740143	-8.336012E+02
Si(x)	A+B*sqrt(i)	1.301831E-03	0.9725266	-9.557017E+02
Si(x)	A+B*log(i)^4	1.448552E-03	0.96598494	-9.155477E+02
sinh(x)	A+B*i	4.463114E-04	0.9999997	-3.169456E+03
sinh(x)	A+B/i	1.067072E-04	0.99999998	-4.603247E+03

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
sinh(x)	A+B*sqrt(i)	2.763432E-05	1	-5.956979E+03
sinh(x)	A+B*log(i)^4	1.197429E-04	0.99999998	-4.487758E+03
tan(x)	A+B*i	4.405820E-06	0.99999999	-1.713536E+03
tan(x)	A+B/i	4.616730E-06	0.99999999	-1.704090E+03
tan(x)	A+B*sqrt(i)	3.809157E-06	0.99999999	-1.742930E+03
tan(x)	A+B*log(i)^4	4.061342E-06	0.99999999	-1.730006E+03
tanh(x)	A+B*i	4.328850E-06	0.99999993	-4.838499E+03
tanh(x)	A+B/i	6.423210E-06	0.99999984	-4.600940E+03
tanh(x)	A+B*sqrt(i)	2.918515E-06	0.99999997	-5.075823E+03
tanh(x)	A+B*log(i)^4	5.366125E-06	0.99999988	-4.709187E+03
tinv(0.95,x)	A+B*i	1.342043E-03	0.88959617	-3.410076E+03
tinv(0.95,x)	A+B/i	1.439543E-03	0.87297177	-3.272476E+03
tinv(0.95,x)	A+B*sqrt(i)	1.838812E-03	0.792735	-2.792190E+03
tinv(0.95,x)	A+B*log(i)^4	1.292081E-03	0.89766353	-3.484513E+03
tinv(0.975,x)	A+B*i	2.470279E-03	0.87678246	-2.212985E+03
tinv(0.975,x)	A+B/i	3.194400E-03	0.79395635	-1.708617E+03
tinv(0.975,x)	A+B*sqrt(i)	3.316221E-03	0.77794139	-1.635186E+03
tinv(0.975,x)	A+B*log(i)^4	3.411410E-03	0.76501041	-1.579662E+03
trigamma(x)	A+B*i	1.803085E-03	0.79682557	-2.849736E+03
trigamma(x)	A+B/i	2.017726E-03	0.74557412	-2.626816E+03
trigamma(x)	A+B*sqrt(i)	2.326298E-03	0.66180474	-2.344764E+03
trigamma(x)	A+B*log(i)^4	1.857942E-03	0.78427474	-2.790335E+03

Cosine Series of Order 4

The next table shows a summary of results for the Sine series of the order 4:

$$Y = a_0 + a_1 * \cos(C_1 * gx(1,A_1,B_1) + Oc_1) + \ldots$$

$$+ a_4 * \cos(C_4 * gx(4,A_4,B_4) + Oc_4) + a_5*x + a_6*x^2$$

The output text files for this series are located in the following folder:

Fourier-Shammas Series Approximations Quadratic Fit with 4 Cosine

The files are named using the following general format:

*fxName_n_*4_cos_run1.txt

Where fxName is the function name and n is the gx series number which varies between 1 and 4. Each output files specifies the function being approximated and the gx series being used. For example, the file for the inverse hyperboplic cosine (acosh) using the first gx series is named acosh_1_4_cos_run1.txt.

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
10^x	A+B*i	5.069737E-08	1	-2.611114E+03
10^x	A+B/i	1.327292E-07	1	-2.416683E+03
10^x	A+B*sqrt(i)	9.976982E-08	1	-2.479436E+03
10^x	A+B*log(i)^4	5.887122E-08	1	-2.580921E+03
acosh(x)	A+B*i	2.498022E-03	0.99233265	-2.199575E+03
acosh(x)	A+B/i	2.815652E-03	0.99025884	-1.962340E+03
acosh(x)	A+B*sqrt(i)	3.358712E-03	0.98613889	-1.612788E+03
acosh(x)	A+B*log(i)^4	1.872143E-03	0.99569344	-2.771215E+03
arccos(x)	A+B*i	6.926182E-04	0.99965013	-6.875993E+02
arccos(x)	A+B/i	6.677230E-04	0.99967483	-6.949936E+02
arccos(x)	A+B*sqrt(i)	7.026962E-04	0.99963987	-6.846813E+02
arccos(x)	A+B*log(i)^4	6.812750E-04	0.99966149	-6.909349E+02
arcsin(x)	A+B*i	6.798887E-04	0.99966287	-6.913464E+02
arcsin(x)	A+B/i	6.965965E-04	0.9996461	-6.864424E+02
arcsin(x)	A+B*sqrt(i)	6.897543E-04	0.99965301	-6.884363E+02
arcsin(x)	A+B*log(i)^4	6.825828E-04	0.99966019	-6.905475E+02
arctan(x)	A+B*i	1.267587E-09	1	-3.355927E+03
arctan(x)	A+B/i	5.160020E-10	1	-3.534459E+03
arctan(x)	A+B*sqrt(i)	2.229195E-09	1	-3.242193E+03
arctan(x)	A+B*log(i)^4	1.988670E-09	1	-3.265123E+03
asinh(x)	A+B*i	2.764179E-03	0.99166136	-2.009314E+03
asinh(x)	A+B/i	3.209094E-03	0.988761	-1.710526E+03
asinh(x)	A+B*sqrt(i)	2.232401E-03	0.99456115	-2.437073E+03
asinh(x)	A+B*log(i)^4	2.292277E-03	0.99426548	-2.384084E+03
atanh(x)	A+B*i	1.341250E-03	0.99458242	-3.454562E+03
atanh(x)	A+B/i	1.404187E-03	0.99406206	-3.362849E+03
atanh(x)	A+B*sqrt(i)	1.314115E-03	0.99479941	-3.495439E+03
atanh(x)	A+B*log(i)^4	1.367607E-03	0.9943674	-3.415640E+03
Ci(x)	A+B*i	3.102107E-03	0.90477482	-6.447339E+02
Ci(x)	A+B/i	1.785334E-03	0.96845887	-8.613045E+02
Ci(x)	A+B*sqrt(i)	1.867830E-03	0.96547667	-8.435973E+02
Ci(x)	A+B*log(i)^4	1.842732E-03	0.96639819	-8.489002E+02
cosh(x)	A+B*i	8.646797E-04	0.99999886	-2.502734E+03
cosh(x)	A+B/i	4.142490E-05	1	-5.547288E+03
cosh(x)	A+B*sqrt(i)	2.097812E-04	0.99999993	-3.921861E+03
cosh(x)	A+B*log(i)^4	2.296202E-04	0.99999992	-3.831319E+03
digamma(x)	A+B/i	1.842455E-03	0.99541257	-2.784278E+03
digamma(x)	A+B*i	1.149124E-03	0.99821553	-3.710537E+03
digamma(x)	A+B*sqrt(i)	1.369356E-03	0.99746599	-3.366518E+03
digamma(x)	A+B*log(i)^4	1.274024E-03	0.99780654	-3.508096E+03
erf(x)	A+B*i	5.277695E-09	1	-6.286632E+03
erf(x)	A+B/i	3.872551E-08	1	-5.445590E+03
erf(x)	A+B*sqrt(i)	1.584439E-08	1	-5.822724E+03
erf(x)	A+B*log(i)^4	2.025972E-08	1	-5.718988E+03
exp(x)	A+B*i	4.826552E-08	1	-5.107355E+03
exp(x)	A+B/i	2.341154E-08	1	-5.398199E+03
exp(x)	A+B*sqrt(i)	1.153235E-08	1	-5.682818E+03
exp(x)	A+B*log(i)^4	2.195808E-07	1	-4.498323E+03
FresnelCosine(x)	A+B*i	2.050126E-03	0.91995696	-1.518841E+03

Copyright © 2020 by Namir Clement Shammas

Function	*gx(i,A,B)*	*MSSE*	*RsqrAdj*	*AICc*
FresnelCosine(x)	A+B/i	1.277471E-03	0.9689212	-1.945504E+03
FresnelCosine(x)	A+B*sqrt(i)	6.763070E-04	0.99128936	-2.519168E+03
FresnelCosine(x)	A+B*log(i)^4	1.839088E-03	0.93558792	-1.616827E+03
FresnelSine(x)	A+B*i	7.721365E-04	0.98678424	-2.177560E+03
FresnelSine(x)	A+B/i	1.961850E-03	0.91468299	-1.429709E+03
FresnelSine(x)	A+B*sqrt(i)	1.264471E-03	0.96455772	-1.781975E+03
FresnelSine(x)	A+B*log(i)^4	1.748404E-03	0.93223782	-1.522088E+03
J0(x)	A+B*i	6.092170E-04	0.99622964	-1.750444E+03
J0(x)	A+B/i	5.380626E-04	0.99705894	-1.820244E+03
J0(x)	A+B*sqrt(i)	1.217358E-03	0.98494518	-1.361391E+03
J0(x)	A+B*log(i)^4	8.405576E-04	0.99282249	-1.569541E+03
J1(x)	A+B*i	8.422236E-04	0.99474751	-1.661415E+03
J1(x)	A+B/i	7.894761E-04	0.99538482	-1.700349E+03
J1(x)	A+B*sqrt(i)	2.020956E-03	0.96975703	-1.134496E+03
J1(x)	A+B*log(i)^4	7.410210E-04	0.99593396	-1.738480E+03
J2(x)	A+B*i	1.588796E-03	0.97760772	-1.279333E+03
J2(x)	A+B/i	2.699011E-03	0.93537936	-9.603283E+02
J2(x)	A+B*sqrt(i)	8.622342E-04	0.99340503	-1.647279E+03
J2(x)	A+B*log(i)^4	8.078753E-04	0.99421037	-1.686480E+03
J3(x)	A+B*i	1.570706E-03	0.97517108	-1.286227E+03
J3(x)	A+B/i	1.556587E-03	0.97561546	-1.291663E+03
J3(x)	A+B*sqrt(i)	2.501639E-03	0.9370179	-1.006044E+03
J3(x)	A+B*log(i)^4	7.615084E-04	0.99416397	-1.722063E+03
J4(x)	A+B*i	2.717922E-03	0.91859884	-9.561249E+02
J4(x)	A+B/i	5.225646E-04	0.9969909	-1.948747E+03
J4(x)	A+B*sqrt(i)	1.741622E-03	0.96657552	-1.224045E+03
J4(x)	A+B*log(i)^4	4.216532E-04	0.99804085	-2.077916E+03
J5(x)	A+B*i	3.522341E-03	0.84743238	-8.000516E+02
J5(x)	A+B/i	1.652086E-03	0.96643669	-1.255818E+03
J5(x)	A+B*sqrt(i)	1.653031E-03	0.96639827	-1.255474E+03
J5(x)	A+B*log(i)^4	1.556087E-03	0.97022391	-1.291856E+03
ln(x)	A+B*i	8.180813E-05	0.99998249	-8.243852E+03
ln(x)	A+B/i	1.650630E-05	0.99999929	-1.112820E+04
ln(x)	A+B*sqrt(i)	2.729954E-05	0.99999805	-1.022156E+04
ln(x)	A+B*log(i)^4	7.179364E-05	0.99998652	-8.479159E+03
log(x)	A+B*i	1.998316E-05	0.99999446	-1.078375E+04
log(x)	A+B/i	7.128960E-06	0.9999993	-1.264111E+04
log(x)	A+B*sqrt(i)	1.778897E-05	0.99999561	-1.099334E+04
log(x)	A+B*log(i)^4	2.610412E-05	0.99999055	-1.030225E+04
log10Gamma(x)	A+B*i	1.563562E-03	0.9999989	-3.106305E+03
log10Gamma(x)	A+B/i	3.279480E-03	0.99999516	-1.653016E+03
log10Gamma(x)	A+B*sqrt(i)	4.599381E-03	0.99999048	-9.893954E+02
log10Gamma(x)	A+B*log(i)^4	9.274208E-04	0.99999961	-4.131086E+03
Si(x)	A+B*i	2.786515E-03	0.87343358	-6.654001E+02
Si(x)	A+B/i	7.601870E-05	0.9999058	-2.019590E+03
Si(x)	A+B*sqrt(i)	1.257325E-03	0.97423139	-9.646228E+02
Si(x)	A+B*log(i)^4	4.417974E-04	0.99681843	-1.357878E+03
sinh(x)	A+B*i	3.891805E-04	0.99999977	-3.302647E+03
sinh(x)	A+B/i	1.968971E-05	1	-6.292562E+03

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
sinh(x)	A+B*sqrt(i)	8.923110E-05	0.99999999	-4.778406E+03
sinh(x)	A+B*log(i)^4	2.396731E-04	0.99999991	-3.788384E+03
tan(x)	A+B*i	5.594299E-07	1	-2.126105E+03
tan(x)	A+B/i	1.241445E-06	1	-1.965088E+03
tan(x)	A+B*sqrt(i)	7.936549E-07	1	-2.055460E+03
tan(x)	A+B*log(i)^4	1.217820E-06	1	-1.968970E+03
tanh(x)	A+B*i	7.796762E-07	1	-5.866338E+03
tanh(x)	A+B/i	1.225333E-06	0.99999999	-5.594180E+03
tanh(x)	A+B*sqrt(i)	7.437817E-07	1	-5.894711E+03
tanh(x)	A+B*log(i)^4	1.097768E-06	1	-5.660360E+03
tinv(0.95,x)	A+B*i	1.316330E-03	0.89367723	-3.444003E+03
tinv(0.95,x)	A+B/i	1.552001E-03	0.85219777	-3.120866E+03
tinv(0.95,x)	A+B*sqrt(i)	1.405202E-03	0.8788358	-3.315819E+03
tinv(0.95,x)	A+B*log(i)^4	8.913399E-04	0.95124899	-4.208942E+03
tinv(0.975,x)	A+B*i	2.119447E-03	0.90920303	-2.509487E+03
tinv(0.975,x)	A+B/i	3.046444E-03	0.81240871	-1.797635E+03
tinv(0.975,x)	A+B*sqrt(i)	2.970278E-03	0.82167159	-1.847312E+03
tinv(0.975,x)	A+B*log(i)^4	1.789196E-03	0.93529439	-2.841829E+03
trigamma(x)	A+B*i	1.870742E-03	0.78106986	-2.772699E+03
trigamma(x)	A+B/i	1.626225E-03	0.83456054	-3.050325E+03
trigamma(x)	A+B*sqrt(i)	1.800071E-03	0.79729835	-2.849023E+03
trigamma(x)	A+B*log(i)^4	1.594071E-03	0.84103804	-3.089906E+03

Cosine Series of Order 5

The next table shows a summary of results for the Sine series of the order 5:

$$Y = a_0 + a_1 * \cos(C_1 * gx(1,A_1,B_1) + Oc_1) + \ldots$$

$$+ a_5 * \cos(C_5 * gx(5,A_5,B_5) + Oc_5) + a_6*x + a_7*x^2$$

The output text files for this series are located in the following folder:

Fourier-Shammas Series Approximations Quadratic Fit with 5 Cosine

The files are named using the following general format:

*fxName_n*_5_cos_run1.txt

Where fxName is the function name and n is the gx series number which varies between 1 and 4. Each output files specifies the function being approximated and the gx series being used. For example, the file for the inverse hyperboplic cosine (acosh) using the first gx series is named acosh_1_5_cos_run1.txt.

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
10^x	A+B*i	1.261086E-08	1	-2.887164E+03
10^x	A+B/i	9.427953E-09	1	-Inf
10^x	A+B*sqrt(i)	1.143440E-08	1	-2.907553E+03
10^x	A+B*log(i)^4	9.463674E-09	1	-Inf
acosh(x)	A+B*i	2.364323E-03	0.99312445	-2.304568E+03
acosh(x)	A+B/i	2.591022E-03	0.99174273	-2.123094E+03
acosh(x)	A+B*sqrt(i)	2.548873E-03	0.99200919	-2.155601E+03
acosh(x)	A+B*log(i)^4	2.004756E-03	0.9950567	-2.631537E+03
arccos(x)	A+B*i	5.956971E-04	0.99973841	-7.136892E+02
arccos(x)	A+B/i	5.859818E-04	0.99974688	-7.170108E+02
arccos(x)	A+B*sqrt(i)	5.756909E-04	0.99975569	-7.205898E+02
arccos(x)	A+B*log(i)^4	5.899332E-04	0.99974345	-7.156533E+02
arcsin(x)	A+B*i	5.821906E-04	0.99975014	-7.183219E+02
arcsin(x)	A+B/i	5.622897E-04	0.99976693	-7.253477E+02
arcsin(x)	A+B*sqrt(i)	5.675281E-04	0.99976257	-7.234745E+02
arcsin(x)	A+B*log(i)^4	5.614364E-04	0.99976764	-7.256545E+02
arctan(x)	A+B*i	1.078296E-09	1	-3.383956E+03
arctan(x)	A+B/i	3.817123E-10	1	-3.565059E+03
arctan(x)	A+B*sqrt(i)	3.370024E-10	1	-3.574378E+03
arctan(x)	A+B*log(i)^4	3.712585E-10	1	-Inf
asinh(x)	A+B*i	2.210007E-03	0.99466435	-2.453225E+03
asinh(x)	A+B/i	3.351385E-03	0.98772988	-1.619636E+03
asinh(x)	A+B*sqrt(i)	2.819393E-03	0.99131617	-1.965686E+03
asinh(x)	A+B*log(i)^4	1.721368E-03	0.99676297	-2.953478E+03
atanh(x)	A+B*i	1.180504E-03	0.99579894	-3.705850E+03
atanh(x)	A+B/i	1.190007E-03	0.99573103	-3.689814E+03
atanh(x)	A+B*sqrt(i)	1.156150E-03	0.99597049	-3.747543E+03
atanh(x)	A+B*log(i)^4	1.160650E-03	0.99593906	-3.739772E+03
Ci(x)	A+B*i	1.720440E-03	0.97055435	-8.716442E+02
Ci(x)	A+B/i	4.918157E-04	0.99759372	-1.362519E+03
Ci(x)	A+B*sqrt(i)	3.140732E-03	0.90186954	-6.357089E+02
Ci(x)	A+B*log(i)^4	1.527311E-03	0.97679417	-9.183203E+02
cosh(x)	A+B*i	3.130648E-04	0.99999985	-3.516650E+03
cosh(x)	A+B/i	5.248782E-06	1	-7.613241E+03
cosh(x)	A+B*sqrt(i)	3.215390E-05	1	-5.797078E+03
cosh(x)	A+B*log(i)^4	1.218171E-04	0.99999998	-4.462427E+03
digamma(x)	A+B/i	9.049193E-04	0.99889225	-4.175243E+03
digamma(x)	A+B*i	1.082618E-03	0.99841448	-3.823473E+03
digamma(x)	A+B*sqrt(i)	1.051702E-03	0.99850374	-3.880317E+03
digamma(x)	A+B*log(i)^4	5.511355E-04	0.9995891	-5.148130E+03
erf(x)	A+B*i	5.759600E-09	1	-6.245605E+03
erf(x)	A+B/i	2.728209E-08	1	-5.589242E+03
erf(x)	A+B*sqrt(i)	1.293968E-08	1	-5.904025E+03
erf(x)	A+B*log(i)^4	3.594503E-08	1	-5.472871E+03
exp(x)	A+B*i	1.213386E-07	1	-4.732594E+03
exp(x)	A+B/i	3.197596E-09	1	-Inf
exp(x)	A+B*sqrt(i)	3.062886E-09	1	-Inf
exp(x)	A+B*log(i)^4	1.453887E-08	1	-5.585528E+03
FresnelCosine(x)	A+B*i	1.028607E-03	0.97980517	-2.136874E+03

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
FresnelCosine(x)	A+B/i	5.437954E-04	0.99435567	-2.711798E+03
FresnelCosine(x)	A+B*sqrt(i)	1.736160E-03	0.9424665	-1.664704E+03
FresnelCosine(x)	A+B*log(i)^4	1.880958E-03	0.93246957	-1.592449E+03
FresnelSine(x)	A+B*i	1.750107E-03	0.93193299	-1.517224E+03
FresnelSine(x)	A+B/i	1.167805E-03	0.96969263	-1.841674E+03
FresnelSine(x)	A+B*sqrt(i)	2.125811E-03	0.89957146	-1.361254E+03
FresnelSine(x)	A+B*log(i)^4	1.715983E-03	0.93456146	-1.533016E+03
J0(x)	A+B*i	1.106625E-03	0.98751387	-1.410869E+03
J0(x)	A+B/i	8.419080E-04	0.99277304	-1.564519E+03
J0(x)	A+B*sqrt(i)	1.021012E-03	0.98937109	-1.456121E+03
J0(x)	A+B*log(i)^4	9.043286E-04	0.99166167	-1.524324E+03
J1(x)	A+B*i	1.779327E-03	0.9764765	-1.207041E+03
J1(x)	A+B/i	1.282912E-03	0.9877712	-1.403957E+03
J1(x)	A+B*sqrt(i)	1.114708E-03	0.99076764	-1.488561E+03
J1(x)	A+B*log(i)^4	1.932535E-04	0.99972251	-2.543473E+03
J2(x)	A+B*i	2.234846E-03	0.95554331	-1.069823E+03
J2(x)	A+B/i	1.084796E-03	0.9895254	-1.504937E+03
J2(x)	A+B*sqrt(i)	6.009292E-04	0.99678569	-1.860520E+03
J2(x)	A+B*log(i)^4	1.025427E-03	0.99064052	-1.538818E+03
J3(x)	A+B*i	1.484136E-03	0.97775692	-1.316245E+03
J3(x)	A+B/i	2.264668E-03	0.94820871	-1.061843E+03
J3(x)	A+B*sqrt(i)	2.290894E-04	0.99947002	-2.441067E+03
J3(x)	A+B*log(i)^4	2.575185E-04	0.99933032	-2.370646E+03
J4(x)	A+B*i	1.501662E-03	0.97506663	-1.309178E+03
J4(x)	A+B/i	5.361936E-05	0.99996821	-3.315293E+03
J4(x)	A+B*sqrt(i)	4.155566E-04	0.9980906	-2.082573E+03
J4(x)	A+B*log(i)^4	3.129656E-03	0.89169973	-8.670986E+02
J5(x)	A+B*i	1.973881E-03	0.95192477	-1.144573E+03
J5(x)	A+B/i	1.874836E-03	0.95662834	-1.175565E+03
J5(x)	A+B*sqrt(i)	1.030541E-03	0.98689582	-1.535824E+03
J5(x)	A+B*log(i)^4	1.005937E-03	0.98751408	-1.550371E+03
ln(x)	A+B*i	7.486808E-05	0.99998532	-8.399562E+03
ln(x)	A+B/i	9.521688E-06	0.99999976	-1.211557E+04
ln(x)	A+B*sqrt(i)	6.884317E-05	0.99998759	-8.550743E+03
ln(x)	A+B*log(i)^4	4.379610E-05	0.99999498	-9.365763E+03
log(x)	A+B*i	3.195652E-05	0.99998582	-9.933697E+03
log(x)	A+B/i	4.219132E-06	0.99999975	-1.358229E+04
log(x)	A+B*sqrt(i)	1.803726E-05	0.99999548	-1.096433E+04
log(x)	A+B*log(i)^4	2.015971E-05	0.99999436	-1.076386E+04
log10Gamma(x)	A+B*i	2.025516E-03	0.99999815	-2.594393E+03
log10Gamma(x)	A+B/i	6.726325E-04	0.9999998	-4.757264E+03
log10Gamma(x)	A+B*sqrt(i)	1.540070E-03	0.99999893	-3.131974E+03
log10Gamma(x)	A+B*log(i)^4	1.851794E-03	0.99999845	-2.770325E+03
Si(x)	A+B*i	1.802197E-03	0.94676388	-8.250731E+02
Si(x)	A+B/i	1.773597E-04	0.9994844	-1.696860E+03
Si(x)	A+B*sqrt(i)	2.056815E-03	0.93065866	-7.753839E+02
Si(x)	A+B*log(i)^4	1.947845E-04	0.99937811	-1.661623E+03
sinh(x)	A+B*i	3.700941E-04	0.99999979	-3.348968E+03
sinh(x)	A+B/i	7.523817E-06	1	-7.252443E+03

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
sinh(x)	A+B*sqrt(i)	4.068446E-05	1	-5.561295E+03
sinh(x)	A+B*log(i)^4	4.596017E-05	1	-5.439122E+03
tan(x)	A+B*i	1.814989E-07	1	-2.349129E+03
tan(x)	A+B/i	2.768043E-07	1	-2.263873E+03
tan(x)	A+B*sqrt(i)	1.664458E-07	1	-2.366618E+03
tan(x)	A+B*log(i)^4	3.252721E-07	1	-2.231280E+03
tanh(x)	A+B*i	3.677636E-07	1	-6.314593E+03
tanh(x)	A+B/i	4.438108E-07	1	-6.201442E+03
tanh(x)	A+B*sqrt(i)	3.864070E-07	1	-6.284823E+03
tanh(x)	A+B*log(i)^4	7.297068E-06	0.99999979	-4.515945E+03
tinv(0.95,x)	A+B*i	1.171604E-03	0.91568507	-3.668492E+03
tinv(0.95,x)	A+B/i	8.425605E-04	0.9563941	-4.315331E+03
tinv(0.95,x)	A+B*sqrt(i)	1.230355E-03	0.90701694	-3.572493E+03
tinv(0.95,x)	A+B*log(i)^4	1.075282E-03	0.9289788	-3.836813E+03
tinv(0.975,x)	A+B*i	2.345354E-03	0.8887015	-2.306740E+03
tinv(0.975,x)	A+B/i	3.285813E-03	0.78154691	-1.645198E+03
tinv(0.975,x)	A+B*sqrt(i)	2.720557E-03	0.8502427	-2.015579E+03
tinv(0.975,x)	A+B*log(i)^4	2.082624E-03	0.91224053	-2.539842E+03
trigamma(x)	A+B*i	1.615518E-03	0.83656569	-3.059384E+03
trigamma(x)	A+B/i	1.804337E-03	0.79612923	-2.840299E+03
trigamma(x)	A+B*sqrt(i)	1.911847E-03	0.77111059	-2.725588E+03
trigamma(x)	A+B*log(i)^4	1.790284E-03	0.79929259	-2.855796E+03

Cosine Series of Order 6

The next table shows a summary of results for the Sine series of the order 6:

$$Y = a_0 + a_1 * \cos(C_1 * gx(1,A_1,B_1) + Oc_1) + \ldots$$

$$+ a_6 * \cos(C_6 * gx(6,A_6,B_6) + Oc_6) + a_7*x + a_8*x^2$$

The output text files for this series are located in the following folder:

Fourier-Shammas Series Approximations Quadratic Fit with 6 Cosine

The files are named using the following general format:

*fxName_n*_6_cos_run1.txt

Where fxName is the function name and n is the gx series number which varies between 1 and 4. Each output files specifies the function being approximated and the gx series being used. For example, the file for the inverse hyperboplic cosine (acosh) using the first gx series is named acosh_1_6_cos_run1.txt.

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
10^x	A+B*i	1.244195E-08	1	-Inf
10^x	A+B/i	1.413461E-09	1	-Inf
10^x	A+B*sqrt(i)	6.459074E-09	1	-Inf
10^x	A+B*log(i)^4	1.391683E-08	1	-2.863129E+03
acosh(x)	A+B*i	2.673093E-03	0.9912024	-2.057251E+03
acosh(x)	A+B/i	2.042294E-03	0.99486462	-2.590732E+03
acosh(x)	A+B*sqrt(i)	2.349822E-03	0.99320161	-2.312724E+03
acosh(x)	A+B*log(i)^4	2.651669E-03	0.99134285	-2.073200E+03
arccos(x)	A+B*i	4.898429E-04	0.9998212	-7.487972E+02
arccos(x)	A+B/i	4.974471E-04	0.9998156	-7.456855E+02
arccos(x)	A+B*sqrt(i)	5.093359E-04	0.99980668	-7.409146E+02
arccos(x)	A+B*log(i)^4	5.016109E-04	0.9998125	-7.440018E+02
arcsin(x)	A+B*i	4.766002E-04	0.99983073	-7.543334E+02
arcsin(x)	A+B/i	4.893454E-04	0.99982156	-7.490025E+02
arcsin(x)	A+B*sqrt(i)	4.888080E-04	0.99982195	-7.492244E+02
arcsin(x)	A+B*log(i)^4	5.034508E-04	0.99981112	-7.432622E+02
arctan(x)	A+B*i	3.546696E-10	1	-Inf
arctan(x)	A+B/i	2.671005E-10	1	-Inf
arctan(x)	A+B*sqrt(i)	2.544082E-10	1	-Inf
arctan(x)	A+B*log(i)^4	1.099288E-09	1	-3.375273E+03
asinh(x)	A+B*i	2.073691E-03	0.99529753	-2.576647E+03
asinh(x)	A+B/i	6.547087E-04	0.99953126	-4.884743E+03
asinh(x)	A+B*sqrt(i)	2.287617E-03	0.99427725	-2.380089E+03
asinh(x)	A+B*log(i)^4	1.543380E-03	0.99739515	-3.167949E+03
atanh(x)	A+B*i	1.076465E-03	0.99650327	-3.886332E+03
atanh(x)	A+B/i	1.032225E-03	0.99678478	-3.970262E+03
atanh(x)	A+B*sqrt(i)	1.000005E-03	0.99698237	-4.033687E+03
atanh(x)	A+B*log(i)^4	1.043255E-03	0.9967157	-3.949005E+03
Ci(x)	A+B*i	1.758127E-03	0.96908573	-8.589523E+02
Ci(x)	A+B/i	1.596301E-03	0.97451481	-8.968038E+02
Ci(x)	A+B*sqrt(i)	2.287150E-03	0.94768242	-7.558338E+02
Ci(x)	A+B*log(i)^4	2.064581E-03	0.95736932	-7.959664E+02
cosh(x)	A+B*i	6.699631E-04	0.99999932	-2.750242E+03
cosh(x)	A+B/i	8.069879E-06	1	-7.178164E+03
cosh(x)	A+B*sqrt(i)	1.045104E-05	1	-6.919084E+03
cosh(x)	A+B*log(i)^4	1.253599E-04	0.99999998	-4.429628E+03
digamma(x)	A+B/i	1.965277E-03	0.99476983	-2.649591E+03
digamma(x)	A+B*i	7.470073E-04	0.99924435	-4.547460E+03
digamma(x)	A+B*sqrt(i)	1.030025E-03	0.99856331	-3.917141E+03
digamma(x)	A+B*log(i)^4	4.451600E-04	0.99973165	-5.563072E+03
erf(x)	A+B*i	1.725405E-09	1	-6.749603E+03
erf(x)	A+B/i	7.978414E-10	1	-Inf
erf(x)	A+B*sqrt(i)	1.267278E-09	1	-Inf
erf(x)	A+B*log(i)^4	4.672378E-08	1	-5.358014E+03
exp(x)	A+B*i	3.790530E-08	1	-5.196126E+03
exp(x)	A+B/i	1.949972E-09	1	-Inf
exp(x)	A+B*sqrt(i)	3.522171E-09	1	-Inf
exp(x)	A+B*log(i)^4	1.277873E-08	1	-5.633114E+03
FresnelCosine(x)	A+B*i	4.023908E-04	0.99690244	-2.979352E+03

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
FresnelCosine(x)	A+B/i	1.015116E-03	0.98028694	-2.144700E+03
FresnelCosine(x)	A+B*sqrt(i)	1.658709E-04	0.99947366	-3.778717E+03
FresnelCosine(x)	A+B*log(i)^4	1.144674E-03	0.97493391	-2.036355E+03
FresnelSine(x)	A+B*i	8.080840E-04	0.9854512	-2.132890E+03
FresnelSine(x)	A+B/i	5.342270E-04	0.99364134	-2.464794E+03
FresnelSine(x)	A+B*sqrt(i)	2.028422E-04	0.99908329	-3.241444E+03
FresnelSine(x)	A+B*log(i)^4	1.725580E-03	0.93365861	-1.524450E+03
J0(x)	A+B*i	1.065838E-03	0.98837474	-1.427839E+03
J0(x)	A+B/i	7.765933E-04	0.99382826	-1.605768E+03
J0(x)	A+B*sqrt(i)	1.278366E-03	0.98327636	-1.325656E+03
J0(x)	A+B*log(i)^4	9.355746E-04	0.9910427	-1.501099E+03
J1(x)	A+B*i	1.656104E-03	0.97955202	-1.246119E+03
J1(x)	A+B/i	1.385346E-03	0.98569158	-1.353587E+03
J1(x)	A+B*sqrt(i)	1.071393E-03	0.991442	-1.508295E+03
J1(x)	A+B*log(i)^4	1.461512E-03	0.98407499	-1.321367E+03
J2(x)	A+B*i	1.067339E-04	0.99989825	-2.896734E+03
J2(x)	A+B/i	6.068244E-05	0.99996711	-3.236674E+03
J2(x)	A+B*sqrt(i)	1.175986E-03	0.98764817	-1.452220E+03
J2(x)	A+B*log(i)^4	1.212901E-04	0.99986861	-2.819770E+03
J3(x)	A+B*i	1.453723E-04	0.99978586	-2.710740E+03
J3(x)	A+B/i	1.010661E-04	0.9998965	-2.929581E+03
J3(x)	A+B*sqrt(i)	1.006340E-04	0.99989738	-2.932160E+03
J3(x)	A+B*log(i)^4	1.623079E-03	0.97330611	-1.258245E+03
J4(x)	A+B*i	4.632135E-04	0.99761942	-2.013088E+03
J4(x)	A+B/i	2.061462E-04	0.99952851	-2.500469E+03
J4(x)	A+B*sqrt(i)	1.154952E-03	0.98520043	-1.463085E+03
J4(x)	A+B*log(i)^4	1.375376E-03	0.97901234	-1.357935E+03
J5(x)	A+B*i	3.886681E-03	0.8129654	-7.325605E+02
J5(x)	A+B/i	1.075637E-04	0.99985675	-2.892071E+03
J5(x)	A+B*sqrt(i)	3.054530E-03	0.88448113	-8.776002E+02
J5(x)	A+B*log(i)^4	5.100291E-04	0.99677928	-1.955128E+03
ln(x)	A+B*i	4.723527E-05	0.99999415	-9.225499E+03
ln(x)	A+B/i	3.268288E-06	0.99999997	-1.403841E+04
ln(x)	A+B*sqrt(i)	4.248854E-05	0.99999527	-9.416342E+03
ln(x)	A+B*log(i)^4	7.248189E-05	0.99998623	-8.453890E+03
log(x)	A+B*i	3.473324E-05	0.99998323	-9.779512E+03
log(x)	A+B/i	1.765875E-06	0.99999996	-1.514776E+04
log(x)	A+B*sqrt(i)	1.897424E-05	0.999995	-1.086903E+04
log(x)	A+B*log(i)^4	1.520007E-05	0.99999679	-1.126868E+04
log10Gamma(x)	A+B*i	1.818795E-03	0.99999851	-2.801565E+03
log10Gamma(x)	A+B/i	8.249481E-04	0.99999969	-4.352741E+03
log10Gamma(x)	A+B*sqrt(i)	1.009717E-03	0.99999954	-3.956210E+03
log10Gamma(x)	A+B*log(i)^4	1.530267E-03	0.99999894	-3.140466E+03
Si(x)	A+B*i	1.522864E-05	0.99999618	-2.615734E+03
Si(x)	A+B/i	3.486849E-04	0.99799605	-1.438481E+03
Si(x)	A+B*sqrt(i)	3.003528E-04	0.99851309	-1.494585E+03
Si(x)	A+B*log(i)^4	8.595492E-05	0.99987822	-1.965011E+03
sinh(x)	A+B*i	1.359119E-04	0.99999997	-4.348648E+03
sinh(x)	A+B/i	7.141460E-07	1	-9.607820E+03

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
sinh(x)	A+B*sqrt(i)	4.438705E-05	1	-5.469945E+03
sinh(x)	A+B*log(i)^4	1.732925E-04	0.99999995	-4.105188E+03
tan(x)	A+B*i	3.130466E-08	1	-2.699728E+03
tan(x)	A+B/i	1.023249E-07	1	-2.460454E+03
tan(x)	A+B*sqrt(i)	1.206286E-07	1	-2.427240E+03
tan(x)	A+B*log(i)^4	8.562868E-08	1	-2.496466E+03
tanh(x)	A+B*i	1.312597E-06	0.99999999	-5.544529E+03
tanh(x)	A+B/i	1.609004E-07	1	-6.808113E+03
tanh(x)	A+B*sqrt(i)	3.031151E-08	1	-7.813006E+03
tanh(x)	A+B*log(i)^4	1.264622E-06	0.99999999	-5.566944E+03
tinv(0.95,x)	A+B*i	1.555549E-03	0.85121571	-3.108316E+03
tinv(0.95,x)	A+B/i	1.442418E-03	0.8720701	-3.256462E+03
tinv(0.95,x)	A+B*sqrt(i)	1.312958E-03	0.89400352	-3.440965E+03
tinv(0.95,x)	A+B*log(i)^4	1.024942E-03	0.93540658	-3.926847E+03
tinv(0.975,x)	A+B*i	2.946926E-03	0.82410333	-1.854727E+03
tinv(0.975,x)	A+B/i	2.109356E-03	0.90988051	-2.510781E+03
tinv(0.975,x)	A+B*sqrt(i)	2.453195E-03	0.87810579	-2.214502E+03
tinv(0.975,x)	A+B*log(i)^4	2.176264E-03	0.90407269	-2.449513E+03
trigamma(x)	A+B*i	1.642316E-03	0.8309267	-3.022740E+03
trigamma(x)	A+B/i	1.740556E-03	0.81009461	-2.907592E+03
trigamma(x)	A+B*sqrt(i)	1.901491E-03	0.773353	-2.732316E+03
trigamma(x)	A+B*log(i)^4	1.801248E-03	0.79661991	-2.839658E+03

Cosine Series of Order 7

The next table shows a summary of results for the Sine series of the order 7:

$$Y = a_0 + a_1 * \cos(C_1 * gx(1,A_1,B_1) + Oc_1) + \ldots$$

$$+ a_7 * \cos(C_7 * gx(7,A_7,B_7) + Oc_7) + a_8*x + a_9*x^2$$

The output text files for this series are located in the following folder:

Fourier-Shammas Series Approximations Quadratic Fit with 7 Cosine

The files are named using the following general format:

*fxName_n_*7_cos_run1.txt

Where fxName is the function name and n is the gx series number which varies between 1 and 4. Each output files specifies the function being approximated and the gx series being used. For example, the file for the inverse hyperboplic cosine (acosh) using the first gx series is named acosh_1_7_cos_run1.txt.

Copyright © 2020 by Namir Clement Shammas Version 1.00.00

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
10^x	A+B*i	3.556105E-09	1	-Inf
10^x	A+B/i	4.097583E-09	1	-Inf
10^x	A+B*sqrt(i)	9.823078E-09	1	-Inf
10^x	A+B*log(i)^4	7.136680E-09	1	-Inf
acosh(x)	A+B*i	2.154437E-03	0.99427934	-2.480741E+03
acosh(x)	A+B/i	2.212300E-03	0.99396793	-2.428212E+03
acosh(x)	A+B*sqrt(i)	2.215852E-03	0.99394854	-2.425032E+03
acosh(x)	A+B*log(i)^4	1.696223E-03	0.99645395	-2.954687E+03
arccos(x)	A+B*i	4.279487E-04	0.99986203	-7.716173E+02
arccos(x)	A+B/i	4.448086E-04	0.99985094	-7.638118E+02
arccos(x)	A+B*sqrt(i)	4.179335E-04	0.99986841	-7.764008E+02
arccos(x)	A+B*log(i)^4	4.346943E-04	0.99985764	-7.684580E+02
arcsin(x)	A+B*i	3.799370E-04	0.99989125	-7.956548E+02
arcsin(x)	A+B/i	4.083863E-04	0.99987435	-7.810688E+02
arcsin(x)	A+B*sqrt(i)	4.067626E-04	0.99987535	-7.818735E+02
arcsin(x)	A+B*log(i)^4	4.390429E-04	0.99985478	-7.664473E+02
arctan(x)	A+B*i	8.853758E-11	1	-Inf
arctan(x)	A+B/i	4.191796E-10	1	-Inf
arctan(x)	A+B*sqrt(i)	4.381130E-10	1	-Inf
arctan(x)	A+B*log(i)^4	4.119201E-10	1	-Inf
asinh(x)	A+B*i	2.451507E-03	0.99342127	-2.237526E+03
asinh(x)	A+B/i	2.569100E-03	0.992775	-2.143727E+03
asinh(x)	A+B*sqrt(i)	2.384085E-03	0.99377816	-2.293357E+03
asinh(x)	A+B*log(i)^4	1.581322E-03	0.99726274	-3.115286E+03
atanh(x)	A+B*i	8.888040E-04	0.99761377	-4.265413E+03
atanh(x)	A+B/i	8.908933E-04	0.99760254	-4.260717E+03
atanh(x)	A+B*sqrt(i)	8.945200E-04	0.99758298	-4.252592E+03
atanh(x)	A+B*log(i)^4	9.627455E-04	0.99720022	-4.105588E+03
Ci(x)	A+B*i	2.802666E-03	0.92101773	-6.719327E+02
Ci(x)	A+B/i	2.521239E-04	0.99936083	-1.616028E+03
Ci(x)	A+B*sqrt(i)	1.561697E-03	0.9754766	-9.011734E+02
Ci(x)	A+B*log(i)^4	2.529415E-03	0.93566797	-7.121452E+02
cosh(x)	A+B*i	8.209282E-05	0.99999999	-4.849730E+03
cosh(x)	A+B/i	1.046040E-07	1	-1.152854E+04
cosh(x)	A+B*sqrt(i)	4.275611E-05	1	-5.503373E+03
cosh(x)	A+B*log(i)^4	4.130587E-05	1	-5.537950E+03
digamma(x)	A+B/i	1.188082E-03	0.99808659	-3.633010E+03
digamma(x)	A+B*i	5.088592E-04	0.999649	-5.296638E+03
digamma(x)	A+B*sqrt(i)	9.551350E-04	0.99876336	-4.061203E+03
digamma(x)	A+B*log(i)^4	1.363924E-03	0.99747829	-3.362204E+03
erf(x)	A+B*i	3.358551E-09	1	-6.464791E+03
erf(x)	A+B/i	6.627480E-10	1	-Inf
erf(x)	A+B*sqrt(i)	4.172130E-10	1	-Inf
erf(x)	A+B*log(i)^4	1.244902E-08	1	-5.911951E+03
exp(x)	A+B*i	4.264201E-08	1	-5.144575E+03
exp(x)	A+B/i	2.482187E-09	1	-Inf
exp(x)	A+B*sqrt(i)	1.497853E-09	1	-Inf
exp(x)	A+B*log(i)^4	4.941286E-09	1	-Inf
FresnelCosine(x)	A+B*i	6.854180E-04	0.99099222	-2.494850E+03

Copyright © 2020 by Namir Clement Shammas

Function	***gx(i,A,B)***	***MSSE***	***RsqrAdj***	***AICc***
FresnelCosine(x)	A+B/i	7.466407E-04	0.98931117	-2.417679E+03
FresnelCosine(x)	A+B*sqrt(i)	5.764988E-04	0.99362759	-2.650946E+03
FresnelCosine(x)	A+B*log(i)^4	7.080333E-04	0.99038799	-2.465569E+03
FresnelSine(x)	A+B*i	1.990620E-04	0.99911488	-3.252428E+03
FresnelSine(x)	A+B/i	6.600989E-04	0.99026712	-2.291012E+03
FresnelSine(x)	A+B*sqrt(i)	6.271891E-04	0.99121341	-2.332027E+03
FresnelSine(x)	A+B*log(i)^4	4.166869E-04	0.99612168	-2.659975E+03
J0(x)	A+B*i	1.088854E-03	0.98782246	-1.411682E+03
J0(x)	A+B/i	8.656681E-04	0.99230298	-1.540593E+03
J0(x)	A+B*sqrt(i)	3.167650E-04	0.99896939	-2.105595E+03
J0(x)	A+B*log(i)^4	1.111274E-03	0.98731581	-1.400227E+03
J1(x)	A+B*i	2.552271E-03	0.95126752	-9.816048E+02
J1(x)	A+B/i	6.020529E-05	0.99997288	-3.237286E+03
J1(x)	A+B*sqrt(i)	1.420421E-03	0.98490619	-1.334395E+03
J1(x)	A+B*log(i)^4	1.468766E-03	0.98386124	-1.314246E+03
J2(x)	A+B*i	1.501393E-03	0.9797975	-1.301020E+03
J2(x)	A+B/i	4.483804E-05	0.99998198	-3.414697E+03
J2(x)	A+B*sqrt(i)	5.134070E-04	0.99763767	-1.947014E+03
J2(x)	A+B*log(i)^4	9.605542E-04	0.99173086	-1.569896E+03
J3(x)	A+B*i	9.126435E-04	0.99153115	-1.600698E+03
J3(x)	A+B/i	2.608750E-04	0.99930803	-2.354585E+03
J3(x)	A+B*sqrt(i)	2.658106E-03	0.92815992	-9.571453E+02
J3(x)	A+B*log(i)^4	4.190415E-05	0.99998215	-3.455436E+03
J4(x)	A+B*i	1.895409E-03	0.96000397	-1.160729E+03
J4(x)	A+B/i	1.766928E-03	0.9652425	-1.202985E+03
J4(x)	A+B*sqrt(i)	1.973135E-03	0.95665646	-1.136536E+03
J4(x)	A+B*log(i)^4	1.492383E-03	0.9752895	-1.304644E+03
J5(x)	A+B*i	2.661171E-03	0.91201683	-9.564516E+02
J5(x)	A+B/i	9.936720E-04	0.98773295	-1.549490E+03
J5(x)	A+B*sqrt(i)	2.069880E-03	0.94677149	-1.107720E+03
J5(x)	A+B*log(i)^4	1.716767E-03	0.96338351	-1.220322E+03
ln(x)	A+B*i	7.796598E-05	0.99998404	-8.318414E+03
ln(x)	A+B/i	1.785566E-06	0.99999999	-1.512373E+04
ln(x)	A+B*sqrt(i)	1.152558E-05	0.99999965	-1.176330E+04
ln(x)	A+B*log(i)^4	3.755876E-05	0.9999963	-9.634534E+03
log(x)	A+B*i	2.525522E-05	0.99999112	-1.034970E+04
log(x)	A+B/i	1.249084E-06	0.99999998	-1.576763E+04
log(x)	A+B*sqrt(i)	1.617068E-05	0.99999636	-1.115309E+04
log(x)	A+B*log(i)^4	1.503753E-05	0.99999685	-1.128401E+04
log10Gamma(x)	A+B*i	5.897026E-04	0.99999984	-5.007347E+03
log10Gamma(x)	A+B/i	2.876532E-03	0.99999626	-1.898121E+03
log10Gamma(x)	A+B*sqrt(i)	1.547737E-03	0.99999892	-3.114152E+03
log10Gamma(x)	A+B*log(i)^4	5.056845E-04	0.99999988	-5.308917E+03
Si(x)	A+B*i	1.973429E-03	0.93544983	-7.825066E+02
Si(x)	A+B/i	2.757000E-05	0.9999874	-2.388327E+03
Si(x)	A+B*sqrt(i)	8.793501E-05	0.99987183	-1.952216E+03
Si(x)	A+B*log(i)^4	1.865250E-03	0.94233284	-8.037045E+02
sinh(x)	A+B*i	2.654429E-04	0.99999989	-3.673834E+03
sinh(x)	A+B/i	7.721564E-08	1	-1.183280E+04

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
sinh(x)	A+B*sqrt(i)	1.085640E-05	1	-6.876873E+03
sinh(x)	A+B*log(i)^4	1.378702E-05	1	-6.637422E+03
tan(x)	A+B*i	2.025711E-08	1	-2.783184E+03
tan(x)	A+B/i	3.276486E-08	1	-2.686052E+03
tan(x)	A+B*sqrt(i)	1.623760E-08	1	-2.827861E+03
tan(x)	A+B*log(i)^4	4.486363E-08	1	-2.622568E+03
tanh(x)	A+B*i	2.942526E-06	0.99999996	-5.054418E+03
tanh(x)	A+B/i	1.891811E-08	1	-8.092650E+03
tanh(x)	A+B*sqrt(i)	6.133500E-09	1	-8.770551E+03
tanh(x)	A+B*log(i)^4	6.012645E-07	1	-6.010387E+03
tinv(0.95,x)	A+B*i	1.263119E-03	0.90179697	-3.512851E+03
tinv(0.95,x)	A+B/i	1.297217E-03	0.89642336	-3.460588E+03
tinv(0.95,x)	A+B*sqrt(i)	1.290198E-03	0.89754115	-3.471233E+03
tinv(0.95,x)	A+B*log(i)^4	9.285098E-04	0.94693483	-4.116672E+03
tinv(0.975,x)	A+B*i	2.686796E-03	0.85363564	-2.032001E+03
tinv(0.975,x)	A+B/i	1.523960E-03	0.95291167	-3.144527E+03
tinv(0.975,x)	A+B*sqrt(i)	2.003087E-03	0.91864851	-2.608162E+03
tinv(0.975,x)	A+B*log(i)^4	1.949252E-03	0.92296253	-2.661614E+03
trigamma(x)	A+B*i	1.778728E-03	0.80147133	-2.860553E+03
trigamma(x)	A+B/i	2.250337E-03	0.68223997	-2.394423E+03
trigamma(x)	A+B*sqrt(i)	1.692739E-03	0.82020232	-2.958762E+03
trigamma(x)	A+B*log(i)^4	1.518179E-03	0.85537286	-3.174477E+03

Alternating Sine/Cosine Series of Order 3

The next table shows a summary of results for the Sine series of the order 3:

$$Y = a_0 + a_1 * \sin(S_1 * gx(1,A_1,B_1) + Os_1) + a_2 * \cos(S_2 * gx(2,A_2,B_2) + Os_2)$$

$$+ a_3 * \sin(S_3 * gx(3,A_3,B_3) + Os_3) + a_4*x + a_5*x^2$$

The output text files for this series are located in the following folder:

Fourier-Shammas Series Approximations Quadratic Fit with 3 Sine Cosine

The files are named using the following general format:

*fxName_n*_3_run1.txt

Where fxName is the function name and n is the gx series number which varies between 1 and 4. Each output files specifies the function being approximated and the gx series being used. For example, the file for the inverse hyperboplic cosine (acosh) using the first gx series is named acosh_1_3_run1.txt.

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
10^x	A+B*i	9.763849E-07	1	-2.017913E+03
10^x	A+B/i	1.793822E-06	1	-1.895048E+03
10^x	A+B*sqrt(i)	1.945560E-06	1	-1.878644E+03
10^x	A+B*log(i)^4	1.835613E-06	1	-1.890397E+03
acosh(x)	A+B*i	2.577557E-03	0.99184493	-2.141483E+03
acosh(x)	A+B/i	2.816287E-03	0.99026434	-1.965922E+03
acosh(x)	A+B*sqrt(i)	3.299555E-03	0.98663644	-1.652036E+03
acosh(x)	A+B*log(i)^4	2.215993E-03	0.99397235	-2.441045E+03
arccos(x)	A+B*i	8.279565E-04	0.9995053	-6.558567E+02
arccos(x)	A+B/i	8.304098E-04	0.99950237	-6.552591E+02
arccos(x)	A+B*sqrt(i)	7.376709E-04	0.99960731	-6.791802E+02
arccos(x)	A+B*log(i)^4	8.200975E-04	0.99951465	-6.577833E+02
arcsin(x)	A+B*i	8.407086E-04	0.99948995	-6.527692E+02
arcsin(x)	A+B/i	7.923020E-04	0.99954699	-6.647483E+02
arcsin(x)	A+B*sqrt(i)	8.262040E-04	0.99950739	-6.562847E+02
arcsin(x)	A+B*log(i)^4	8.404375E-04	0.99949027	-6.528344E+02
arctan(x)	A+B*i	1.814714E-09	1	-3.287921E+03
arctan(x)	A+B/i	6.946035E-09	1	-3.016937E+03
arctan(x)	A+B*sqrt(i)	5.458370E-09	1	-3.065628E+03
arctan(x)	A+B*log(i)^4	4.689177E-09	1	-3.096302E+03
asinh(x)	A+B*i	2.126086E-03	0.9950718	-2.538789E+03
asinh(x)	A+B/i	2.904950E-03	0.99079967	-1.913898E+03
asinh(x)	A+B*sqrt(i)	3.810770E-03	0.98416741	-1.370525E+03
asinh(x)	A+B*log(i)^4	2.257717E-03	0.99444268	-2.418526E+03
atanh(x)	A+B*i	1.585423E-03	0.99243795	-3.124091E+03
atanh(x)	A+B/i	1.538331E-03	0.99288051	-3.184397E+03
atanh(x)	A+B*sqrt(i)	1.685842E-03	0.99144968	-3.001264E+03
atanh(x)	A+B*log(i)^4	1.565584E-03	0.99262602	-3.149275E+03
Ci(x)	A+B*i	2.672265E-03	0.9297081	-7.073541E+02
Ci(x)	A+B/i	3.176452E-03	0.90068129	-6.396014E+02
Ci(x)	A+B*sqrt(i)	3.307065E-03	0.89234557	-6.238052E+02
Ci(x)	A+B*log(i)^4	2.304036E-03	0.94774537	-7.654737E+02
cosh(x)	A+B*i	9.173051E-05	0.99999999	-4.754783E+03
cosh(x)	A+B/i	1.209528E-04	0.99999998	-4.477685E+03
cosh(x)	A+B*sqrt(i)	6.495741E-04	0.99999936	-2.793406E+03
cosh(x)	A+B*log(i)^4	2.048938E-04	0.99999994	-3.949539E+03
digamma(x)	A+B/i	1.954101E-03	0.99484506	-2.672880E+03
digamma(x)	A+B*i	1.054802E-03	0.99849799	-3.882604E+03
digamma(x)	A+B*sqrt(i)	2.149475E-03	0.99376273	-2.485914E+03
digamma(x)	A+B*log(i)^4	2.206150E-03	0.99342948	-2.434852E+03
erf(x)	A+B*i	4.714559E-07	1	-4.395014E+03
erf(x)	A+B/i	4.373223E-07	1	-4.426729E+03
erf(x)	A+B*sqrt(i)	6.273224E-08	1	-5.246167E+03
erf(x)	A+B*log(i)^4	4.592462E-07	1	-4.406087E+03
exp(x)	A+B*i	3.327591E-07	1	-4.335360E+03
exp(x)	A+B/i	3.697201E-07	1	-4.293018E+03
exp(x)	A+B*sqrt(i)	8.544117E-08	1	-4.881915E+03
exp(x)	A+B*log(i)^4	4.184449E-07	1	-4.243251E+03
FresnelCosine(x)	A+B*i	2.080092E-03	0.91778512	-1.509816E+03

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
FresnelCosine(x)	A+B/i	2.131262E-03	0.91369046	-1.487896E+03
FresnelCosine(x)	A+B*sqrt(i)	1.828157E-03	0.93649437	-1.626268E+03
FresnelCosine(x)	A+B*log(i)^4	2.265256E-03	0.90249656	-1.432897E+03
FresnelSine(x)	A+B*i	1.741897E-03	0.93291149	-1.529150E+03
FresnelSine(x)	A+B/i	2.001537E-03	0.9114211	-1.417719E+03
FresnelSine(x)	A+B*sqrt(i)	1.817035E-03	0.92699887	-1.495280E+03
FresnelSine(x)	A+B*log(i)^4	1.765481E-03	0.93108254	-1.518364E+03
J0(x)	A+B*i	1.218499E-03	0.9849718	-1.364969E+03
J0(x)	A+B/i	6.946426E-04	0.99511596	-1.680800E+03
J0(x)	A+B*sqrt(i)	7.248545E-04	0.99468188	-1.656874E+03
J0(x)	A+B*log(i)^4	5.898311E-04	0.99647863	-1.772721E+03
J1(x)	A+B*i	1.078368E-03	0.99141837	-1.516722E+03
J1(x)	A+B/i	1.864622E-03	0.9743423	-1.187061E+03
J1(x)	A+B*sqrt(i)	1.797312E-03	0.97616127	-1.209194E+03
J1(x)	A+B*log(i)^4	8.163222E-04	0.99508233	-1.684315E+03
J2(x)	A+B*i	2.625620E-03	0.93905319	-9.810210E+02
J2(x)	A+B/i	2.587196E-03	0.94082395	-9.898959E+02
J2(x)	A+B*sqrt(i)	4.105955E-03	0.85095555	-7.118538E+02
J2(x)	A+B*log(i)^4	1.033631E-03	0.99055464	-1.542229E+03
J3(x)	A+B*i	2.288923E-03	0.94745206	-1.063637E+03
J3(x)	A+B/i	2.796696E-03	0.92155167	-9.430217E+02
J3(x)	A+B*sqrt(i)	2.255878E-03	0.94895839	-1.072391E+03
J3(x)	A+B*log(i)^4	1.549844E-03	0.97590821	-1.298373E+03
J4(x)	A+B*i	3.178720E-03	0.88903485	-8.659416E+02
J4(x)	A+B/i	2.414068E-03	0.93599988	-1.031591E+03
J4(x)	A+B*sqrt(i)	2.742702E-03	0.91738878	-9.547579E+02
J4(x)	A+B*log(i)^4	1.204085E-03	0.98407804	-1.450338E+03
J5(x)	A+B*i	3.557174E-03	0.84492735	-7.982240E+02
J5(x)	A+B/i	2.809410E-03	0.9032713	-9.402910E+02
J5(x)	A+B*sqrt(i)	2.357205E-03	0.93190431	-1.045941E+03
J5(x)	A+B*log(i)^4	1.855459E-03	0.95780826	-1.190026E+03
ln(x)	A+B*i	5.168854E-05	0.99999302	-9.075255E+03
ln(x)	A+B/i	4.461177E-05	0.9999948	-9.340579E+03
ln(x)	A+B*sqrt(i)	3.720901E-05	0.99999638	-9.667546E+03
ln(x)	A+B*log(i)^4	5.353924E-05	0.99999251	-9.011863E+03
log(x)	A+B*i	3.273160E-05	0.99998516	-9.898580E+03
log(x)	A+B/i	1.289307E-05	0.9999977	-1.157742E+04
log(x)	A+B*sqrt(i)	1.611397E-05	0.9999964	-1.117558E+04
log(x)	A+B*log(i)^4	1.992416E-05	0.9999945	-1.079311E+04
log10Gamma(x)	A+B*i	1.468573E-03	0.99999903	-3.233303E+03
log10Gamma(x)	A+B/i	3.659863E-03	0.99999398	-1.441733E+03
log10Gamma(x)	A+B*sqrt(i)	4.462645E-03	0.99999105	-1.052637E+03
log10Gamma(x)	A+B*log(i)^4	1.560265E-03	0.99999891	-3.114476E+03
Si(x)	A+B*i	2.542864E-03	0.8951788	-7.039626E+02
Si(x)	A+B/i	2.431742E-03	0.90413989	-7.207634E+02
Si(x)	A+B*sqrt(i)	2.497846E-03	0.89885735	-7.106787E+02
Si(x)	A+B*log(i)^4	6.279131E-04	0.99360851	-1.229853E+03
sinh(x)	A+B*i	1.491566E-04	0.99999997	-4.267668E+03
sinh(x)	A+B/i	1.665362E-04	0.99999996	-4.157233E+03

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
sinh(x)	A+B*sqrt(i)	5.237811E-05	1	-5.316270E+03
sinh(x)	A+B*log(i)^4	7.834047E-04	0.99999908	-2.605699E+03
tan(x)	A+B*i	4.227639E-06	0.99999999	-1.721875E+03
tan(x)	A+B/i	5.049849E-06	0.99999999	-1.685976E+03
tan(x)	A+B*sqrt(i)	6.656342E-06	0.99999997	-1.630182E+03
tan(x)	A+B*log(i)^4	1.690695E-06	1	-1.907008E+03
tanh(x)	A+B*i	1.405271E-06	0.99999999	-5.515792E+03
tanh(x)	A+B/i	4.446366E-06	0.99999992	-4.822374E+03
tanh(x)	A+B*sqrt(i)	3.751164E-06	0.99999994	-4.924727E+03
tanh(x)	A+B*log(i)^4	2.979690E-06	0.99999996	-5.063335E+03
tinv(0.95,x)	A+B*i	1.251471E-03	0.9039952	-3.547167E+03
tinv(0.95,x)	A+B/i	1.540632E-03	0.85450467	-3.139321E+03
tinv(0.95,x)	A+B*sqrt(i)	1.835248E-03	0.79353781	-2.795997E+03
tinv(0.95,x)	A+B*log(i)^4	1.306262E-03	0.89540475	-3.463096E+03
tinv(0.975,x)	A+B*i	2.310295E-03	0.89222566	-2.344353E+03
tinv(0.975,x)	A+B/i	3.050186E-03	0.81214048	-1.799255E+03
tinv(0.975,x)	A+B*sqrt(i)	3.339010E-03	0.77487901	-1.621750E+03
tinv(0.975,x)	A+B*log(i)^4	2.468553E-03	0.87695456	-2.214357E+03
trigamma(x)	A+B*i	1.800076E-03	0.79750301	-2.853046E+03
trigamma(x)	A+B/i	2.238468E-03	0.68685993	-2.421044E+03
trigamma(x)	A+B*sqrt(i)	2.333405E-03	0.65973538	-2.338719E+03
trigamma(x)	A+B*log(i)^4	1.878423E-03	0.77949243	-2.768606E+03

Alternating Sine/Cosine Series of Order 4

The next table shows a summary of results for the Sine series of the order 4:

$$Y = a_0 + a_1 * \sin(S_1 * gx(1,A_1,B_1) + Os_1) + a_2 * \cos(S_2 * gx(2,A_2,B_2) + Os_2)$$

$$+ \ldots + a_4 * \sin(S_4 * gx(4,A_4,B_4) + Os_4) + a_5*x + a_6*x^2$$

The output text files for this series are located in the following folder:

Fourier-Shammas Series Approximations Quadratic Fit with 4 Sine Cosine

The files are named using the following general format:

*fxName_n_*4_run1.txt

Where fxName is the function name and n is the gx series number which varies between 1 and 4. Each output files specifies the function being approximated and the gx series being used. For example, the file for the inverse hyperboplic cosine (acosh) using the first gx series is named acosh_1_4_run1.txt.

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
10^x	A+B*i	6.769250E-08	1	-2.552717E+03
10^x	A+B/i	1.714700E-08	1	-2.829918E+03
10^x	A+B*sqrt(i)	7.874893E-09	1	-Inf
10^x	A+B*log(i)^4	6.639467E-08	1	-2.556628E+03
acosh(x)	A+B*i	2.069897E-03	0.99473559	-2.572192E+03
acosh(x)	A+B/i	1.942087E-03	0.99536564	-2.698516E+03
acosh(x)	A+B*sqrt(i)	3.321178E-03	0.98644696	-1.635062E+03
acosh(x)	A+B*log(i)^4	1.641423E-03	0.9966895	-3.031888E+03
arccos(x)	A+B*i	6.782487E-04	0.99966449	-6.918342E+02
arccos(x)	A+B/i	7.002127E-04	0.99964241	-6.853965E+02
arccos(x)	A+B*sqrt(i)	6.845322E-04	0.99965825	-6.899715E+02
arccos(x)	A+B*log(i)^4	7.010905E-04	0.99964152	-6.851434E+02
arcsin(x)	A+B*i	6.758981E-04	0.99966682	-6.925355E+02
arcsin(x)	A+B/i	6.818329E-04	0.99966094	-6.907696E+02
arcsin(x)	A+B*sqrt(i)	6.714431E-04	0.99967119	-6.938714E+02
arcsin(x)	A+B*log(i)^4	6.717266E-04	0.99967092	-6.937861E+02
arctan(x)	A+B*i	1.618148E-09	1	-3.306733E+03
arctan(x)	A+B/i	1.154499E-09	1	-3.375025E+03
arctan(x)	A+B*sqrt(i)	2.547432E-09	1	-3.215236E+03
arctan(x)	A+B*log(i)^4	1.646261E-09	1	-3.303406E+03
asinh(x)	A+B*i	2.305244E-03	0.99420042	-2.372791E+03
asinh(x)	A+B/i	3.190114E-03	0.98889355	-1.722402E+03
asinh(x)	A+B*sqrt(i)	3.811662E-03	0.98414407	-1.366028E+03
asinh(x)	A+B*log(i)^4	1.898124E-03	0.99606801	-2.761821E+03
atanh(x)	A+B*i	1.362360E-03	0.99441054	-3.423329E+03
atanh(x)	A+B/i	1.325252E-03	0.99471088	-3.478560E+03
atanh(x)	A+B*sqrt(i)	1.320778E-03	0.99474653	-3.485323E+03
atanh(x)	A+B*log(i)^4	1.364319E-03	0.99439445	-3.420454E+03
Ci(x)	A+B*i	3.172094E-03	0.90042959	-6.359883E+02
Ci(x)	A+B/i	2.302704E-03	0.94752958	-7.615489E+02
Ci(x)	A+B*sqrt(i)	2.798341E-03	0.9225111	-6.851314E+02
Ci(x)	A+B*log(i)^4	2.621990E-03	0.93197003	-7.106479E+02
cosh(x)	A+B*i	5.340344E-04	0.99999957	-2.985597E+03
cosh(x)	A+B/i	2.583377E-05	1	-6.020432E+03
cosh(x)	A+B*sqrt(i)	6.585430E-05	0.99999999	-5.082799E+03
cosh(x)	A+B*log(i)^4	2.638273E-04	0.99999989	-3.692173E+03
digamma(x)	A+B/i	1.599130E-03	0.99654425	-3.062174E+03
digamma(x)	A+B*i	1.013346E-03	0.99861232	-3.957243E+03
digamma(x)	A+B*sqrt(i)	1.111802E-03	0.99832956	-3.775317E+03
digamma(x)	A+B*log(i)^4	5.887211E-04	0.99953162	-5.022727E+03
erf(x)	A+B*i	2.071489E-08	1	-5.709612E+03
erf(x)	A+B/i	3.041823E-08	1	-5.547484E+03
erf(x)	A+B*sqrt(i)	3.312960E-08	1	-5.511452E+03
erf(x)	A+B*log(i)^4	1.720275E-08	1	-5.788013E+03
exp(x)	A+B*i	1.601447E-07	1	-4.625212E+03
exp(x)	A+B/i	1.929321E-08	1	-5.475980E+03
exp(x)	A+B*sqrt(i)	6.087623E-09	1	-5.938783E+03
exp(x)	A+B*log(i)^4	4.474816E-08	1	-5.138377E+03
FresnelCosine(x)	A+B*i	1.548471E-03	0.95433658	-1.771972E+03

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
FresnelCosine(x)	A+B/i	8.435299E-04	0.98644925	-2.319872E+03
FresnelCosine(x)	A+B*sqrt(i)	4.128649E-04	0.99675378	-2.964329E+03
FresnelCosine(x)	A+B*log(i)^4	1.891651E-03	0.93185335	-1.591408E+03
FresnelSine(x)	A+B*i	1.716911E-03	0.93465692	-1.536665E+03
FresnelSine(x)	A+B/i	1.950598E-03	0.9156588	-1.434322E+03
FresnelSine(x)	A+B*sqrt(i)	1.300161E-03	0.96252874	-1.759652E+03
FresnelSine(x)	A+B*log(i)^4	1.881229E-03	0.92155101	-1.463363E+03
J0(x)	A+B*i	9.537503E-04	0.99075923	-1.498540E+03
J0(x)	A+B/i	9.497270E-04	0.99083703	-1.500915E+03
J0(x)	A+B*sqrt(i)	1.284910E-03	0.98322802	-1.331040E+03
J0(x)	A+B*log(i)^4	1.004047E-03	0.9897589	-1.469658E+03
J1(x)	A+B*i	1.518990E-03	0.98291478	-1.306382E+03
J1(x)	A+B/i	2.026030E-03	0.96960498	-1.132986E+03
J1(x)	A+B*sqrt(i)	1.563246E-03	0.98190471	-1.289093E+03
J1(x)	A+B*log(i)^4	1.950447E-03	0.97183051	-1.155874E+03
J2(x)	A+B*i	8.495552E-04	0.99359756	-1.656197E+03
J2(x)	A+B/i	2.573281E-03	0.94125967	-9.890459E+02
J2(x)	A+B*sqrt(i)	2.818801E-03	0.92951596	-9.341857E+02
J2(x)	A+B*log(i)^4	8.889254E-04	0.99299041	-1.628926E+03
J3(x)	A+B*i	8.097980E-04	0.99340034	-1.685049E+03
J3(x)	A+B/i	3.667530E-04	0.99864632	-2.161891E+03
J3(x)	A+B*sqrt(i)	2.578610E-03	0.93308258	-9.878005E+02
J3(x)	A+B*log(i)^4	2.276256E-03	0.9478553	-1.062881E+03
J4(x)	A+B*i	3.803003E-04	0.99840629	-2.140055E+03
J4(x)	A+B/i	1.106752E-03	0.98650239	-1.496985E+03
J4(x)	A+B*sqrt(i)	2.718593E-03	0.91855867	-9.559764E+02
J4(x)	A+B*log(i)^4	1.643752E-03	0.97022654	-1.258862E+03
J5(x)	A+B*i	3.756825E-03	0.82644323	-7.612536E+02
J5(x)	A+B/i	1.717854E-03	0.96371123	-1.232318E+03
J5(x)	A+B*sqrt(i)	1.771049E-03	0.96142902	-1.213959E+03
J5(x)	A+B*log(i)^4	1.877566E-03	0.95664988	-1.178799E+03
ln(x)	A+B*i	5.179039E-05	0.99999298	-9.067676E+03
ln(x)	A+B/i	2.074396E-05	0.99999887	-1.071642E+04
ln(x)	A+B*sqrt(i)	4.482994E-05	0.99999474	-9.327756E+03
ln(x)	A+B*log(i)^4	6.305801E-05	0.9999896	-8.712952E+03
log(x)	A+B*i	3.272308E-05	0.99998515	-9.895018E+03
log(x)	A+B/i	8.138995E-06	0.99999908	-1.240235E+04
log(x)	A+B*sqrt(i)	1.292950E-05	0.99999768	-1.156830E+04
log(x)	A+B*log(i)^4	1.970670E-05	0.99999461	-1.080885E+04
log10Gamma(x)	A+B*i	9.120816E-04	0.99999963	-4.163808E+03
log10Gamma(x)	A+B/i	1.599766E-03	0.99999885	-3.061393E+03
log10Gamma(x)	A+B*sqrt(i)	1.464143E-03	0.99999904	-3.235202E+03
log10Gamma(x)	A+B*log(i)^4	1.055174E-03	0.9999995	-3.877883E+03
Si(x)	A+B*i	1.606767E-03	0.9579175	-8.724136E+02
Si(x)	A+B/i	3.686637E-04	0.99778458	-1.425921E+03
Si(x)	A+B*sqrt(i)	3.359139E-04	0.9981607	-1.460900E+03
Si(x)	A+B*log(i)^4	1.503029E-03	0.96317603	-8.975084E+02
sinh(x)	A+B*i	3.795963E-04	0.99999978	-3.327632E+03
sinh(x)	A+B/i	2.188319E-05	1	-6.186728E+03

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
sinh(x)	A+B*sqrt(i)	5.100202E-05	1	-5.338890E+03
sinh(x)	A+B*log(i)^4	1.956079E-04	0.99999994	-3.991954E+03
tan(x)	A+B*i	1.114922E-06	1	-1.986802E+03
tan(x)	A+B/i	1.206554E-06	1	-1.970847E+03
tan(x)	A+B*sqrt(i)	1.076795E-06	1	-1.993830E+03
tan(x)	A+B*log(i)^4	9.728713E-07	1	-2.014337E+03
tanh(x)	A+B*i	7.642636E-07	1	-5.878357E+03
tanh(x)	A+B/i	2.132050E-06	0.99999998	-5.260749E+03
tanh(x)	A+B*sqrt(i)	9.145718E-07	1	-5.770272E+03
tanh(x)	A+B*log(i)^4	1.285457E-06	0.99999999	-5.565343E+03
tinv(0.95,x)	A+B*i	1.261739E-03	0.90231313	-3.527106E+03
tinv(0.95,x)	A+B/i	1.759190E-03	0.81010112	-2.875011E+03
tinv(0.95,x)	A+B*sqrt(i)	1.282712E-03	0.89903858	-3.494761E+03
tinv(0.95,x)	A+B*log(i)^4	9.674438E-04	0.94256874	-4.048192E+03
tinv(0.975,x)	A+B*i	2.809491E-03	0.84045561	-1.956501E+03
tinv(0.975,x)	A+B/i	2.992599E-03	0.81898132	-1.832623E+03
tinv(0.975,x)	A+B*sqrt(i)	2.441965E-03	0.87946725	-2.231575E+03
tinv(0.975,x)	A+B*log(i)^4	1.867344E-03	0.92951849	-2.757951E+03
trigamma(x)	A+B*i	1.513821E-03	0.85664035	-3.192284E+03
trigamma(x)	A+B/i	2.288100E-03	0.67248752	-2.373550E+03
trigamma(x)	A+B*sqrt(i)	1.781517E-03	0.80145545	-2.869558E+03
trigamma(x)	A+B*log(i)^4	1.441913E-03	0.86993639	-3.288741E+03

Alternating Sine/Cosine Series of Order 5

The next table shows a summary of results for the Sine series of the order 5:

$$Y = a_0 + a_1 * \sin(S_1 * gx(1,A_1,B_1) + Os_1) + a_2 * \cos(S_2 * gx(2,A_2,B_2) + Os_2)$$

$$+ \ldots + a_5 * \sin(S_5 * gx(5,A_5,B_5) + Os_5) + a_6*x + a_7*x^2$$

The output text files for this series are located in the following folder:

Fourier-Shammas Series Approximations Quadratic Fit with 5 Sine Cosine

The files are named using the following general format:

*fxName_n*_5_run1.txt

Where fxName is the function name and n is the gx series number which varies between 1 and 4. Each output files specifies the function being approximated and the gx series being used. For example, the file for the inverse hyperboplic cosine (acosh) using the first gx series is named acosh_1_5_run1.txt.

Function	*gx(i,A,B)*	*MSSE*	*RsqrAdj*	*AICc*
10^x	A+B*i	1.261086E-08	1	-2.887164E+03
10^x	A+B/i	9.427953E-09	1	-Inf
10^x	A+B*sqrt(i)	1.143440E-08	1	-2.907553E+03
10^x	A+B*log(i)^4	9.463674E-09	1	-Inf
acosh(x)	A+B*i	2.364323E-03	0.99312445	-2.304568E+03
acosh(x)	A+B/i	2.591022E-03	0.99174273	-2.123094E+03
acosh(x)	A+B*sqrt(i)	2.548873E-03	0.99200919	-2.155601E+03
acosh(x)	A+B*log(i)^4	2.004756E-03	0.9950567	-2.631537E+03
arccos(x)	A+B*i	5.956971E-04	0.99973841	-7.136892E+02
arccos(x)	A+B/i	5.859818E-04	0.99974688	-7.170108E+02
arccos(x)	A+B*sqrt(i)	5.756909E-04	0.99975569	-7.205898E+02
arccos(x)	A+B*log(i)^4	5.899332E-04	0.99974345	-7.156533E+02
arcsin(x)	A+B*i	5.821906E-04	0.99975014	-7.183219E+02
arcsin(x)	A+B/i	5.622897E-04	0.99976693	-7.253477E+02
arcsin(x)	A+B*sqrt(i)	5.675281E-04	0.99976257	-7.234745E+02
arcsin(x)	A+B*log(i)^4	5.614364E-04	0.99976764	-7.256545E+02
arctan(x)	A+B*i	1.078296E-09	1	-3.383956E+03
arctan(x)	A+B/i	3.817123E-10	1	-3.565059E+03
arctan(x)	A+B*sqrt(i)	3.370024E-10	1	-3.574378E+03
arctan(x)	A+B*log(i)^4	3.712585E-10	1	-Inf
asinh(x)	A+B*i	2.210007E-03	0.99466435	-2.453225E+03
asinh(x)	A+B/i	3.351385E-03	0.98772988	-1.619636E+03
asinh(x)	A+B*sqrt(i)	2.819393E-03	0.99131617	-1.965686E+03
asinh(x)	A+B*log(i)^4	1.721368E-03	0.99676297	-2.953478E+03
atanh(x)	A+B*i	1.180504E-03	0.99579894	-3.705850E+03
atanh(x)	A+B/i	1.190007E-03	0.99573103	-3.689814E+03
atanh(x)	A+B*sqrt(i)	1.156150E-03	0.99597049	-3.747543E+03
atanh(x)	A+B*log(i)^4	1.160650E-03	0.99593906	-3.739772E+03
Ci(x)	A+B*i	1.720440E-03	0.97055435	-8.716442E+02
Ci(x)	A+B/i	4.918157E-04	0.99759372	-1.362519E+03
Ci(x)	A+B*sqrt(i)	3.140732E-03	0.90186954	-6.357089E+02
Ci(x)	A+B*log(i)^4	1.527311E-03	0.97679417	-9.183203E+02
cosh(x)	A+B*i	3.130648E-04	0.99999985	-3.516650E+03
cosh(x)	A+B/i	5.248782E-06	1	-7.613241E+03
cosh(x)	A+B*sqrt(i)	3.215390E-05	1	-5.797078E+03
cosh(x)	A+B*log(i)^4	1.218171E-04	0.99999998	-4.462427E+03
digamma(x)	A+B/i	9.049193E-04	0.99889225	-4.175243E+03
digamma(x)	A+B*i	1.082618E-03	0.99841448	-3.823473E+03
digamma(x)	A+B*sqrt(i)	1.051702E-03	0.99850374	-3.880317E+03
digamma(x)	A+B*log(i)^4	5.511355E-04	0.9995891	-5.148130E+03
erf(x)	A+B*i	5.759600E-09	1	-6.245605E+03
erf(x)	A+B/i	2.728209E-08	1	-5.589242E+03
erf(x)	A+B*sqrt(i)	1.293968E-08	1	-5.904025E+03
erf(x)	A+B*log(i)^4	3.594503E-08	1	-5.472871E+03
exp(x)	A+B*i	1.213386E-07	1	-4.732594E+03
exp(x)	A+B/i	3.197596E-09	1	-Inf
exp(x)	A+B*sqrt(i)	3.062886E-09	1	-Inf
exp(x)	A+B*log(i)^4	1.453887E-08	1	-5.585528E+03
FresnelCosine(x)	A+B*i	1.028607E-03	0.97980517	-2.136874E+03

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
FresnelCosine(x)	A+B/i	5.437954E-04	0.99435567	-2.711798E+03
FresnelCosine(x)	A+B*sqrt(i)	1.736160E-03	0.9424665	-1.664704E+03
FresnelCosine(x)	A+B*log(i)^4	1.880958E-03	0.93246957	-1.592449E+03
FresnelSine(x)	A+B*i	1.750107E-03	0.93193299	-1.517224E+03
FresnelSine(x)	A+B/i	1.167805E-03	0.96969263	-1.841674E+03
FresnelSine(x)	A+B*sqrt(i)	2.125811E-03	0.89957146	-1.361254E+03
FresnelSine(x)	A+B*log(i)^4	1.715983E-03	0.93456146	-1.533016E+03
J0(x)	A+B*i	1.106625E-03	0.98751387	-1.410869E+03
J0(x)	A+B/i	8.419080E-04	0.99277304	-1.564519E+03
J0(x)	A+B*sqrt(i)	1.021012E-03	0.98937109	-1.456121E+03
J0(x)	A+B*log(i)^4	9.043286E-04	0.99166167	-1.524324E+03
J1(x)	A+B*i	1.779327E-03	0.9764765	-1.207041E+03
J1(x)	A+B/i	1.282912E-03	0.9877712	-1.403957E+03
J1(x)	A+B*sqrt(i)	1.114708E-03	0.99076764	-1.488561E+03
J1(x)	A+B*log(i)^4	1.932535E-04	0.99972251	-2.543473E+03
J2(x)	A+B*i	2.234846E-03	0.95554331	-1.069823E+03
J2(x)	A+B/i	1.084796E-03	0.9895254	-1.504937E+03
J2(x)	A+B*sqrt(i)	6.009292E-04	0.99678569	-1.860520E+03
J2(x)	A+B*log(i)^4	1.025427E-03	0.99064052	-1.538818E+03
J3(x)	A+B*i	1.484136E-03	0.97775692	-1.316245E+03
J3(x)	A+B/i	2.264668E-03	0.94820871	-1.061843E+03
J3(x)	A+B*sqrt(i)	2.290894E-04	0.99947002	-2.441067E+03
J3(x)	A+B*log(i)^4	2.575185E-04	0.99933032	-2.370646E+03
J4(x)	A+B*i	1.501662E-03	0.97506663	-1.309178E+03
J4(x)	A+B/i	5.361936E-05	0.99996821	-3.315293E+03
J4(x)	A+B*sqrt(i)	4.155566E-04	0.9980906	-2.082573E+03
J4(x)	A+B*log(i)^4	3.129656E-03	0.89169973	-8.670986E+02
J5(x)	A+B*i	1.973881E-03	0.95192477	-1.144573E+03
J5(x)	A+B/i	1.874836E-03	0.95662834	-1.175565E+03
J5(x)	A+B*sqrt(i)	1.030541E-03	0.98689582	-1.535824E+03
J5(x)	A+B*log(i)^4	1.005937E-03	0.98751408	-1.550371E+03
ln(x)	A+B*i	7.486808E-05	0.99998532	-8.399562E+03
ln(x)	A+B/i	9.521688E-06	0.99999976	-1.211557E+04
ln(x)	A+B*sqrt(i)	6.884317E-05	0.99998759	-8.550743E+03
ln(x)	A+B*log(i)^4	4.379610E-05	0.99999498	-9.365763E+03
log(x)	A+B*i	3.195652E-05	0.99998582	-9.933697E+03
log(x)	A+B/i	4.219132E-06	0.99999975	-1.358229E+04
log(x)	A+B*sqrt(i)	1.803726E-05	0.99999548	-1.096433E+04
log(x)	A+B*log(i)^4	2.015971E-05	0.99999436	-1.076386E+04
log10Gamma(x)	A+B*i	2.025516E-03	0.99999815	-2.594393E+03
log10Gamma(x)	A+B/i	6.726325E-04	0.9999998	-4.757264E+03
log10Gamma(x)	A+B*sqrt(i)	1.540070E-03	0.99999893	-3.131974E+03
log10Gamma(x)	A+B*log(i)^4	1.851794E-03	0.99999845	-2.770325E+03
Si(x)	A+B*i	1.802197E-03	0.94676388	-8.250731E+02
Si(x)	A+B/i	1.773597E-04	0.9994844	-1.696860E+03
Si(x)	A+B*sqrt(i)	2.056815E-03	0.93065866	-7.753839E+02
Si(x)	A+B*log(i)^4	1.947845E-04	0.99937811	-1.661623E+03
sinh(x)	A+B*i	3.700941E-04	0.99999979	-3.348968E+03
sinh(x)	A+B/i	7.523817E-06	1	-7.252443E+03

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
sinh(x)	A+B*sqrt(i)	4.068446E-05	1	-5.561295E+03
sinh(x)	A+B*log(i)^4	4.596017E-05	1	-5.439122E+03
tan(x)	A+B*i	1.814989E-07	1	-2.349129E+03
tan(x)	A+B/i	2.768043E-07	1	-2.263873E+03
tan(x)	A+B*sqrt(i)	1.664458E-07	1	-2.366618E+03
tan(x)	A+B*log(i)^4	3.252721E-07	1	-2.231280E+03
tanh(x)	A+B*i	3.677636E-07	1	-6.314593E+03
tanh(x)	A+B/i	4.438108E-07	1	-6.201442E+03
tanh(x)	A+B*sqrt(i)	3.864070E-07	1	-6.284823E+03
tanh(x)	A+B*log(i)^4	7.297068E-06	0.99999979	-4.515945E+03
tinv(0.95,x)	A+B*i	1.171604E-03	0.91568507	-3.668492E+03
tinv(0.95,x)	A+B/i	8.425605E-04	0.9563941	-4.315331E+03
tinv(0.95,x)	A+B*sqrt(i)	1.230355E-03	0.90701694	-3.572493E+03
tinv(0.95,x)	A+B*log(i)^4	1.075282E-03	0.9289788	-3.836813E+03
tinv(0.975,x)	A+B*i	2.345354E-03	0.8887015	-2.306740E+03
tinv(0.975,x)	A+B/i	3.285813E-03	0.78154691	-1.645198E+03
tinv(0.975,x)	A+B*sqrt(i)	2.720557E-03	0.8502427	-2.015579E+03
tinv(0.975,x)	A+B*log(i)^4	2.082624E-03	0.91224053	-2.539842E+03
trigamma(x)	A+B*i	1.615518E-03	0.83656569	-3.059384E+03
trigamma(x)	A+B/i	1.804337E-03	0.79612923	-2.840299E+03
trigamma(x)	A+B*sqrt(i)	1.911847E-03	0.77111059	-2.725588E+03
trigamma(x)	A+B*log(i)^4	1.790284E-03	0.79929259	-2.855796E+03

Alternating Sine/Cosine Series of Order 6

The next table shows a summary of results for the Sine series of the order 6:

$$Y = a_0 + a_1 * \sin(S_1 * gx(1,A_1,B_1) + Os_1) + a_2 * \cos(S_2 * gx(2,A_2,B_2) + Os_2)$$

$$+ \ldots + a_6 * \sin(S_6 * gx(6,A_6,B_6) + Os_6) + a_7*x + a_8*x^2$$

The output text files for this series are located in the following folder:

Fourier-Shammas Series Approximations Quadratic Fit with 6 Sine Cosine

The files are named using the following general format:

*fxName_n*_6_run1.txt

Where fxName is the function name and n is the gx series number which varies between 1 and 4. Each output files specifies the function being approximated and the gx series being used. For example, the file for the inverse hyperboplic cosine (acosh) using the first gx series is named acosh_1_6_run1.txt.

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
10^x	A+B*i	1.244195E-08	1	-Inf
10^x	A+B/i	1.413461E-09	1	-Inf
10^x	A+B*sqrt(i)	6.459074E-09	1	-Inf
10^x	A+B*log(i)^4	1.391683E-08	1	-2.863129E+03
acosh(x)	A+B*i	2.673093E-03	0.9912024	-2.057251E+03
acosh(x)	A+B/i	2.042294E-03	0.99486462	-2.590732E+03
acosh(x)	A+B*sqrt(i)	2.349822E-03	0.99320161	-2.312724E+03
acosh(x)	A+B*log(i)^4	2.651669E-03	0.99134285	-2.073200E+03
arccos(x)	A+B*i	4.898429E-04	0.9998212	-7.487972E+02
arccos(x)	A+B/i	4.974471E-04	0.9998156	-7.456855E+02
arccos(x)	A+B*sqrt(i)	5.093359E-04	0.99980668	-7.409146E+02
arccos(x)	A+B*log(i)^4	5.016109E-04	0.9998125	-7.440018E+02
arcsin(x)	A+B*i	4.766002E-04	0.99983073	-7.543334E+02
arcsin(x)	A+B/i	4.893454E-04	0.99982156	-7.490025E+02
arcsin(x)	A+B*sqrt(i)	4.888080E-04	0.99982195	-7.492244E+02
arcsin(x)	A+B*log(i)^4	5.034508E-04	0.99981112	-7.432622E+02
arctan(x)	A+B*i	3.546696E-10	1	-Inf
arctan(x)	A+B/i	2.671005E-10	1	-Inf
arctan(x)	A+B*sqrt(i)	2.544082E-10	1	-Inf
arctan(x)	A+B*log(i)^4	1.099288E-09	1	-3.375273E+03
asinh(x)	A+B*i	2.073691E-03	0.99529753	-2.576647E+03
asinh(x)	A+B/i	6.547087E-04	0.99953126	-4.884743E+03
asinh(x)	A+B*sqrt(i)	2.287617E-03	0.99427725	-2.380089E+03
asinh(x)	A+B*log(i)^4	1.543380E-03	0.99739515	-3.167949E+03
atanh(x)	A+B*i	1.076465E-03	0.99650327	-3.886332E+03
atanh(x)	A+B/i	1.032225E-03	0.99678478	-3.970262E+03
atanh(x)	A+B*sqrt(i)	1.000005E-03	0.99698237	-4.033687E+03
atanh(x)	A+B*log(i)^4	1.043255E-03	0.9967157	-3.949005E+03
Ci(x)	A+B*i	1.758127E-03	0.96908573	-8.589523E+02
Ci(x)	A+B/i	1.596301E-03	0.97451481	-8.968038E+02
Ci(x)	A+B*sqrt(i)	2.287150E-03	0.94768242	-7.558338E+02
Ci(x)	A+B*log(i)^4	2.064581E-03	0.95736932	-7.959664E+02
cosh(x)	A+B*i	6.699631E-04	0.99999932	-2.750242E+03
cosh(x)	A+B/i	8.069879E-06	1	-7.178164E+03
cosh(x)	A+B*sqrt(i)	1.045104E-05	1	-6.919084E+03
cosh(x)	A+B*log(i)^4	1.253599E-04	0.99999998	-4.429628E+03
digamma(x)	A+B/i	1.965277E-03	0.99476983	-2.649591E+03
digamma(x)	A+B*i	7.470073E-04	0.99924435	-4.547460E+03
digamma(x)	A+B*sqrt(i)	1.030025E-03	0.99856331	-3.917141E+03
digamma(x)	A+B*log(i)^4	4.451600E-04	0.99973165	-5.563072E+03
erf(x)	A+B*i	1.725405E-09	1	-6.749603E+03
erf(x)	A+B/i	7.978414E-10	1	-Inf
erf(x)	A+B*sqrt(i)	1.267278E-09	1	-Inf
erf(x)	A+B*log(i)^4	4.672378E-08	1	-5.358014E+03
exp(x)	A+B*i	3.790530E-08	1	-5.196126E+03
exp(x)	A+B/i	1.949972E-09	1	-Inf
exp(x)	A+B*sqrt(i)	3.522171E-09	1	-Inf
exp(x)	A+B*log(i)^4	1.277873E-08	1	-5.633114E+03
FresnelCosine(x)	A+B*i	4.023908E-04	0.99690244	-2.979352E+03

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
FresnelCosine(x)	A+B/i	1.015116E-03	0.98028694	-2.144700E+03
FresnelCosine(x)	A+B*sqrt(i)	1.658709E-04	0.99947366	-3.778717E+03
FresnelCosine(x)	A+B*log(i)^4	1.144674E-03	0.97493391	-2.036355E+03
FresnelSine(x)	A+B*i	8.080840E-04	0.9854512	-2.132890E+03
FresnelSine(x)	A+B/i	5.342270E-04	0.99364134	-2.464794E+03
FresnelSine(x)	A+B*sqrt(i)	2.028422E-04	0.99908329	-3.241444E+03
FresnelSine(x)	A+B*log(i)^4	1.725580E-03	0.93365861	-1.524450E+03
J0(x)	A+B*i	1.065838E-03	0.98837474	-1.427839E+03
J0(x)	A+B/i	7.765933E-04	0.99382826	-1.605768E+03
J0(x)	A+B*sqrt(i)	1.278366E-03	0.98327636	-1.325656E+03
J0(x)	A+B*log(i)^4	9.355746E-04	0.9910427	-1.501099E+03
J1(x)	A+B*i	1.656104E-03	0.97955202	-1.246119E+03
J1(x)	A+B/i	1.385346E-03	0.98569158	-1.353587E+03
J1(x)	A+B*sqrt(i)	1.071393E-03	0.991442	-1.508295E+03
J1(x)	A+B*log(i)^4	1.461512E-03	0.98407499	-1.321367E+03
J2(x)	A+B*i	1.067339E-04	0.99989825	-2.896734E+03
J2(x)	A+B/i	6.068244E-05	0.99996711	-3.236674E+03
J2(x)	A+B*sqrt(i)	1.175986E-03	0.98764817	-1.452220E+03
J2(x)	A+B*log(i)^4	1.212901E-04	0.99986861	-2.819770E+03
J3(x)	A+B*i	1.453723E-04	0.99978586	-2.710740E+03
J3(x)	A+B/i	1.010661E-04	0.9998965	-2.929581E+03
J3(x)	A+B*sqrt(i)	1.006340E-04	0.99989738	-2.932160E+03
J3(x)	A+B*log(i)^4	1.623079E-03	0.97330611	-1.258245E+03
J4(x)	A+B*i	4.632135E-04	0.99761942	-2.013088E+03
J4(x)	A+B/i	2.061462E-04	0.99952851	-2.500469E+03
J4(x)	A+B*sqrt(i)	1.154952E-03	0.98520043	-1.463085E+03
J4(x)	A+B*log(i)^4	1.375376E-03	0.97901234	-1.357935E+03
J5(x)	A+B*i	3.886681E-03	0.8129654	-7.325605E+02
J5(x)	A+B/i	1.075637E-04	0.99985675	-2.892071E+03
J5(x)	A+B*sqrt(i)	3.054530E-03	0.88448113	-8.776002E+02
J5(x)	A+B*log(i)^4	5.100291E-04	0.99677928	-1.955128E+03
ln(x)	A+B*i	4.723527E-05	0.99999415	-9.225499E+03
ln(x)	A+B/i	3.268288E-06	0.99999997	-1.403841E+04
ln(x)	A+B*sqrt(i)	4.248854E-05	0.99999527	-9.416342E+03
ln(x)	A+B*log(i)^4	7.248189E-05	0.99998623	-8.453890E+03
log(x)	A+B*i	3.473324E-05	0.99998323	-9.779512E+03
log(x)	A+B/i	1.765875E-06	0.99999996	-1.514776E+04
log(x)	A+B*sqrt(i)	1.897424E-05	0.999995	-1.086903E+04
log(x)	A+B*log(i)^4	1.520007E-05	0.99999679	-1.126868E+04
log10Gamma(x)	A+B*i	1.818795E-03	0.99999851	-2.801565E+03
log10Gamma(x)	A+B/i	8.249481E-04	0.99999969	-4.352741E+03
log10Gamma(x)	A+B*sqrt(i)	1.009717E-03	0.99999954	-3.956210E+03
log10Gamma(x)	A+B*log(i)^4	1.530267E-03	0.99999894	-3.140466E+03
Si(x)	A+B*i	1.522864E-05	0.99999618	-2.615734E+03
Si(x)	A+B/i	3.486849E-04	0.99799605	-1.438481E+03
Si(x)	A+B*sqrt(i)	3.003528E-04	0.99851309	-1.494585E+03
Si(x)	A+B*log(i)^4	8.595492E-05	0.99987822	-1.965011E+03
sinh(x)	A+B*i	1.359119E-04	0.99999997	-4.348648E+03
sinh(x)	A+B/i	7.141460E-07	1	-9.607820E+03

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
sinh(x)	A+B*sqrt(i)	4.438705E-05	1	-5.469945E+03
sinh(x)	A+B*log(i)^4	1.732925E-04	0.99999995	-4.105188E+03
tan(x)	A+B*i	3.130466E-08	1	-2.699728E+03
tan(x)	A+B/i	1.023249E-07	1	-2.460454E+03
tan(x)	A+B*sqrt(i)	1.206286E-07	1	-2.427240E+03
tan(x)	A+B*log(i)^4	8.562868E-08	1	-2.496466E+03
tanh(x)	A+B*i	1.312597E-06	0.99999999	-5.544529E+03
tanh(x)	A+B/i	1.609004E-07	1	-6.808113E+03
tanh(x)	A+B*sqrt(i)	3.031151E-08	1	-7.813006E+03
tanh(x)	A+B*log(i)^4	1.264622E-06	0.99999999	-5.566944E+03
tinv(0.95,x)	A+B*i	1.555549E-03	0.85121571	-3.108316E+03
tinv(0.95,x)	A+B/i	1.442418E-03	0.8720701	-3.256462E+03
tinv(0.95,x)	A+B*sqrt(i)	1.312958E-03	0.89400352	-3.440965E+03
tinv(0.95,x)	A+B*log(i)^4	1.024942E-03	0.93540658	-3.926847E+03
tinv(0.975,x)	A+B*i	2.946926E-03	0.82410333	-1.854727E+03
tinv(0.975,x)	A+B/i	2.109356E-03	0.90988051	-2.510781E+03
tinv(0.975,x)	A+B*sqrt(i)	2.453195E-03	0.87810579	-2.214502E+03
tinv(0.975,x)	A+B*log(i)^4	2.176264E-03	0.90407269	-2.449513E+03
trigamma(x)	A+B*i	1.642316E-03	0.8309267	-3.022740E+03
trigamma(x)	A+B/i	1.740556E-03	0.81009461	-2.907592E+03
trigamma(x)	A+B*sqrt(i)	1.901491E-03	0.773353	-2.732316E+03
trigamma(x)	A+B*log(i)^4	1.801248E-03	0.79661991	-2.839658E+03

Alternating Sine/Cosine Series of Order 7

The next table shows a summary of results for the Sine series of the order 7:

$Y = a_0 + a_1 * \sin(S_1 * gx(1,A_1,B_1) + Os_1) + a_1 * \cos(S_2 * gx(2,A_2,B_2) + Os_2)$

$+ \ldots + a_7 * \sin(S_7 * gx(7,A_7,B_7) + Os_7) + a_8*x + a_9*x^2$

The output text files for this series are located in the following folder:

Fourier-Shammas Series Approximations Quadratic Fit with 7 Sine Cosine

The files are named using the following general format:

fxName_n_7_run1.txt

Where fxName is the function name and n is the gx series number which varies between 1 and 4. Each output files specifies the function being approximated and the gx series being used. For example, the file for the inverse hyperboplic cosine (acosh) using the first gx series is named acosh_1_7_run1.txt.

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
10^x	A+B*i	8.181023E-09	1	-Inf
10^x	A+B/i	3.285424E-09	1	-Inf
10^x	A+B*sqrt(i)	4.522525E-09	1	-Inf
10^x	A+B*log(i)^4	5.469850E-09	1	-Inf
acosh(x)	A+B*i	1.438578E-03	0.99744938	-3.281220E+03
acosh(x)	A+B/i	1.312787E-03	0.99787594	-3.462578E+03
acosh(x)	A+B*sqrt(i)	2.288571E-03	0.99354484	-2.361032E+03
acosh(x)	A+B*log(i)^4	1.753435E-03	0.99621071	-2.888939E+03
arccos(x)	A+B*i	4.060749E-04	0.99987577	-7.822153E+02
arccos(x)	A+B/i	4.112686E-04	0.99987258	-7.796494E+02
arccos(x)	A+B*sqrt(i)	4.048980E-04	0.99987649	-7.828016E+02
arccos(x)	A+B*log(i)^4	5.117243E-04	0.99980272	-7.355031E+02
arcsin(x)	A+B*i	4.285471E-04	0.99986164	-7.713350E+02
arcsin(x)	A+B/i	4.239081E-04	0.99986462	-7.735336E+02
arcsin(x)	A+B*sqrt(i)	4.200394E-04	0.99986708	-7.753855E+02
arcsin(x)	A+B*log(i)^4	4.496380E-04	0.99984769	-7.616305E+02
arctan(x)	A+B*i	2.533062E-10	1	-Inf
arctan(x)	A+B/i	4.032852E-10	1	-Inf
arctan(x)	A+B*sqrt(i)	7.332569E-11	1	-Inf
arctan(x)	A+B*log(i)^4	3.903202E-10	1	-Inf
asinh(x)	A+B*i	2.474324E-03	0.99329824	-2.218979E+03
asinh(x)	A+B/i	1.245631E-03	0.99830154	-3.593002E+03
asinh(x)	A+B*sqrt(i)	2.283237E-03	0.9942934	-2.379886E+03
asinh(x)	A+B*log(i)^4	1.837380E-03	0.99630449	-2.814827E+03
atanh(x)	A+B*i	9.285458E-04	0.9973956	-4.177927E+03
atanh(x)	A+B/i	9.043716E-04	0.99752945	-4.230686E+03
atanh(x)	A+B*sqrt(i)	8.840704E-04	0.99763912	-4.276093E+03
atanh(x)	A+B*log(i)^4	1.046352E-03	0.99669283	-3.939035E+03
Ci(x)	A+B*i	1.870220E-03	0.96483	-8.305025E+02
Ci(x)	A+B/i	2.723386E-04	0.99925423	-1.585794E+03
Ci(x)	A+B*sqrt(i)	1.001926E-03	0.98990612	-1.075162E+03
Ci(x)	A+B*log(i)^4	9.592205E-04	0.99074825	-1.092237E+03
cosh(x)	A+B*i	7.471693E-04	0.99999915	-2.636872E+03
cosh(x)	A+B/i	2.909783E-08	1	-1.280857E+04
cosh(x)	A+B*sqrt(i)	4.233979E-05	1	-5.513178E+03
cosh(x)	A+B*log(i)^4	2.496715E-05	1	-6.042400E+03
digamma(x)	A+B/i	1.744267E-03	0.99587579	-2.879614E+03
digamma(x)	A+B*i	1.424986E-03	0.99724744	-3.276276E+03
digamma(x)	A+B*sqrt(i)	1.049588E-03	0.99850668	-3.876186E+03
digamma(x)	A+B*log(i)^4	6.433486E-04	0.99943894	-4.836519E+03
erf(x)	A+B*i	4.354973E-09	1	-6.355178E+03
erf(x)	A+B/i	8.856572E-10	1	-Inf
erf(x)	A+B*sqrt(i)	7.285830E-10	1	-Inf
erf(x)	A+B*log(i)^4	2.124335E-08	1	-5.686433E+03
exp(x)	A+B*i	3.860863E-09	1	-Inf
exp(x)	A+B/i	2.493402E-09	1	-Inf
exp(x)	A+B*sqrt(i)	3.186020E-09	1	-Inf
exp(x)	A+B*log(i)^4	4.337997E-09	1	-Inf
FresnelCosine(x)	A+B*i	5.289715E-04	0.99463498	-2.728553E+03

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
FresnelCosine(x)	A+B/i	5.177008E-04	0.99486117	-2.747980E+03
FresnelCosine(x)	A+B*sqrt(i)	1.653610E-03	0.94757086	-1.700470E+03
FresnelCosine(x)	A+B*log(i)^4	1.262375E-04	0.99969445	-4.020912E+03
FresnelSine(x)	A+B*i	1.377384E-04	0.99957623	-3.547773E+03
FresnelSine(x)	A+B/i	2.354983E-04	0.99876121	-3.117622E+03
FresnelSine(x)	A+B*sqrt(i)	7.628567E-05	0.99987001	-4.021651E+03
FresnelSine(x)	A+B*log(i)^4	1.125168E-03	0.97172136	-1.863306E+03
J0(x)	A+B*i	1.387764E-03	0.98021886	-1.275359E+03
J0(x)	A+B/i	1.286622E-03	0.98299715	-1.317887E+03
J0(x)	A+B*sqrt(i)	6.198664E-04	0.99605346	-1.728300E+03
J0(x)	A+B*log(i)^4	7.283580E-04	0.99455109	-1.637656E+03
J1(x)	A+B*i	5.054282E-04	0.9980889	-1.956443E+03
J1(x)	A+B/i	8.069341E-04	0.99512874	-1.674806E+03
J1(x)	A+B*sqrt(i)	1.535222E-03	0.98236777	-1.287607E+03
J1(x)	A+B*log(i)^4	2.573616E-03	0.95044901	-9.765912E+02
J2(x)	A+B*i	1.523549E-03	0.97919683	-1.292201E+03
J2(x)	A+B/i	1.868494E-04	0.9996871	-2.555495E+03
J2(x)	A+B*sqrt(i)	5.367397E-04	0.99741807	-1.920259E+03
J2(x)	A+B*log(i)^4	2.484982E-03	0.94465696	-9.976890E+02
J3(x)	A+B*i	1.035147E-04	0.99989105	-2.911030E+03
J3(x)	A+B/i	2.647208E-04	0.99928748	-2.345775E+03
J3(x)	A+B*sqrt(i)	6.059357E-04	0.99626685	-1.847260E+03
J3(x)	A+B*log(i)^4	7.449382E-04	0.99435762	-1.722930E+03
J4(x)	A+B*i	3.124299E-03	0.89132841	-8.598646E+02
J4(x)	A+B/i	2.633431E-05	0.99999228	-3.735072E+03
J4(x)	A+B*sqrt(i)	8.134095E-05	0.99992634	-3.056151E+03
J4(x)	A+B*log(i)^4	1.158425E-03	0.98506012	-1.457138E+03
J5(x)	A+B*i	8.469667E-05	0.99991088	-3.031814E+03
J5(x)	A+B/i	1.952345E-03	0.95264485	-1.142912E+03
J5(x)	A+B*sqrt(i)	1.111621E-03	0.98464792	-1.481966E+03
J5(x)	A+B*log(i)^4	1.787215E-04	0.99960317	-2.582269E+03
ln(x)	A+B*i	8.197213E-05	0.99998236	-8.228122E+03
ln(x)	A+B/i	3.953556E-06	0.99999996	-1.369136E+04
ln(x)	A+B*sqrt(i)	3.819033E-05	0.99999617	-9.604484E+03
ln(x)	A+B*log(i)^4	4.300623E-05	0.99999515	-9.390474E+03
log(x)	A+B*i	2.214058E-05	0.99999318	-1.058688E+04
log(x)	A+B/i	7.617454E-07	0.99999999	-1.665882E+04
log(x)	A+B*sqrt(i)	9.152686E-06	0.99999883	-1.217870E+04
log(x)	A+B*log(i)^4	7.663881E-06	0.99999918	-1.249861E+04
log10Gamma(x)	A+B*i	1.175981E-03	0.99999938	-3.653096E+03
log10Gamma(x)	A+B/i	2.178210E-03	0.99999786	-2.443718E+03
log10Gamma(x)	A+B*sqrt(i)	2.142722E-03	0.99999793	-2.475947E+03
log10Gamma(x)	A+B*log(i)^4	6.407182E-04	0.99999981	-4.844557E+03
Si(x)	A+B*i	2.210701E-04	0.99918995	-1.605589E+03
Si(x)	A+B/i	1.314400E-04	0.99971364	-1.801082E+03
Si(x)	A+B*sqrt(i)	9.821548E-05	0.99984011	-1.910643E+03
Si(x)	A+B*log(i)^4	1.607537E-03	0.95716724	-8.596128E+02
sinh(x)	A+B*i	9.095214E-05	0.99999999	-4.747042E+03
sinh(x)	A+B/i	4.455213E-08	1	-1.238303E+04

Copyright © 2020 by Namir Clement Shammas

Function	*gx(i,A,B)*	*MSSE*	*RsqrAdj*	*AICc*
sinh(x)	A+B*sqrt(i)	5.514513E-05	1	-5.248408E+03
sinh(x)	A+B*log(i)^4	1.738415E-05	1	-6.405126E+03
tan(x)	A+B*i	7.285245E-08	1	-2.524643E+03
tan(x)	A+B/i	1.257155E-08	1	-2.879553E+03
tan(x)	A+B*sqrt(i)	1.857831E-08	1	-2.800658E+03
tan(x)	A+B*log(i)^4	7.934663E-08	1	-2.507390E+03
tanh(x)	A+B*i	2.306099E-07	1	-6.587288E+03
tanh(x)	A+B/i	2.136347E-08	1	-8.019472E+03
tanh(x)	A+B*sqrt(i)	9.364477E-08	1	-7.129822E+03
tanh(x)	A+B*log(i)^4	8.590631E-07	1	-5.795589E+03
tinv(0.95,x)	A+B*i	1.391936E-03	0.88074531	-3.322317E+03
tinv(0.95,x)	A+B/i	1.228207E-03	0.90715049	-3.567843E+03
tinv(0.95,x)	A+B*sqrt(i)	1.437787E-03	0.87275941	-3.258730E+03
tinv(0.95,x)	A+B*log(i)^4	9.976718E-04	0.93873506	-3.975715E+03
tinv(0.975,x)	A+B*i	2.578174E-03	0.86523086	-2.112968E+03
tinv(0.975,x)	A+B/i	1.469046E-03	0.95624407	-3.216531E+03
tinv(0.975,x)	A+B*sqrt(i)	2.365235E-03	0.88657344	-2.282100E+03
tinv(0.975,x)	A+B*log(i)^4	1.916722E-03	0.92551237	-2.694633E+03
trigamma(x)	A+B*i	2.178301E-03	0.70225834	-2.458908E+03
trigamma(x)	A+B/i	2.148008E-03	0.71048201	-2.486665E+03
trigamma(x)	A+B*sqrt(i)	1.173289E-03	0.9136198	-3.685238E+03
trigamma(x)	A+B*log(i)^4	1.671563E-03	0.82467278	-2.983714E+03

Alternating Cosine/Sine Series of Order 7

The next table shows a summary of results for the Sine series of the order 7:

$$Y = a_0 + a_1 * \cos(S_1 * gx(1,A_1,B_1) + Os_1) + a_2 * \sin(S_2 * gx(2,A_2,B_2) + Os_2)$$

$$+ \ldots + a_7 * \cos(S_7 * gx(7,A_7,B_7) + Os_7) + a_8*x + a_9*x^2$$

The output text files for this series are located in the following folder:

Fourier-Shammas Series Approximations Quadratic Fit with 7 Cosine Sine

The files are named using the following general format:

fxName_n_7_run1.txt

Where fxName is the function name and n is the gx series number which varies between 1 and 4. Each output files specifies the function being approximated and the gx series being used. For example, the file for the inverse hyperboplic cosine (acosh) using the first gx series is named acosh_1_7_run1.txt.

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
10^x	A+B*i	1.649681E-08	1	-2.824648E+03
10^x	A+B/i	2.306767E-09	1	-Inf
10^x	A+B*sqrt(i)	6.165401E-09	1	-Inf
10^x	A+B*log(i)^4	5.237900E-09	1	-Inf
acosh(x)	A+B*i	1.909328E-03	0.99550696	-2.720123E+03
acosh(x)	A+B/i	1.753369E-03	0.99621099	-2.889013E+03
acosh(x)	A+B*sqrt(i)	2.158530E-03	0.99425758	-2.476979E+03
acosh(x)	A+B*log(i)^4	1.572922E-03	0.99695075	-3.104267E+03
arccos(x)	A+B*i	3.977646E-04	0.9998808	-7.863922E+02
arccos(x)	A+B/i	4.434763E-04	0.99985183	-7.644178E+02
arccos(x)	A+B*sqrt(i)	4.051741E-04	0.99987632	-7.826640E+02
arccos(x)	A+B*log(i)^4	4.331320E-04	0.99985867	-7.691873E+02
arcsin(x)	A+B*i	4.224396E-04	0.99986556	-7.742345E+02
arcsin(x)	A+B/i	4.208907E-04	0.99986654	-7.749765E+02
arcsin(x)	A+B*sqrt(i)	4.183794E-04	0.99986813	-7.761854E+02
arcsin(x)	A+B*log(i)^4	4.111916E-04	0.99987263	-7.796866E+02
arctan(x)	A+B*i	3.399907E-10	1	-Inf
arctan(x)	A+B/i	3.083056E-10	1	-Inf
arctan(x)	A+B*sqrt(i)	3.009754E-10	1	-Inf
arctan(x)	A+B*log(i)^4	3.111048E-10	1	-Inf
asinh(x)	A+B*i	2.729825E-03	0.99184272	-2.022242E+03
asinh(x)	A+B/i	1.518089E-03	0.99747727	-3.196985E+03
asinh(x)	A+B*sqrt(i)	1.068905E-03	0.9987493	-3.899322E+03
asinh(x)	A+B*log(i)^4	2.206499E-03	0.99467054	-2.448329E+03
atanh(x)	A+B*i	9.123498E-04	0.99748567	-4.213119E+03
atanh(x)	A+B/i	8.853481E-04	0.99763229	-4.273204E+03
atanh(x)	A+B*sqrt(i)	9.064250E-04	0.99751822	-4.226150E+03
atanh(x)	A+B*log(i)^4	9.505761E-04	0.99727056	-4.131030E+03
Ci(x)	A+B*i	2.359129E-03	0.94403836	-7.394658E+02
Ci(x)	A+B/i	2.399127E-03	0.94212465	-7.328753E+02
Ci(x)	A+B*sqrt(i)	1.271024E-03	0.98375595	-9.819060E+02
Ci(x)	A+B*log(i)^4	1.953823E-03	0.96161538	-8.133597E+02
cosh(x)	A+B*i	1.895094E-04	0.99999995	-4.011469E+03
cosh(x)	A+B/i	1.455439E-07	1	-1.119751E+04
cosh(x)	A+B*sqrt(i)	5.738199E-05	0.99999999	-5.208567E+03
cosh(x)	A+B*log(i)^4	3.447990E-05	1	-5.718939E+03
digamma(x)	A+B/i	4.923954E-04	0.99967134	-5.361167E+03
digamma(x)	A+B*i	1.477552E-03	0.99704062	-3.205203E+03
digamma(x)	A+B*sqrt(i)	8.449074E-04	0.99903232	-4.301794E+03
digamma(x)	A+B*log(i)^4	3.646074E-04	0.9998198	-5.950671E+03
erf(x)	A+B*i	4.360589E-08	1	-5.382953E+03
erf(x)	A+B/i	6.101522E-10	1	-Inf
erf(x)	A+B*sqrt(i)	6.424399E-10	1	-Inf
erf(x)	A+B*log(i)^4	1.517552E-09	1	-6.798981E+03
exp(x)	A+B*i	4.497085E-08	1	-5.123199E+03
exp(x)	A+B/i	8.161468E-10	1	-Inf
exp(x)	A+B*sqrt(i)	3.065304E-09	1	-Inf
exp(x)	A+B*log(i)^4	1.377842E-08	1	-5.598445E+03
FresnelCosine(x)	A+B*i	1.677029E-03	0.94607526	-1.687785E+03

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
FresnelCosine(x)	A+B/i	1.593616E-03	0.95130615	-1.733804E+03
FresnelCosine(x)	A+B*sqrt(i)	5.816431E-04	0.99351336	-2.642933E+03
FresnelCosine(x)	A+B*log(i)^4	1.010570E-03	0.9804188	-2.144657E+03
FresnelSine(x)	A+B*i	3.000501E-04	0.997989	-2.923341E+03
FresnelSine(x)	A+B/i	8.857238E-04	0.98247653	-2.055211E+03
FresnelSine(x)	A+B*sqrt(i)	3.124036E-04	0.99782	-2.890983E+03
FresnelSine(x)	A+B*log(i)^4	1.336563E-03	0.96009727	-1.725227E+03
J0(x)	A+B*i	1.468804E-03	0.97784114	-1.243463E+03
J0(x)	A+B/i	3.751437E-04	0.99855451	-2.010533E+03
J0(x)	A+B*sqrt(i)	1.691508E-03	0.97061214	-1.164124E+03
J0(x)	A+B*log(i)^4	8.115321E-04	0.99323557	-1.576886E+03
J1(x)	A+B*i	2.271255E-05	0.99999614	-3.824141E+03
J1(x)	A+B/i	8.335122E-05	0.99994803	-3.041453E+03
J1(x)	A+B*sqrt(i)	2.805463E-03	0.94111914	-9.246645E+02
J1(x)	A+B*log(i)^4	9.159158E-04	0.9937241	-1.598543E+03
J2(x)	A+B*i	1.437281E-03	0.98148601	-1.327291E+03
J2(x)	A+B/i	2.534802E-04	0.99942416	-2.371896E+03
J2(x)	A+B*sqrt(i)	1.103981E-03	0.98907707	-1.486118E+03
J2(x)	A+B*log(i)^4	1.717378E-03	0.97356691	-1.220108E+03
J3(x)	A+B*i	1.514703E-03	0.97667203	-1.295707E+03
J3(x)	A+B/i	1.474132E-03	0.97790495	-1.312051E+03
J3(x)	A+B*sqrt(i)	8.345811E-04	0.99291794	-1.654526E+03
J3(x)	A+B*log(i)^4	1.548654E-04	0.99975615	-2.668519E+03
J4(x)	A+B*i	2.826140E-03	0.91108023	-9.202439E+02
J4(x)	A+B/i	1.615299E-03	0.97095195	-1.256998E+03
J4(x)	A+B*sqrt(i)	2.515459E-03	0.92955574	-9.903507E+02
J4(x)	A+B*log(i)^4	2.000172E-03	0.95546049	-1.128343E+03
J5(x)	A+B*i	2.581552E-03	0.9172028	-9.747377E+02
J5(x)	A+B/i	8.451451E-04	0.99112606	-1.646954E+03
J5(x)	A+B*sqrt(i)	1.710568E-03	0.96364745	-1.222500E+03
J5(x)	A+B*log(i)^4	1.484023E-04	0.99972639	-2.694182E+03
ln(x)	A+B*i	7.569869E-05	0.99998496	-8.371594E+03
ln(x)	A+B/i	2.262582E-06	0.99999999	-1.469707E+04
ln(x)	A+B*sqrt(i)	6.935453E-05	0.99998737	-8.529322E+03
ln(x)	A+B*log(i)^4	2.344693E-05	0.99999856	-1.048358E+04
log(x)	A+B*i	3.734024E-05	0.9999806	-9.645048E+03
log(x)	A+B/i	8.939457E-07	0.99999999	-1.637044E+04
log(x)	A+B*sqrt(i)	1.136315E-05	0.9999982	-1.178888E+04
log(x)	A+B*log(i)^4	1.708719E-05	0.99999594	-1.105375E+04
log10Gamma(x)	A+B*i	2.261989E-03	0.99999769	-2.369670E+03
log10Gamma(x)	A+B/i	1.332434E-03	0.9999992	-3.408034E+03
log10Gamma(x)	A+B*sqrt(i)	1.298781E-03	0.99999924	-3.458223E+03
log10Gamma(x)	A+B*log(i)^4	7.294406E-04	0.99999976	-4.590109E+03
Si(x)	A+B*i	1.032894E-03	0.98231661	-1.025932E+03
Si(x)	A+B/i	2.304831E-04	0.99911949	-1.589910E+03
Si(x)	A+B*sqrt(i)	7.036918E-04	0.99179234	-1.170233E+03
Si(x)	A+B*log(i)^4	3.030194E-05	0.99998478	-2.352801E+03
sinh(x)	A+B*i	2.445051E-04	0.99999991	-3.756162E+03
sinh(x)	A+B/i	1.205757E-07	1	-1.138614E+04

Copyright © 2020 by Namir Clement Shammas

Function	gx(i,A,B)	MSSE	RsqrAdj	AICc
sinh(x)	A+B*sqrt(i)	7.990838E-05	0.99999999	-4.876754E+03
sinh(x)	A+B*log(i)^4	7.882734E-05	0.99999999	-4.890402E+03
tan(x)	A+B*i	3.210095E-08	1	-2.690188E+03
tan(x)	A+B/i	1.430474E-08	1	-2.853492E+03
tan(x)	A+B*sqrt(i)	1.100231E-08	1	-2.906485E+03
tan(x)	A+B*log(i)^4	1.159012E-07	1	-2.430849E+03
tanh(x)	A+B*i	1.700913E-06	0.99999999	-5.384376E+03
tanh(x)	A+B/i	7.519449E-08	1	-7.261919E+03
tanh(x)	A+B*sqrt(i)	8.201197E-09	1	-8.595764E+03
tanh(x)	A+B*log(i)^4	1.212119E-06	0.99999999	-5.588331E+03
tinv(0.95,x)	A+B*i	1.508350E-03	0.85996354	-3.164727E+03
tinv(0.95,x)	A+B/i	5.196129E-04	0.98338132	-5.255607E+03
tinv(0.95,x)	A+B*sqrt(i)	1.272465E-03	0.90033835	-3.498387E+03
tinv(0.95,x)	A+B*log(i)^4	1.205577E-03	0.91054049	-3.604330E+03
tinv(0.975,x)	A+B*i	2.411143E-03	0.88212762	-2.244384E+03
tinv(0.975,x)	A+B/i	1.120329E-03	0.97455178	-3.748214E+03
tinv(0.975,x)	A+B*sqrt(i)	1.859307E-03	0.929908	-2.754302E+03
tinv(0.975,x)	A+B*log(i)^4	1.606646E-03	0.94766331	-3.040863E+03
trigamma(x)	A+B*i	1.932965E-03	0.76554916	-2.695737E+03
trigamma(x)	A+B/i	1.931477E-03	0.76590982	-2.697263E+03
trigamma(x)	A+B*sqrt(i)	1.782689E-03	0.80058635	-2.856145E+03
trigamma(x)	A+B*log(i)^4	1.423576E-03	0.87283575	-3.301998E+03

Selected Results

The next section displays the output results for selected approximations for the best Fourier-Shammas series model. Most of the output files are located in the folders creaed by the ZIP file that you download with this study. I am limiting the selection of output files to reduce the page count of this report.

Copyright © 2020 by Namir Clement Shammas

RESULTS FOR THE ARC TANGENT

Using Power A+B* i

```
Fitting arctan(x) in range (0.000000, 1.000000)
Fourier Shammas Series factor is A+B*i

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

Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
(Intercept)	-0.74621285303703	0	-Inf	0
x1	-2.37786977743792	0	-Inf	0
x2	0.0470204782842227	0	Inf	0
x3	-0.00444873090406345	0	-Inf	0
x4	-0.00430561816219679	0	-Inf	0
x5	5.38600414170577e-05	0	Inf	0
x6	6.54951469589442e-06	0	Inf	0
x7	8.11167451185342e-07	0	Inf	0
x8	-0.408550792192642	0	-Inf	0
x9	0.0838289832992024	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	5.42875862162705	100	0.0542875862162705		
Model	5.42875862162705	9	0.603195402403005	Inf	0
Residual	0	91	0		

```
Model is ----------------------------------
y = -7.462129e-01 +
    -2.377870e+00 * cos(5.618817e-01 * x + 1.894836) +
    4.702048e-02 * cos(3.046604e+00 * x - 1.884739) +
    -4.448731e-03 * cos(3.445345e+00 * x - 2.000000) +
    -4.305618e-03 * cos(4.900385e+00 * x + 2.000000) +
    5.386004e-05 * cos(8.709440e+00 * x - 1.970228) +
    6.549515e-06 * cos(9.458717e+00 * x - 1.485253) +
    8.111675e-07 * cos(1.331840e+01 * x - 1.978779) +
    (-4.085508e-01)*x + (8.382898e-02)*x^2
List of factors: [0.561882, 3.046604, 3.445345, 4.900385, 8.709440, 9.458717,
13.318404]
List of offsets: [1.894836, -1.884739, -2.000000, 2.000000, -1.970228, -1.485253, -
1.978779]
```

```
Fitting arctan(x) in range (0.000000, 1.000000)
MSS of errors squared = 8.853758e-11
R-Squared =   1.00000000
R-Squared Adjusted =   1.00000000
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

Comments on the Above Output

The above output (typical of all output text files) shows the following information:

- The name of the fitting function and the range of x values used to fit that function.
- The expressin of the Shammas sequence used.
- The linear regression coefficients for the trigonometric terms, the linear term (the one before last), and the quadratic term (the last coefficient). These coefficients are accompanied by their standard errors, student-t values, and p-values.
- The ANOVA table for the regression that includes the F statistic and its p-value.
- The model used in fitting, showing the various terms. This is the model you employ to calculate the function values for x values of your choosing. Notice that the term inside the trigonometric functions shows the values of C_i * gx(i,A_i,B_i).
- The list of factors and offset values.
- The mean square of the sum of errors squared.
- The regression coefficient of determination and its adjusted value. The latter is used to select the best models.
- The AIC and AICc statistics.

Using Power A+B/i

```
Fitting arctan(x) in range (0.000000, 1.000000)
Fourier Shammas Series factor is A+B/i

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

Estimated Coefficients:
                         Estimate            SE    tStat    pValue
                   _____________________     __    _____    ______

    (Intercept)        -35827.3742510553     0     -Inf       0
    x1                  65358.5992398841     0      Inf       0
    x2                   6.3863788020988     0      Inf       0
    x3               -0.0028130923603192     0     -Inf       0
    x4             -2.01558532692475e-05     0     -Inf       0
    x5               -0.0089560963102451 8   0     -Inf       0
    x6              0.000202082709054578     0      Inf       0
    x7                0.0801983614091124     0      Inf       0
    x8                  2039.94053953075     0      Inf       0
    x9                  24.2091614085919     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       5.42875862169747    100    0.0542875862169747
    Model       5.42875862169747      9      0.60319540241083    Inf       0
    Residual                   0     91                     0

Model is ----------------------------------
y = -3.582737e+04 +
    6.535860e+04 * cos(3.739083e-02 * x + 0.990578) +
    6.386379e+00 * cos(8.096353e-01 * x - 2.000000) +
    -2.813092e-03 * cos(5.556750e+00 * x - 1.212363) +
    -2.015585e-05 * cos(9.734901e+00 * x + 2.000000) +
    -8.956096e-03 * cos(4.896318e+00 * x + 2.000000) +
    2.020827e-04 * cos(7.727486e+00 * x - 2.000000) +
    8.019836e-02 * cos(2.443548e+00 * x - 1.544172) +
    (2.039941e+03)*x + (2.420916e+01)*x^2
List of factors: [0.037391, 0.809635, 5.556750, 9.734901, 4.896318, 7.727486,
2.443548]
List of offsets: [0.990578, -2.000000, -1.212363, 2.000000, 2.000000, -2.000000, -
1.544172]
Fitting arctan(x) in range (0.000000, 1.000000)
MSS of errors squared = 4.191796e-10
```

```
R-Squared =   1.00000000
R-Squared Adjusted =   1.00000000
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

Using Power A+B*sqrt(i)

```
Fitting arctan(x) in range (0.000000, 1.000000)
Fourier Shammas Series factor is A+B*sqrt(i)

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

Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
(Intercept)	1.15309860476907	0	Inf	0
x1	-3.29606970317624	0	-Inf	0
x2	2.37611995358013e-05	0	Inf	0
x3	0.000613553295357533	0	Inf	0
x4	0.0467246722155204	0	Inf	0
x5	-0.0437913745109221	0	-Inf	0
x6	-0.0592730985622017	0	-Inf	0
x7	8.35137474319301e-06	0	Inf	0
x8	-0.76311212296137	0	-Inf	0
x9	-0.171348572771056	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	5.42875862162704	100	0.0542875862162704		
Model	5.42875862162704	9	0.603195402403005	Inf	0
Residual	0	91	0		

```
Model is ---------------------------------
y = 1.153099e+00 +
    -3.296070e+00 * cos(5.310271e-01 * x + 1.213677) +
    2.376120e-05 * cos(7.966672e+00 * x - 1.630726) +
    6.135533e-04 * cos(5.960658e+00 * x - 1.693003) +
    4.672467e-02 * cos(4.414620e+00 * x - 0.612839) +
    -4.379137e-02 * cos(3.371415e+00 * x + 1.791864) +
    -5.927310e-02 * cos(4.227249e+00 * x - 0.609095) +
    8.351375e-06 * cos(1.072711e+01 * x - 2.000000) +
    (-7.631121e-01)*x + (-1.713486e-01)*x^2
```

```
List of factors: [0.531027, 7.966672, 5.960658, 4.414620, 3.371415, 4.227249,
10.727109]
List of offsets: [1.213677, -1.630726, -1.693003, -0.612839, 1.791864, -0.609095, -
2.000000]
Fitting arctan(x) in range (0.000000, 1.000000)
MSS of errors squared = 4.381130e-10
R-Squared =   1.00000000
R-Squared Adjusted =   1.00000000
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

Using Power A+B*log10(i)^4

```
Fitting arctan(x) in range (0.000000, 1.000000)
Fourier Shammas Series factor is A+B*log(i)^4

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

Estimated Coefficients:
                          Estimate            SE    tStat    pValue
                   _____________________     __    _____    ______

    (Intercept)        1.77297048986775      0      Inf        0
    x1                 4.31215098630207      0      Inf        0
    x2               0.0360664626630873      0      Inf        0
    x3             0.000806556605606578      0      Inf        0
    x4              -0.0271056136369307      0     -Inf        0
    x5             2.28709166276624e-07      0      Inf        0
    x6            -1.23077099417518e-10      0     -Inf        0
    x7              0.00203617254259285      0      Inf        0
    x8               -1.09188334837511      0     -Inf        0
    x9              -0.245032476644573      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       5.42875862162704    100    0.0542875862162704
    Model       5.42875862162704      9     0.603195402403005     Inf       0
    Residual                   0     91                     0

Model is ----------------------------------
y = 1.772970e+00 +
    4.312151e+00 * cos(5.159192e-01 * x - 2.000000) +
    3.606646e-02 * cos(1.650917e+00 * x + 0.531335) +
    8.065566e-04 * cos(5.161159e+00 * x + 0.081547) +
```

```
    -2.710561e-02 * cos(3.756705e+00 * x + 1.117339) +
    2.287092e-07 * cos(1.609019e+01 * x + 1.696301) +
    -1.230771e-10 * cos(2.218875e+02 * x + 1.934000) +
    2.036173e-03 * cos(6.225904e+02 * x + 0.747029) +
    (-1.091883e+00)*x + (-2.450325e-01)*x^2
List of factors: [0.515919, 1.650917, 5.161159, 3.756705, 16.090192, 221.887544,
622.590376]
List of offsets: [-2.000000, 0.531335, 0.081547, 1.117339, 1.696301, 1.934000,
0.747029]
Fitting arctan(x) in range (0.000000, 1.000000)
MSS of errors squared = 4.119201e-10
R-Squared =   1.00000000
R-Squared Adjusted =   1.00000000
Particle swarm AICc =  -1.000000e+99
AIC =  -Inf
AICc =  -Inf
```

RESULTS FOR INEVRSE HYPERBOLIC SINE

Using Power A+B* i

```
Fitting asinh(x) in range (0.000000, 100.000000)
Fourier Shammas Series factor is A+B*i

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

Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
(Intercept)	-8.22708760744686	0.139172217102184	-59.1144394962562	0
x1	12.0971889700263	0.176436616704757	68.5639364206884	0
x2	-4.38950287615152e-06	0.00348208456564342	-0.00126059628748287	0.998994443655562
x3	-0.00190344028973075	0.00348481035742913	-0.546210580920962	0.585044135946494
x4	-0.000991661743228925	0.0034852538274343	-0.28453070919054	0.776063147314719
x5	-0.000550585844363366	0.00348592101544781	-0.15794558795895	0.874531851653824
x6	1.97836619244864	0.0362231741940534	54.6160361830857	4.05344710487582e-301
x7	0.000997701855201349	0.0034861428442575	0.286190755735899	0.774791838251294
x8	0.815412927255248	0.0102275285306607	79.7272698688398	0
x9	-0.00719059540912014	9.27258184070653e-05	-77.5468529978728	0

```
Number of observations: 1001, Error degrees of freedom: 991
Root Mean Squared Error: 0.078
R-squared: 0.993,  Adjusted R-Squared: 0.993
F-statistic vs. constant model: 1.68e+04, p-value = 0
```

	SumSq	DF	MeanSq	F	pValue
Total	923.674148234163	1000	0.923674148234163		
Model	917.652235446301	9	101.961359494033	16779.3375324603	0
Residual	6.02191278786268	991	0.00607660220773227		

```
Model is ---------------------------------
```

Copyright © 2020 by Namir Clement Shammas

```
y = -8.227088e+00 +
    1.209719e+01 * cos(4.552863e-02 * x + 0.631189) +
    -4.389503e-06 * cos(2.964559e+01 * x - 1.928927) +
    -1.903440e-03 * cos(3.953658e+00 * x - 0.008555) +
    -9.916617e-04 * cos(1.613023e+01 * x + 0.050956) +
    -5.505858e-04 * cos(2.037750e+02 * x - 1.807781) +
    1.978366e+00 * cos(3.140808e+02 * x - 1.989125) +
    9.977019e-04 * cos(2.707339e+02 * x + 2.000000) +
    (8.154129e-01)*x + (-7.190595e-03)*x^2
List of factors: [0.045529, 29.645592, 3.953658, 16.130230, 203.774952, 314.080752,
270.733880]
List of offsets: [0.631189, -1.928927, -0.008555, 0.050956, -1.807781, -1.989125,
2.000000]
Fitting asinh(x) in range (0.000000, 100.000000)
MSS of errors squared = 2.451507e-03
R-Squared =   0.99348048
R-Squared Adjusted =   0.99342127
Particle swarm AICc =  -2.237526e+03
AIC =  -2.257748e+03
AICc =  -2.237526e+03
```

Using Power A+B/i

```
Fitting asinh(x) in range (0.000000, 100.000000)
Fourier Shammas Series factor is A+B/i

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

Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
(Intercept)	1.13357180969837	0.0195025355357425	58.1243299170436	1.52172218919104e-321
x1	11.8952748221678	0.210344974807451	56.551266951144	1.81440088753565e-312
x2	0.0890802822740878	0.00598467309750505	14.8847365299242	2.24558635287477e-45
x3	-0.0129772815621515	0.00365271665932132	-3.55277531013388	0.000399146400344406
x4	-8.54645059791905	0.249361592711638	-34.2733237503908	2.02165619053123e-170
x5	0.306268962830935	0.00989747888067861	30.9441390603843	1.15125603699023e-147
x6	-0.00218507216809253	0.00365490651581663	-0.597846253696672	0.550079115942994
x7	-8.36291435483144	0.245404169985619	-34.0781265262181	4.33195557078828e-169
x8	0.517864378481569	0.00783953243295406	66.0580695227035	0
x9	-0.00432913521480011	6.82155332170356e-05	-63.4626017072456	0

```
Number of observations: 1001, Error degrees of freedom: 991
Root Mean Squared Error: 0.0817
R-squared: 0.993,  Adjusted R-Squared: 0.993
F-statistic vs. constant model: 1.53e+04, p-value = 0
```

	SumSq	DF	MeanSq	F	pValue
Total	923.674148234167	1000	0.923674148234167		
Model	917.060665603505	9	101.895629511501	15268.5921299202	0

```
    Residual    6.61348263066247     991    0.00667354453144548

Model is ----------------------------------
y = 1.133572e+00 +
    1.189527e+01 * cos(2.748053e-02 * x + 1.556756) +
    8.908028e-02 * cos(2.057289e-01 * x + 1.731856) +
    -1.297728e-02 * cos(1.688628e+00 * x - 0.944415) +
    -8.546451e+00 * cos(2.732437e-01 * x + 1.259797) +
    3.062690e-01 * cos(3.124091e-01 * x - 0.771170) +
    -2.185072e-03 * cos(5.380368e+00 * x - 1.692055) +
    -8.362914e+00 * cos(2.717337e-01 * x - 1.801491) +
    (5.178644e-01)*x + (-4.329135e-03)*x^2
List of factors: [0.027481, 0.205729, 1.688628, 0.273244, 0.312409, 5.380368,
0.271734]
List of offsets: [1.556756, 1.731856, -0.944415, 1.259797, -0.771170, -1.692055, -
1.801491]
Fitting asinh(x) in range (0.000000, 100.000000)
MSS of errors squared = 2.569100e-03
R-Squared =   0.99284003
R-Squared Adjusted =   0.99277500
Particle swarm AICc =  -2.143727e+03
AIC =  -2.163949e+03
AICc =  -2.143727e+03
```

Using Power A+B*sqrt(i)

```
Fitting asinh(x) in range (0.000000, 100.000000)
Fourier Shammas Series factor is A+B*sqrt(i)

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

Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
(Intercept)	-15.5311795415065	0.321028451797776	-48.3794487825958	3.64878475193708e-263
x1	-50.2524221956959	1.00111964834956	-50.1962200807283	1.94972658326642e-274
x2	-0.00164036125678566	0.00338875962579627	-0.484059490174142	0.628450600423854
x3	-0.000934009017413908	0.00339006993075894	-0.275513200757133	0.782979421168161
x4	-0.000494435165354794	0.00339079045781435	-0.145817080561651	0.884095445429432
x5	0.00127361144644004	0.00338842444548139	0.375871283817014	0.707093034664779
x6	7.37769900606826	0.165659765922282	44.5352494915962	9.49539329595438e-239
x7	-1.92712375600411	0.0473604545839076	-40.6905671183933	1.28201121631004e-213
x8	2.11155360008308	0.0398222426367182	53.0244772838613	1.2416872011493e-291
x9	-0.0166113514348011	0.000318726995894896	-52.1178050455408	3.62711737074114e-286

```
Number of observations: 1001, Error degrees of freedom: 991
Root Mean Squared Error: 0.0758
R-squared: 0.994,  Adjusted R-Squared: 0.994
F-statistic vs. constant model: 1.77e+04, p-value = 0
```

SumSq	DF	MeanSq	F	pValue

```
    Total       923.674148234172    1000      0.923674148234172
    Model       917.978915549037       9       101.997657283226    17748.1209207661       0
    Residual    5.69523268513512     991    0.00574695528267924

Model is ----------------------------------
y = -1.553118e+01 +
    -5.025242e+01 * cos(2.940150e-02 * x - 2.000000) +
    -1.640361e-03 * cos(9.914193e+00 * x - 1.583289) +
    -9.340090e-04 * cos(3.627407e+01 * x + 0.145157) +
    -4.944352e-04 * cos(7.336078e+01 * x - 1.928117) +
    1.273611e-03 * cos(1.537630e+01 * x + 2.000000) +
    7.377699e+00 * cos(6.287940e+01 * x + 2.000000) +
    -1.927124e+00 * cos(1.255873e+02 * x + 0.586321) +
    (2.111554e+00)*x + (-1.661135e-02)*x^2
List of factors: [0.029401, 9.914193, 36.274065, 73.360776, 15.376297, 62.879397,
125.587323]
List of offsets: [-2.000000, -1.583289, 0.145157, -1.928117, 2.000000, 2.000000,
0.586321]
Fitting asinh(x) in range (0.000000, 100.000000)
MSS of errors squared = 2.384085e-03
R-Squared =   0.99383415
R-Squared Adjusted =   0.99377816
Particle swarm AICc =  -2.293357e+03
AIC =  -2.313580e+03
AICc =  -2.293357e+03
```

Using Power A+B*log10(i)^4

```
Fitting asinh(x) in range (0.000000, 100.000000)
Fourier Shammas Series factor is A+B*log(i)^4

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

Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
(Intercept)	0.0864564637751587	0.0152970386351939	5.65184319899986	2.07455210795982e-08
x1	-83.7478248248121	1.15396891393238	-72.5737269121268	0
x2	4.25404405715162	0.0662774138098049	64.1854262654148	0
x3	-0.000228341292020096	0.0022474745480113	-0.101599055803389	0.919095481180545
x4	-1.68674351433134	0.0242952084382217	-69.4270032142518	0
x5	0.000329720992440556	0.00224886977907423	0.146616311672919	0.883464697583105
x6	-0.188422425683647	0.00546137724654905	-34.500899164713	5.68572839093041e-172
x7	-0.000261423755151524	0.00224543270538602	-0.116424667069495	0.907339575305916
x8	3.08667350703146	0.0407927647075799	75.6671809120579	0
x9	-0.0277969642209294	0.000370244029224736	-75.0774138860097	0

```
Number of observations: 1001, Error degrees of freedom: 991
Root Mean Squared Error: 0.0503
R-squared: 0.997,  Adjusted R-Squared: 0.997
```

```
F-statistic vs. constant model: 4.05e+04, p-value = 0
```

	SumSq	DF	MeanSq	F	pValue
Total	923.674148234168	1000	0.923674148234168		
Model	921.168563735846	9	102.352062637316	40481.9291233329	0
Residual	2.5055844983217	991	0.00252833955431049		

```
Model is ---------------------------------
y = 8.645646e-02 +
    -8.374782e+01 * cos(2.888622e-02 * x - 1.587254) +
    4.254044e+00 * cos(7.503927e-02 * x + 1.985265) +
    -2.283413e-04 * cos(1.534994e+01 * x - 1.948664) +
    -1.686744e+00 * cos(6.289818e+01 * x + 1.991869) +
    3.297210e-04 * cos(4.090767e+01 * x - 1.592696) +
    -1.884224e-01 * cos(6.298016e+01 * x - 1.992837) +
    -2.614238e-04 * cos(9.928852e+01 * x - 0.767854) +
    (3.086674e+00)*x + (-2.779696e-02)*x^2
List of factors: [0.028886, 0.075039, 15.349940, 62.898179, 40.907671, 62.980159,
99.288522]
List of offsets: [-1.587254, 1.985265, -1.948664, 1.991869, -1.592696, -1.992837, -
0.767854]
Fitting asinh(x) in range (0.000000, 100.000000)
MSS of errors squared = 1.581322e-03
R-Squared =   0.99728737
R-Squared Adjusted =   0.99726274
Particle swarm AICc =  -3.115286e+03
AIC =  -3.135508e+03
AICc =  -3.115286e+03
```

Results for Inverse Hyperbolic Tangent

Using Power A+B* i

```
Fitting atanh(x) in range (0.000000, 0.999000)
Fourier Shammas Series factor is A+B*i

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

Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
(Intercept)	-2963489.59902211	100157.310951306	-29.5883502749279	2.32791023857592e-138
x1	-7243418.33663088	244805.054690933	-29.5885162411198	2.32184454110916e-138
x2	-2355.8199551441	82.0554341191472	-28.7101028765916	2.27391996007918e-132
x3	49396.233880831	1678.21467948799	29.4337991942137	2.64214401990379e-137
x4	-10936.1911233375	375.910248472527	-29.0925591089245	5.62354153918773e-135
x5	102.671534338819	3.5969061982178	28.5444014052107	3.05662530260025e-131
x6	27.8553603815327	1.06055533676271	26.26488162943	8.47100086086732e-116
x7	4.50857909172674	0.185640621242698	24.2865977367767	1.39095831136712e-102
x8	2087978.86166647	70560.4570967056	29.591345458616	2.22084021766651e-138

Copyright © 2020 by Namir Clement Shammas

x9	169767.534061254	5737.00186662383	29.591681858936	2.20912663226061e-138

Number of observations: 1000, Error degrees of freedom: 990
Root Mean Squared Error: 0.0282
R-squared: 0.998, Adjusted R-Squared: 0.998
F-statistic vs. constant model: 4.64e+04, p-value = 0

	SumSq	DF	MeanSq	F	pValue
Total	334.064284076852	999	0.334398682759611		
Model	333.274311494294	9	37.0304790549215	46406.8944591197	0
Residual	0.789972582558107	990	0.000797952103594048		

```
Model is ---------------------------------
y = -2.963490e+06 +
    -7.243418e+06 * cos(3.290311e-01 * x - 1.992125) +
    -2.355820e+03 * cos(5.819333e+00 * x + 0.769080) +
    4.939623e+04 * cos(2.455114e+00 * x - 1.604969) +
    -1.093619e+04 * cos(4.372506e+00 * x - 1.998611) +
    1.026715e+02 * cos(6.318779e+00 * x + 2.000000) +
    2.785536e+01 * cos(1.124406e+01 * x - 1.354821) +
    4.508579e+00 * cos(1.397960e+01 * x + 0.659540) +
    (2.087979e+06)*x + (1.697675e+05)*x^2
List of factors: [0.329031, 5.819333, 2.455114, 4.372506, 6.318779, 11.244064,
13.979602]
List of offsets: [-1.992125, 0.769080, -1.604969, -1.998611, 2.000000, -1.354821,
0.659540]
Fitting atanh(x) in range (0.000000, 0.999000)
MSS of errors squared = 8.888040e-04
R-Squared =   0.99763527
R-Squared Adjusted =   0.99761377
Particle swarm AICc =  -4.265413e+03
AIC =  -4.285635e+03
AICc =  -4.265413e+03
```

Using Power A+B/i

Fitting atanh(x) in range (0.000000, 0.999000)
Fourier Shammas Series factor is A+B/i

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

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	-459141.618974218	15326.0847439012	-29.9581808822327	6.93708457995025e-141
x1	-205647.183517335	6984.58346881489	-29.4430132355808	2.28596543573987e-137
x2	-2158641.12628399	72318.9537385505	-29.8488987283745	3.86989080176224e-140
x3	72053.5084476782	2473.08891167021	29.135025476709	2.88666347538295e-135
x4	214253.205955638	7280.34456890022	29.4289925329734	2.84945607497481e-137
x5	-713739.946996199	24094.8269386304	-29.6221238199426	1.36892042534449e-138

Copyright © 2020 by Namir Clement Shammas

x6	-196.323534380059	7.09913204873977	-27.6545827056295	3.41574305096511e-125
x7	-34541.663760373	1188.58723234879	-29.0611095427254	9.21446706000875e-135
x8	2379849.90441287	79579.6666427851	29.9052509869824	1.59501614324498e-140
x9	-472670.504027492	15806.2250674119	-29.9040727315726	1.6248524524073e-140

```
Number of observations: 1000, Error degrees of freedom: 990
Root Mean Squared Error: 0.0283
R-squared: 0.998,  Adjusted R-Squared: 0.998
F-statistic vs. constant model: 4.62e+04, p-value = 0
```

	SumSq	DF	MeanSq	F	pValue
Total	334.06428414772	999	0.33439868283055		
Model	333.270593278509	9	37.0300659198343	46188.9719067702	0
Residual	0.793690869210764	990	0.000801707948697741		

```
Model is ---------------------------------
y = -4.591416e+05 +
    -2.056472e+05 * cos(4.243841e+00 * x + 1.986805) +
    -2.158641e+06 * cos(1.486956e+00 * x - 1.995111) +
    7.205351e+04 * cos(5.461311e+00 * x + 1.325618) +
    2.142532e+05 * cos(4.300869e+00 * x + 0.000144) +
    -7.137399e+05 * cos(3.322833e+00 * x + 0.015346) +
    -1.963235e+02 * cos(9.221793e+00 * x + 0.448467) +
    -3.454166e+04 * cos(5.709036e+00 * x + 0.483731) +
    (2.379850e+06)*x + (-4.726705e+05)*x^2
List of factors: [4.243841, 1.486956, 5.461311, 4.300869, 3.322833, 9.221793,
5.709036]
List of offsets: [1.986805, -1.995111, 1.325618, 0.000144, 0.015346, 0.448467,
0.483731]
Fitting atanh(x) in range (0.000000, 0.999000)
MSS of errors squared = 8.908933e-04
R-Squared =   0.99762414
R-Squared Adjusted =   0.99760254
Particle swarm AICc =  -4.260717e+03
AIC =  -4.280939e+03
AICc =  -4.260717e+03
```

Using Power A+B*sqrt(i)

```
Fitting atanh(x) in range (0.000000, 0.999000)
Fourier Shammas Series factor is A+B*sqrt(i)

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

Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
(Intercept)	-36523969.5830005	1217558.93104627	-29.997701673146	3.72540078261413e-141
x1	48508228.9888844	1617069.75751583	29.997610655587	3.73073883092914e-141
x2	3055.51389129597	106.108392734696	28.7961565767536	5.8951865487612e-133

Copyright © 2020 by Namir Clement Shammas

x3	12.8985356412853	0.495386234726625	26.0373315548488	2.87502632411885e-114
x4	87821.9090834991	3040.23373062359	28.8865649370602	1.42688753998299e-133
x5	-11750.3152582352	397.692582488887	-29.5462268486327	4.51378652689811e-138
x6	-73158.6593233049	2535.80360221085	-28.8502860629747	2.52138832279344e-133
x7	-16376.2138213782	560.526135375747	-29.2157899299386	8.11913028406068e-136
x8	6556800.77748411	218569.672907022	29.998675892576	3.66874036998501e-141
x9	707612.320613513	23587.767399853	29.9991223678898	3.64306218645376e-141

Number of observations: 1000, Error degrees of freedom: 990
Root Mean Squared Error: 0.0284
R-squared: 0.998, Adjusted R-Squared: 0.998
F-statistic vs. constant model: 4.58e+04, p-value = 0

	SumSq	DF	MeanSq	F	pValue
Total	334.064283905159	999	0.334398682587747		
Model	333.264117838088	9	37.0293464264542	45814.30589321	0
Residual	0.8001660670717	990	0.000808248552597677		

```
Model is ----------------------------------
y = -3.652397e+07 +
    4.850823e+07 * cos(2.055901e-01 * x + 0.717979) +
    3.055514e+03 * cos(6.778647e+00 * x + 0.216840) +
    1.289854e+01 * cos(1.204885e+01 * x + 2.000000) +
    8.782191e+04 * cos(6.507535e+00 * x - 1.879067) +
    -1.175032e+04 * cos(4.173313e+00 * x + 0.266847) +
    -7.315866e+04 * cos(6.611994e+00 * x - 1.866573) +
    -1.637621e+04 * cos(5.457946e+00 * x - 1.837601) +
    (6.556801e+06)*x + (7.076123e+05)*x^2
List of factors: [0.205590, 6.778647, 12.048849, 6.507535, 4.173313, 6.611994,
5.457946]
List of offsets: [0.717979, 0.216840, 2.000000, -1.879067, 0.266847, -1.866573, -
1.837601]
Fitting atanh(x) in range (0.000000, 0.999000)
MSS of errors squared = 8.945200e-04
R-Squared =   0.99760475
R-Squared Adjusted =   0.99758298
Particle swarm AICc =  -4.252592e+03
AIC =  -4.272814e+03
AICc =  -4.252592e+03
```

Using Power A+B*log10(i)^4

```
Fitting atanh(x) in range (0.000000, 0.999000)
Fourier Shammas Series factor is A+B*log(i)^4

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

Estimated Coefficients:
```

	Estimate	SE	tStat	pValue

(Intercept)	5345.20825435954	159.997567309847	33.4080595363564	1.81549324454138e-164
x1	116829.910672447	3474.60414703472	33.6239484351482	6.10372547981761e-166
x2	-564.451430903998	17.2209719757321	-32.7769786571528	3.70810096975099e-160
x3	81644.7089584919	2431.68751308626	33.5753292802289	1.31025883572722e-165
x4	-1049.4838000613	31.7093061889756	-33.0970281660145	2.41401718469421e-162
x5	8.00046593484574	0.26713562997494	29.9490784347871	8.00501395231147e-141
x6	0.297682733429423	0.0127611063985882	23.3273451479376	2.89089662527152e-96
x7	-0.00951447323650643	0.00146462653401654	-6.49617702228451	1.30244392942235e-10
x8	62314.9649619853	1844.73798904375	33.7798458816839	5.27256847764188e-167
x9	-30718.7567169622	909.443492673862	-33.7775320450595	5.46770741743355e-167

Number of observations: 1000, Error degrees of freedom: 990
Root Mean Squared Error: 0.0306
R-squared: 0.997, Adjusted R-Squared: 0.997
F-statistic vs. constant model: 3.95e+04, p-value = 0

	SumSq	DF	MeanSq	F	pValue
Total	334.064284132172	999	0.334398682814987		
Model	333.137405248475	9	37.0152672498305	39536.0334795357	0
Residual	0.926878883697327	990	0.000936241296663967		

```
Model is ---------------------------------
y = 5.345208e+03 +
    1.168299e+05 * cos(2.187033e+00 * x + 1.345509) +
    -5.644514e+02 * cos(5.582842e+00 * x + 1.882440) +
    8.164471e+04 * cos(2.505247e+00 * x - 1.973921) +
    -1.049484e+03 * cos(4.574900e+00 * x - 1.966343) +
    8.000466e+00 * cos(1.102834e+01 * x - 0.314863) +
    2.976827e-01 * cos(1.761544e+01 * x - 0.175154) +
    -9.514473e-03 * cos(3.691126e+01 * x + 0.884907) +
    (6.231496e+04)*x + (-3.071876e+04)*x^2
List of factors: [2.187033, 5.582842, 2.505247, 4.574900, 11.028345, 17.615445,
36.911262]
List of offsets: [1.345509, 1.882440, -1.973921, -1.966343, -0.314863, -0.175154,
0.884907]
Fitting atanh(x) in range (0.000000, 0.999000)
MSS of errors squared = 9.627455e-04
R-Squared =   0.99722545
R-Squared Adjusted =   0.99720022
Particle swarm AICc =  -4.105588e+03
AIC =  -4.125811e+03
AICc =  -4.105588e+03
```

CONCLUSIONS

Overall, the Hybrid Quadratic Fourier-Shammas series improved model fitting compared to the Fourier-Shammas series. The various HQFS series perform better than their counterpart in the Fourier-Shammas series.

DOCUMENT HISTORY

Date	***Version***	***Comments***
11/7/2020	1.00.00	Initial release.

Copyright © 2020 by Namir Clement Shammas