5 Introduction

Functions:

eulsum: sum a series Euler van Wijngaarden algorithm
ddpoly: fast evaluation of specified derivatives
poldiv: polynomials divide one by another
chebft: fit a Chebyshev polynomial to a function
chevev: Chebyshev polynomial evaluation
chint: integrate a function already Chebyshev fitted
chder: integrate a function already Chebyshev fitted
chebpc: polynomial coefficients from a Chebyshev fit
pcshft: polynomial coefficients of a shifted polynomial

eulsum

Description:

This function incorporates into a sum the next term of an alternating series.

eulsum(sum, term, jterm) $\longrightarrow$ sum

Where:

sum is a real number on input being the current sum and on output the current sum plus the next term **term**,
term is a real number representing the next term to be added to **sum**,
jterm is an integer representing the ranking order of the term to be added, where the first term has ranking order 1.

Pre-variables:

eulsum_jmax [100] an integer being an upper bound for the value of *jterm*.

Example:

In this example the first five terms of the series expansion $ln(1+x) = \sum_1^{\mathrm{inf}} \frac{x^j}{j}$ are summed using `eulsum`.

```
-> sum : 0.0;
-> term(x,j) := (-1)^(j+1)*x^j/j;
-> x : 1.0;
-> for n to 5 do
  sprog(
         sum:eulsum(sum,term(x,n),n),
         print(sum));
0.5000000
0.6250000
0.7083333
0.6979167
0.6947917
-> ln(1+x)?
0.6931472
```

ddpoly

Description:

Given the coefficients of a polynomial and an evaluation point this function computes the value of the polynomial and a given number of derivatives at that point using synthetic division.

ddpoly(c, x, nd) ⟶ pd

Where:

c is a real list representing the coefficients with c_1 being the constant term,
x is a real number being the evaluation point,
nd is an integer such that the function will compute the value of the polynomial at x and $nd - 1$ derivatives,
pb is a real list being the value and derivatives where pd_1 is the value of the polynomial at x.

Example:

This example uses `ddpoly` to compute the value of the polynomial $g(x) = (x - 1)^5$ and of its derivative `g1` at $x = 3$. The computed value is checked against the actual value.

```
-> g(x) := -1 + 5*x - 10*x^2 + 10*x^3 - 5*x^4 + x^5;
-> g1(x) := 5 - 20*x + 30*x^2 - 20*x^3 + 5*x^4;
-> c : [-1,5,-10,10,-5,1];
-> x : 3; nd : 2;
-> ddpoly(c,x,nd)?
[32.0000000, 80.0000000]
-> [g(x),g1(x)]?
[32, 80]
-> x : -2; nd : 4;
-> ddpoly(c,x,nd)?
[-243.0000000, 405.0000000, -540.0000000, 540.0000000]
-> [g(x),g1(x)]?
[-243, 405]
```

poldiv

Description:

This function computes the quotient and remainder when one polynomal with real coefficients is divided by another.

$$\mathbf{poldiv(u,\ v)} \longrightarrow \mathbf{[q,\ r]}$$

Where:

u is a real list being the coefficients of a polynomial such that u_1 is the constant term,
v is is a real list being the coefficients of a polynomial such that v_1 is the constant term,
q is a real list representing the quotient when u is divided by v and such that if u has length n and v has length nv then q has length $nv - 2$,
r is a real list representing the remainder when u is divided by v and such that if u has length n and v has length nv then r has length $n - nv$.

Example:

The polynomial $(x-1)^5$ is divided by the polynomial $(x+1)^3$.

```
-> u : [-1,5,-10,10,-5,1];
-> v : [1,3,3,1];
-> res : poldiv(u,v)?
[[31.0000000, -8.0000000, 1.0000000], [-32.0000000, -80.0000000,
 -80.0000000]]
```

chebft

Description:

This function computes the coefficients of a Chebyshev approximation to a function over a real interval. Given a function func, lower and upper limits of the interval $[a, b]$ and a maximum degree n, this routine computes the n coefficients c_k of a Chebyshev fit for $func$ such that

$$func(x) \approx [\sum_{k=1}^{n} c_k T_{k-1}(y)] - c_1/2$$

where y and x are related by the equation

$$y = \frac{x - \frac{1}{2}(b+a)}{\frac{1}{2}(b-a)}$$

This routine is to be used with moderately large n (say 30 or 50) the array of c's should be truncated at the smaller value m such that c_{m+1} and subsequent values are negligible.

chebft(int, n, func) $\longrightarrow$ **c**

Where:

int is real interval of the form $[a, b]$ being the domain,
func is the name of a real valued function of a real variable,
n is a positive integer being the order of the Chebyshev approximation,
c is a real list being the Chebyshev coefficients.

Example:

A list of 40 terms of the Chebyshev expansion of $f(x) = x^2(x^2-2)sin(x)$ on $[-\frac{\pi}{2}, \frac{\pi}{2}]$ is computed.

```
-> f(x) := x^2*(x^2-2)*sin(x);
-> pi : float(%pi)?
3.1415927
-> c : chebft([-pi/2,pi/2],40,f)?
[0.0000000, 0.0271960, 0.0000000, 0.7849335, -0.0000000, 0.3907023, -0.0000000,
-0.0512295, -0.0000000, 0.0016891, 0.0000000, -0.0000256, 0.0000000, 0.0000002,
-0.0000000, -0.0000000, -0.0000000, 0.0000000, -0.0000000, -0.0000000,
-0.0000000, 0.0000000, -0.0000000, -0.0000000, -0.0000000, 0.0000000,
0.0000000, 0.0000000, 0.0000000, 0.0000000, -0.0000000, 0.0000000, 0.0000000,
0.0000000, -0.0000000, 0.0000000, -0.0000000, 0.0000000, 0.0000000, 0.0000000]
```

chebev

Description:

This function performs an evaluation of a function represented by a Chebyshev approximation.

chebev(int, c, x) $\longrightarrow$ value

Where:

int is a real interval of the form $[a, b]$ representing the domain of the original function,
c is a real list of Chebyshev coefficients of length m, being the first m elements of c output from **chebft** (which must have been called with the same $[a, b]$),
x is a real number representing the point at which the interpolant is to be evaluated,
value is a real number representing the interpolated value.

Example:

The Chebyshev series computed in the example with the function `chebft` given above is evaluated at $-\frac{\pi}{3}$ and $\frac{\pi}{6}$ and the values compared with the values of the original function.

```
-> chebev([-pi/2,pi/2],c,-pi/3)?
0.8579402
-> f(-pi/3)?
0.8579402
-> chebev([-pi/2,pi/2],c,pi/6)?
-0.2365750
-> f(pi/6)?
-0.2365750
```

chint

Description:

The function integrates a function represented by a Chebyshev approximation and returns the Chebyshev coefficients of the integral.

chint(int, c) $\longrightarrow$ cint

Where:

int is a real interval of the form $[a, b]$ representing the function's domain,
c is a real list of Chebyshev coefficients of length n, being the first n elements of c output from *chebft* (which must have been called with the same $[a, b]$),
cint is a real list representing the Chebyshev coefficients of the integral.

Example:

The Chebyshev series representation for the function used in the example with `chebft` above is integrated using `chint` and the result compared with the true integral using `chebev`.

```
-> pi : float(%pi);
-> f(x) := x^2*(x^2-2)*sin(x);
-> fint(x) := 4*x*(x^2-7)*sin(x) - (x^4-14*x^2+28)*cos(x);
-> c : chebft([-pi/2,pi/2],40,f);
-> res : chint([-pi/2,pi/2],c)?
[0.3347385, -0.0000000, -0.2975628, 0.0000000, 0.0774071, 0.0000000, 0.0578487,
-0.0000000, -0.0051953, -0.0000000, 0.0001347, -0.0000000, -0.0000017,
0.0000000, 0.0000000, -0.0000000, -0.0000000, 0.0000000, 0.0000000, 0.0000000,
-0.0000000, -0.0000000, 0.0000000, 0.0000000, -0.0000000, -0.0000000,
0.0000000, 0.0000000, -0.0000000, 0.0000000, 0.0000000, -0.0000000, -0.0000000,
0.0000000, 0.0000000, -0.0000000, 0.0000000, -0.0000000, 0.0000000, 0.0000000]
-> Compare with true vales at, say, -pi/3 & pi/6: #
-> chebev([-pi/2,pi/2],res,-pi/3)?
0.1391432
-> fint(-pi/3) - fint(-pi/2)?
0.1391432
-> chebev([-pi/2,pi/2],res,pi/6)?
0.4460291
-> fint(pi/6) - fint(-pi/2)?
0.4460291
```

chder

Description:

The function differentiates a function represented by a Chebyshev approximation and returns the Chebyshev coefficients of the derivative.

chder(int, c) $\longrightarrow$ **cder**

Where:

int is a real interval of the form $[a, b]$ representing the original function's domain,
c is a real list of Chebyshev coefficients of length n, being the first n elements of c output from **chebft** (which must have been called with the same $[a, b]$),
cder is a real list representing the Chebyshev coefficients of the derivative.

Example:

The Chebyschev representation for the function used in `chebev` above is differentiated using `chder` and the resulting series evaluated at two points.

```
-> fder(x) := 4*x*(x^2-1)*sin(x) + x^2*(x^2-2)*cos(x);
-> res : chder([-pi/2,pi/2],c)?
[5.0825492, -0.0000000, 5.0479223, -0.0000000, 2.0496972, -0.0000000,
-0.4375910, -0.0000000, 0.0190010, -0.0000000, -0.0003547, -0.0000000,
0.0000037, -0.0000000, -0.0000000, -0.0000000, 0.0000000, -0.0000000,
-0.0000000, -0.0000000, 0.0000000, 0.0000000, 0.0000000, 0.0000000,
0.0000000, 0.0000000, 0.0000000, 0.0000000, -0.0000000, 0.0000000,
0.0000000, -0.0000000, 0.0000000, 0.0000000, 0.0000000, 0.0000000,
0.0000000, 0.0000000, -0.0000000, 0.0000000]
-> Compare with true vales at, say, -pi/3 & pi/6: #
-> chebev([-pi/2,pi/2],res,-pi/3)?
-0.1448236
-> fder(-pi/3)?
-0.1448236
-> chebev([-pi/2,pi/2],res,pi/6)?
-1.1698623
-> fder(pi/6)?
-1.1698623
```

chebpc

Description:

This function computes Chebyshev polynomial coefficients using Clenshaw's recurrence applied algebraically.

chebpc(c) $\longrightarrow$ d

Where:

c is a real list being the Chebyshev coefficients,
d is a real list such that $\sum_{k=1}^{n} d_k y^{k-1} = \sum_{k=1}^{n} c_k T_{k-1}(y)$.

Example:

The function `chebpc` computes the Chebyshev polynomial coefficients for the function used with `chebev` above.

```
-> c : chebft([-pi/2,pi/2],40,f);
-> d : chebpc(c)?
[-0.0000000, 0.0000000, 0.0000000, -7.7515692, -0.0000000, 12.7508202,
0.0000000, -4.3259408, -0.0000001, 0.5082776, 0.0000014, -0.0292936,
-0.0000148, 0.0009828, 0.0001067, 0.0000674, -0.0005567, -0.0004758, 0.0021585,
0.0018562, -0.0063308, -0.0054514, 0.0141963, 0.0122402, -0.0244482,
-0.0211449, 0.0322763, 0.0280750, -0.0323749, -0.0284024, 0.0242270, 0.0214967,
-0.0130957, -0.0117835, 0.0048293, 0.0044170, -0.0010870, -0.0010127,
0.0001126, 0.0001071]
```

pcshft

Description:

This function computes a polynomial coefficient shift transforming a polynomial in y to a polynomial in x where

$$y = \frac{x - \frac{1}{2}(b+a)}{\frac{1}{2}(b-a)}$$

pcshift(int, d) $\longrightarrow$ **g**

Where:

int is a real interval of the form $[a, b]$,
d is a real list representing the coefficients of the polynomial in y in increasing order,
g is a real list such that $\sum_{k=1}^{n} d_k y^{k-1} = \sum_{k=1}^{n} g_k x^{k-1}$.

Example:

The polynomial coefficients relating to [-1,1] computed by `chebpc` above are translated to $[a, b]$ using the function `pcshft`.

```
-> pcshft([-pi/2,pi/2],d)?
[-0.0000000, 0.0000000, 0.0000000, -2.0000000, -0.0000000, 1.3333333,
0.0000000, -0.1833333, -0.0000000, 0.0087302, 0.0000000, -0.0002039,
-0.0000001, 0.0000028, 0.0000002, 0.0000001, -0.0000004, -0.0000002, 0.0000006,
0.0000003, -0.0000008, -0.0000004, 0.0000007, 0.0000004, -0.0000005,
-0.0000003, 0.0000003, 0.0000001, -0.0000001, -0.0000001, 0.0000000, 0.0000000,
-0.0000000, -0.0000000, 0.0000000, 0.0000000, -0.0000000, -0.0000000,
0.0000000, 0.0000000]
```