Sheet to test the DDL for computing 2F1 using double double precision.

Note that all testdata are first converted to their hardware presentation to allow reasonable check against Maple's exactness.

```
> restart; kernelopts(version);
```

Maple 11.00, IBM INTEL NT, Feb 16 2007 Build ID 277223

```
> Digits:=18;
```

Digits := 18

```
> # this assumes, that the DLL is in the directory of this sheet
  currentdir(): theDLL:=cat(%,`\\hyp2f1.dll`);
```

theDLL := "D:_Work\vc2005\hyp2f1_www\release\hyp2f1.dll"

## Functions

```
> MPL_2F1:=proc(a,b,c, Z)
  local z;
  #z:= piecewise(Im(evalf(Z))=0.0,evalf(Z)-1e-306*I,Z); # only for Maple versions <= 10
  evalf(hypergeom([a,b],[c],z));
  fnormal(%,Digits,Float(1,-Digits+0));
  simplify(%,zero);
  end proc:
```

```
> MPL_2F1:=proc(a,b,c, z) evalf(hypergeom([a,b],[c],z)); end proc;
```

MPL_2F1 := **proc**(a, b, c, z) evalf( hypergeom( [ a, b ], [ c ], z ) ) **end proc**

```
> Gamma:=GAMMA;
  invSinn:=proc(z) evalf(Pi / sin(Pi*z)) end proc;
```

Γ := Γ

invSinn := **proc**(z) evalf( π / sin( π*z ) ) **end proc**

```
> hyp2f1_mpl:=define_external(
    'hyp2f1_mpl',
  'C',
    'inputAarry'::ARRAY(1..8,float[8]),
    'result'::ARRAY(1..2,datatype=float[8], NO_COPY),
    LIB=theDLL);

  hyp2f1_DLL:= proc(a,b,c,z)
  local A, R, result;
  A:=Array(1..8, 0, order= Fortran_order, datatype= float[8]);
  R:=Array(1..2, 0, order= Fortran_order, datatype= float[8]);: #print(R);
  #R:=Array(1..2,0,datatype=float[8],order=C_order):
  R[1], R[2] := 0.0, 0.0;
  A[1] := Re(evalf(a)); A[2] := Im(evalf(a));
  A[3] := Re(evalf(b)); A[4] := Im(evalf(b));
  A[5] := Re(evalf(c)); A[6] := Im(evalf(c));
  A[7] := Re(evalf(z)); A[8] := Im(evalf(z));

  evalf(
  proc(AA,RR)
    hyp2f1_mpl(AA, RR);
  end proc(A,R)
  );

  result := copy(R[1]+R[2]*I);
  simplify(%,zero);
  end proc;

  test_hyp2f1_DLL:= proc(a,b,c,z)
  local m,d;
  m:=evalf(MPL_2F1(a,b,c,z));
  d:=evalf(hyp2f1_DLL(a,b,c,z));
  print( 'MPL' = m );
  print( 'DLL' = d );
  `error abs, rel` = m-d, abs((m-d)/m);
  end proc;
```

hyp2f1_mpl := **proc**(inputAarry::rtable( datatype = float[ 8 ] ), result::rtable( datatype = float[ 8 ] ))
**option** call_external, define_external( hyp2f1_mpl, C, inputAarry::ARRAY( 1 .. 8, float[ 8 ] ),
result::ARRAY( 1 .. 2, datatype = float[ 8 ], NO_COPY ), LIB = "D:_Work\vc2005\hyp2f1_www\release\hyp2f1.dll" );
call_external( Array(1..10, [...], datatype = integer[4], readonly), args)
**end proc**
hyp2f1_DLL := **proc**(a, b, c, z)
**local** A, R, result;
A := Array( 1 .. 8, 0, order = Fortran_order, datatype = float[ 8 ] );
R := Array( 1 .. 2, 0, order = Fortran_order, datatype = float[ 8 ] );
R[ 1 ], R[ 2 ] := 0., 0.;

```
        A[1] := ℜ(evalf(a));
        A[2] := ℑ(evalf(a));
        A[3] := ℜ(evalf(b));
        A[4] := ℑ(evalf(b));
        A[5] := ℜ(evalf(c));
        A[6] := ℑ(evalf(c));
        A[7] := ℜ(evalf(z));
        A[8] := ℑ(evalf(z));
        evalf((proc(AA, RR) hyp2f1_mpl(AA, RR) end proc)(A, R));
        result := copy(R[1] + R[2]*I);
        simplify(%, zero)
    end proc
    test_hyp2f1_DLL := proc(a, b, c, z)
    local m, d;
        m := evalf(MPL_2F1(a, b, c, z));
        d := evalf(hyp2f1_DLL(a, b, c, z));
        print('MPL' = m);
        print('DLL' = d);
        `error abs, rel` = m – d, abs((m – d) / m)
    end proc
```

## pre-process data

```
> preproc:=proc(assignmentList)
  local tstData, TstData, i;
  tstData:=evalf(assignmentList); # work with floating point numbers only
  tstData:= map( _x -> fnormal(_x,Digits,Float(1,-Digits+0)), tstData): # to avoid Maple's symbolic
  exactness
  tstData:= simplify(%,zero); # kick off symbolic zeros
  # compare the hardware representations only
  TstData:=(eval([a,b,c, z], (tstData)));
  for i from 1 to nops(TstData) do
    TstData[i]; evalhf(%); TstData[i]:=%;
  end do;
  TstData:=op(TstData);
  end proc;
```

```
preproc := proc(assignmentList)
local tstData, TstData, i;
    tstData := evalf(assignmentList);
    tstData := map(_x → fnormal(_x, Digits, Float(1, –Digits)), tstData);
    tstData := simplify(%, zero);
    TstData := eval([a, b, c, z], tstData);
    for i to nops(TstData) do TstData[i]; evalhf(%); TstData[i] := % end do;
    TstData := op(TstData)
end proc
```

## test routine

```
> testRoutine := proc(A,B,C,Z)
  local g,h, a,b,c,z;
  Digits:=Digits+6;
  a,b,c,z := evalhf(A), evalhf(B), evalhf(C), evalhf(Z);
  #gc(); forget(hyp2f1_DLL);
  g:=evalf(hyp2f1_DLL(a,b,c,z));

  h:=MPL_2F1(a,b,c,z);
  Digits:=Digits-6;
  #h:=evalhf(h);
  h:=evalf(h);
  print( 'DLL' = evalf(g) );
  print( 'MPL' = evalf(h) );
  print( `error, absolute resp. relative` = abs(h-g), abs(h-g)/abs(h) );
  #print( `normalized error (abs resp rel)` = fnormal(abs(h-g),Digits-0),
  fnormal(abs(h-g)/abs(h),Digits-0) );
  if 4 < nargs then
    print(`check with doubled digits:` );
    Digits := 2*Digits;
    h:=MPL_2F1(TstData);
    print( 'MPL' = evalf[Digits/2](h) );
```

```
    end if;
    end proc;
  testRoutine := proc(A, B, C, Z)
  local g, h, a, b, c, z;
      Digits := Digits + 6;
      a, b, c, z := evalhf(A), evalhf(B), evalhf(C), evalhf(Z);
      g := evalf(hyp2f1_DLL(a, b, c, z));
      h := MPL_2F1(a, b, c, z);
      Digits := Digits – 6;
      h := evalf(h);
      print('DLL' = evalf(g));
      print('MPL' = evalf(h));
      print(`error, absolute resp. relative` = abs(h – g), abs(h – g) / abs(h));
      if 4 < nargs then print(`check with doubled digits:`); Digits := 2*Digits; h := MPL_2F1(TstData); print('MPL' = evalf[Digits / 2](h)) end if
  end proc
> 
> # Gosper's example: z = intersection of unit circle and diagonal
  tstData:=[a=1/2, b=1/6, c=1/3, z= 'exp(I*Pi/3)'];
  TstData:=preproc(tstData);
  testRoutine(TstData);
```

$$\text{tstData} := \left[ a = \frac{1}{2}, b = \frac{1}{6}, c = \frac{1}{3}, z = e^{(1/3\,I\,\pi)} \right]$$

TstData := 0.500000000000000000, 0.166666666666666658, 0.333333333333333315, 0.500000000000000000 + 0.866025403784438596 I

DLL = 0.957626387164578374 + 0.256595217091482420 I

MPL = 0.957626387164578201 + 0.256595217091482375 I

error, absolute resp. relative = $0.178756818051787888\ 10^{-15}$, $0.180306045756238554\ 10^{-15}$

```
> 
```

## tests

```
> Digits:=15;
```

Digits := 15

### 1

```
> # Winding number = 2
  tstData:=[a=-3*Pi, b=sqrt(2), c=exp(1), z= 1+1/exp(Pi) - 1e-290*I];
  TstData:=preproc(tstData);
  testRoutine(TstData);
```

$$\text{tstData} := \left[ a = -3\,\pi, b = \sqrt{2}, c = e, z = \frac{1}{e^{\pi}} + 1. - 0.1\ 10^{-289}\ I \right]$$

TstData := -9.42477796076937047, 1.41421356237310003, 2.71828182845904998, 1.04321391826377008

DLL = 0.0560182538639228 – $0.172417988082839\ 10^{-15}$ I

MPL = 0.0560182538639225 – $0.172417988082840\ 10^{-15}$ I

error, absolute resp. relative = $0.3\ 10^{-15}$, $0.535539720193259\ 10^{-14}$

### 2

```
> #tstData:=[a=1.75, b=1.76, c=2.2, z= 1.01*I];
  tstData:=[a=1, b=1, c=2.2, z= 1.9];
  TstData:=preproc(tstData);
  testRoutine(TstData);
```

tstData := [a = 1, b = 1, c = 2.2, z = 1.9]

TstData := 1., 1., 2.20000000000000016, 1.89999999999999990

DLL = 0.315767495606574 – 1.70873729899200 I

MPL = 0.315767495606591 – 1.70873729899200 I

error, absolute resp. relative = $0.17\ 10^{-13}$, $0.978322316747915\ 10^{-14}$

### 3

```
> tstData:=[a=1.756, b=4.76, c=+2.20,  z= 0.85]; # <-----
  #tstData:=[a=1.756, b=4.76, c=+2.20,  z= 2*(2^(1/2)-1)];
  #tstData:=[a=2.75, b=4.8, c=-26.2,  z= 0.5*I];
  #tstData:=[a=2.75, b=4.8, c=-6.2,  z= 1.5*I];
  #tstData:=[a=-3.45, b= -1.4, c= -6.2,  z= 1.5*I]; # a+c,b+c,c

  TstData:=preproc(tstData);

  testRoutine(TstData);
```

tstData := [ a = 1.756, b = 4.76, c = 2.20, z = 0.85 ]

TstData := 1.75600000000000000, 4.75999999999999980, 2.20000000000000016, 0.849999999999999976

DLL = 2443.22685352900

MPL = 2443.22685352899

error, absolute resp. relative = 0.1 $10^{-10}$, 0.409294781021092 $10^{-14}$

4

```
> # case W2: winding = 1
  #tstData:=[a=1.756, b=4.76, c=+2.20,  z= 0.85];
  tstData:=[a=1.756, b=4.76, c=+2.20,  z= 1.85*I+0.1*I];
  TstData:=preproc(tstData);
  testRoutine(TstData, Digits);
```

tstData := [ a = 1.756, b = 4.76, c = 2.20, z = 1.95 I]

TstData := 1.75600000000000000, 4.75999999999999980, 2.20000000000000016, 1.94999999999999996 I

DLL = -0.0291190062044294 – 0.0104896762749080 I

MPL = -0.0291190062043906 – 0.0104896762749491 I

error, absolute resp. relative = 0.565212349475841 $10^{-13}$, 0.182616602975501 $10^{-11}$

check with doubled digits:

MPL = -0.0291190062043906 – 0.0104896762749491 I

5

```
> # case z large

  tstData:=[a=1.756, b=4.76, c=+2.20,  z= 2.85+0*I]; # <--- Vorsicht für relle z
  #tstData:=[a=1.756, b=4.76, c=+2.20,  z= -2.85+0.1*I*0];
  #tstData:=[a=1.756, b=1.76, c=+2.20,  z= 12.85+0.1*I];
  #tstData:=[a=1.756, b=1.76, c=+2.20,  z= -12.85+0.1*I];
  TstData:=preproc(tstData);
  testRoutine(TstData, Digits, check);
```

tstData := [ a = 1.756, b = 4.76, c = 2.20, z = 2.85 ]

TstData := 1.75600000000000000, 4.75999999999999980, 2.20000000000000016, 2.85000000000000008

DLL = 0.0231635195043357 – 0.0188908693591645 I

MPL = 0.0231635195042665 – 0.0188908693592299 I

error, absolute resp. relative = 0.952144946948730 $10^{-13}$, 0.318549400396187 $10^{-11}$

check with doubled digits:

MPL = 0.0231635195042665 – 0.0188908693592299 I

6

```
> tstData:=[a=1.756, b=.556, c=.996,  z= .778162990325950434e-1-.605574311537704596e-3*I];
  #tstData:=[a=.56, b=1.76, c=1.004,  z= .778162990325950434e-1-.605574311537704596e-3*I];
  TstData:=preproc(tstData);
  testRoutine(TstData);
```

tstData := [ a = 1.756, b = 0.556, c = 0.996, z = 0.0778162990325950434 – 0.00060557431153704596 I]

TstData :=

1.75600000000000000, 0.556000000000000050, 0.995999999999999996, 0.0778162990325950016 – 0.00060557431153705030 I

DLL = 1.08323344382660 – 0.000706748046851244 I

MPL = 1.08323344382660 – 0.000706748046851244 I

error, absolute resp. relative = 0., 0.

7

```
> tstData:=[a=1.756, b=2.76, c=+2.20,  z= 4];
  TstData:=preproc(tstData);
  testRoutine(TstData);
```

tstData := [ a = 1.756, b = 2.76, c = 2.20, z = 4 ]

TstData := 1.75600000000000000, 2.75999999999999980, 2.20000000000000016, 4.

DLL = 0.0393278956028085 – 0.0210256495322391 I

MPL = 0.0393278956028130 – 0.0210256495322348 I

error, absolute resp. relative = 0.622414652783817 $10^{-14}$, 0.139568842542105 $10^{-12}$

8

```
> #hypergeom([1,2],[3],z); tmp:=simplify(%);
  #tstData:=[a=1, b=1 + 1e-6, c=+2.20,  z= 4];
  #tstData:=[a=1.756, b=2.756 + 1e-6, c=+2.20,  z= 4]; # ok
  tstData:=[a=1, b=2 + 1e-8, c=3,  z= 2];
  #tstData:=[a=1, b=2 + 1e-18, c=3,  z= 2];
  TstData:=preproc(tstData);
```

```
testRoutine(TstData, check);
```

tstData := [ a = 1, b = 2.00000001, c = 3, z = 2 ]

TstData := 1., 2.00000000999999994, 3., 2.

DLL = -0.999999927835502 – 1.57079632516199 I

MPL = -1.00000001467401 – 1.57079631108693 I

error, absolute resp. relative = $0.879717783477728\ 10^{-7}$, $0.472434202317717\ 10^{-7}$

check with doubled digits:

MPL = -1.00000001467401 – 1.57079631108693 I

⊏ >

9

```
> #tstData:=[a=1, b=1 + 1e-6, c=+2.20,  z= 4];
  #tstData:=[a=1.756, b=2.756 + 1e-6, c=+2.20,  z= 4];
  tstData:=[a=1, b=2 + 1e-12*0, c=3,  z= 2];
  #tstData:=[a=1, b=2.1, c=4.1,  z= 3/2];
  TstData:=preproc(tstData);

  testRoutine(TstData);
```

tstData := [ a = 1, b = 2., c = 3, z = 2 ]

TstData := 1., 2., 3., 2.

DLL = -0.999999999999804 – 1.57079632679494 I

MPL = -1.00000000000000 – 1.57079632679490 I

error, absolute resp. relative = $0.200039996000800\ 10^{-12}$, $0.107427333452461\ 10^{-12}$

⊏ >

10

```
> #tstData:=[a=1, b=1 + 1e-6, c=+2.20,  z= 4];
  #tstData:=[a=1.756, b=2.756 + 1e-6, c=+2.20,  z= 4];
  tstData:=[a=1 + 1e-8, b=2 + 2e-8, c=-3 + 3e-8,  z= 2];
  #tstData:=[a=1 + 1e-3, b=2 + 2e-3, c=-3 + 3e-2,  z= 3];
  TstData:=preproc(tstData);

  testRoutine(TstData);
```

tstData := [ a = 1.00000001, b = 2.00000002, c = -2.99999997, z = 2 ]

TstData := 1.00000000999999994, 2.00000001999999988, -2.99999997000000018, 2.

DLL = $-0.106666672227503\ 10^{11} + 0.0000110990435797413$ I

MPL = $-0.106666669030000\ 10^{11} – 0.110214535291490\ 10^{-7}$ I

error, absolute resp. relative = 319.750300000000, $0.299765899608312\ 10^{-7}$

⊏ >

11

```
> #tstData:=[a=1, b=1 + 1e-6, c=+2.20,  z= 4];
  #tstData:=[a=1.756, b=2.756 + 1e-6, c=+2.20,  z= 4];
  #tstData:=[a=1, b=2 + 1e-12*0, c=3,  z= 2];
  tstData:=[a=1, b=2, c=3,  z= 2];
  TstData:=preproc(tstData);

  testRoutine(TstData);
```

tstData := [ a = 1, b = 2, c = 3, z = 2 ]

TstData := 1., 2., 3., 2.

DLL = -0.999999999999804 – 1.57079632679494 I

MPL = -1.00000000000000 – 1.57079632679490 I

error, absolute resp. relative = $0.200039996000800\ 10^{-12}$, $0.107427333452461\ 10^{-12}$

12

```
> #tstData:=[a=1, b=1 + 1e-6, c=+2.20,  z= 4];
  #tstData:=[a=1.756, b=2.756 + 1e-6, c=+2.20,  z= 4];
  #tstData:=[a=1, b=2 + 1e-12, c=3,  z= 1+ 1.0*I/10^(Digits-1)];
  tstData:=[a=1, b=2 + 1/10^(Digits-7), c=3,  z= 1 + I/10^(Digits-7)];
  TstData:=preproc(tstData);

  testRoutine(TstData);
```

tstData := $\left[ a = 1, b = \frac{200000001}{100000000}, c = 3, z = 1 + \frac{1}{100000000} I \right]$

TstData := 1., 2.00000000999999994, 3., 1. + $0.100000000000000002\ 10^{-7}$ I

DLL = 34.8413644943036 + 3.14159248404937 I

MPL = 34.8413645708639 + 3.14159248404937 I

error, absolute resp. relative = $0.765603\ 10^{-7}$, $0.218851801197873\ 10^{-8}$

13

```
> Epsilon:=10^(-8);
  #tstData:=[a=1, b=2 + Epsilon, c=3,  z= 5/2];
  tstData:=[a=1, b=2 + Epsilon, c=3,  z= 3/2];
  TstData:=preproc(tstData);

  testRoutine(TstData);


  `check:`;
  hypergeom([1,2+delta],[3],z): convert(%,StandardFunctions): theSol:=simplify(%);

  Digits:=2*Digits:
  theSol:
  subs(delta=Epsilon,%): eval(%,tstData): f:=evalf(%):
  Digits:= floor(Digits/2);
  f:=evalf(f);
  Epsilon:='Epsilon':
```

$$E := \frac{1}{100000000}$$

$$tstData := \left[ a = 1, b = \frac{200000001}{100000000}, c = 3, z = \frac{3}{2} \right]$$

TstData := 1., 2.00000000999999994, 3., 1.50000000000000000

DLL = -0.717202535056624 – 2.79252679462198 I

MPL = -0.717202540726474 – 2.79252679462198 I

error, absolute resp. relative = $0.5669850\ 10^{-8}$, $0.196654328448446\ 10^{-8}$

check:

$$theSol := -\frac{2\,(-(1-z)^{(-\delta)} + 1 + \delta\, z)}{z^2\,(1+\delta)\,\delta}$$

Digits := 15

f := -0.717202540726474 – 2.79252679462198 I

```
> 
```

14

```
> #tstData:=[a=1, b=2, c=3,  z= 5/2];
  #tstData:=[a=1, b=4, c=2,  z= 3/2]; # <--- bug
  #tstData:=[a=4, b=4, c=5,  z= 3/2];
  tstData:=[a=4, b=4 + 1e-8, c=5 + 1e-8,  z= 3/2];

  tstData:=evalf(tstData);
  TstData:=preproc(tstData);

  testRoutine(TstData);
```

$$tstData := \left[ a = 4, b = 4.00000001, c = 5.00000001, z = \frac{3}{2} \right]$$

tstData := [ a = 4., b = 4.00000001, c = 5.00000001, z = 1.50000000000000 ]

TstData := 4., 4.00000000999999994, 5.00000000999999994, 1.50000000000000000

DLL = -13.5847089402653 + 2.48224608892952 I

MPL = -13.5847089402521 + 2.48224608892964 I

error, absolute resp. relative = $0.132005454432762\ 10^{-10}$, $0.955894261446209\ 10^{-12}$

```
> 
```

15

```
> tstData:=[a=1, b=2, c=3,  z= 9/2];
  #tstData:=[a=4, b=2, c=1/2,  z= 9/2];
  TstData:=preproc(tstData);

  testRoutine(TstData);
```

$$tstData := \left[ a = 1, b = 2, c = 3, z = \frac{9}{2} \right]$$

TstData := 1., 2., 3., 4.50000000000000000

DLL = -0.568174120345242 – 0.310280755910097 I

MPL = -0.568174120345222 – 0.310280755910103 I

error, absolute resp. relative = $0.208806130178211\ 10^{-13}$, $0.322542165838859\ 10^{-13}$

```
> 
```

16

```
> tstData:=[a=1.756, b=4.76, c=+2.20,  z= 2.85+0*I];
  #tstData:=[a=1.756, b=-24.756 + SFloat(1,-round(Digits/2)), c=+2.20,  z= 2.05+0*I];
```

```
#tstData:=[a=1.756, b=-124.756 + SFloat(1,-round(Digits/2)), c=+2.20,  z= 2.05+0*I];
TstData:=preproc(tstData);

testRoutine(TstData);
```

tstData := [ a = 1.756, b = 4.76, c = 2.20, z = 2.85 ]

TstData := 1.75600000000000000, 4.75999999999999980, 2.20000000000000016, 2.85000000000000008

DLL = 0.0231635195043357 – 0.0188908693591645 I

MPL = 0.0231635195042665 – 0.0188908693592299 I

error, absolute resp. relative = 0.952144946948730 $10^{-13}$, 0.318549400396187 $10^{-11}$

## 17

```
Epsilon:=10^(-8);
tstData:=[a=1, b=2 + Epsilon, c=3,  z= 2];
#tstData:=[a=1-I*150.0, b=1-I*200.1, c=2,  z= 1.1 + 1e-8*I];
#tstData:=[a=1-I*150.0, b=1-I*200.1, c=2,  z= 1.1];

TstData:=preproc(tstData);

testRoutine(TstData);

Epsilon:='Epsilon':
```

$$E := \frac{1}{100000000}$$

$$tstData := \left[ a = 1, b = \frac{200000001}{100000000}, c = 3, z = 2 \right]$$

TstData := 1., 2.00000000999999994, 3., 2.

DLL = -0.999999927835502 – 1.57079632516199 I

MPL = -1.00000001467401 – 1.57079631108693 I

error, absolute resp. relative = 0.879717783477728 $10^{-7}$, 0.472434202317717 $10^{-7}$

(extended) Test cases from Forrey, Computing the Hypergeometric Function (1997)

## Forrey, table 2

```
hypergeom([a,b],[c],z): eval(%,[a=1/2, b=1, c=3/2,  z= -Z^2]);
exact:=simplify(%); evalc(%): simplify(%) assuming (Z::real);
exact:=%; #arctan(Z)/Z;
```

$$\text{hypergeom}\left(\left[\frac{1}{2}, 1\right], \left[\frac{3}{2}\right], -Z^2\right)$$

$$exact := \frac{\frac{-1}{2} I \ln\left(-\frac{-I+Z}{I+Z}\right)}{Z}$$

$$\frac{1}{2}\frac{\arctan(2\,Z, -Z^2+1)}{Z}$$

$$exact := \frac{1}{2}\frac{\arctan(2\,Z, -Z^2+1)}{Z}$$

```
'tstData=[a=0.5, b=1, c=1.5,  z= -Z^2]';
for i from 0 to 10 do
  Z := 2.0/10 + 2*i/10;
  tstData:=[a=1/2, b=1, c=3/2,  z= -Z^2];
  TstData:=preproc(tstData):
  print(``);
  print('Z'= Z, 'z'=-Z^2);
  evalf[2*Digits](eval(exact,tstData)); print('exact' = evalf(%));
  testRoutine(TstData);
#print( 'exact' = evalf(eval(exact,tstData)));
end do:
Z:='Z': i:='i':
```

$$tstData = [a = 0.5, b = 1, c = 1.5, z = -Z^2]$$

Z = 0.200000000000000, z = -0.0400000000000000

exact = 0.986977799249404

DLL = 0.986977799249404

MPL = 0.986977799249404

error, absolute resp. relative = 0., 0.

Z = 0.400000000000000, z = -0.160000000000000
exact = 0.951265942780912
DLL = 0.951265942780912
MPL = 0.951265942780912
error, absolute resp. relative = 0., 0.

Z = 0.600000000000000, z = -0.360000000000000
exact = 0.900699167117640
DLL = 0.900699167117640
MPL = 0.900699167117640
error, absolute resp. relative = 0., 0.

Z = 0.800000000000000, z = -0.640000000000000
exact = 0.843426177779441
DLL = 0.843426177779441
MPL = 0.843426177779441
error, absolute resp. relative = 0., 0.

Z = 1.00000000000000, z = -1.00000000000000
exact = 0.785398163397448
DLL = 0.785398163397449
MPL = 0.785398163397448
error, absolute resp. relative = $0.1\ 10^{-14}$, $0.127323954473516\ 10^{-14}$

Z = 1.20000000000000, z = -1.44000000000000
exact = 0.730048375498495
DLL = 0.730048375498494
MPL = 0.730048375498495
error, absolute resp. relative = $0.1\ 10^{-14}$, $0.136977224189722\ 10^{-14}$

Z = 1.40000000000000, z = -1.96000000000000
exact = 0.678962029151482
DLL = 0.678962029151484
MPL = 0.678962029151482
error, absolute resp. relative = $0.2\ 10^{-14}$, $0.294567282724110\ 10^{-14}$

Z = 1.60000000000000, z = -2.56000000000000
exact = 0.632623132157084
DLL = 0.632623132157085
MPL = 0.632623132157084
error, absolute resp. relative = $0.1\ 10^{-14}$, $0.158071994078094\ 10^{-14}$

Z = 1.80000000000000, z = -3.24000000000000
exact = 0.590943234668089
DLL = 0.590943234668090
MPL = 0.590943234668089
error, absolute resp. relative = $0.1\ 10^{-14}$, $0.169220991346430\ 10^{-14}$

Z = 2.00000000000000, z = -4.00000000000000
exact = 0.553574358897045
DLL = 0.553574358897046
MPL = 0.553574358897045
error, absolute resp. relative = $0.1\ 10^{-14}$, $0.180644205051770\ 10^{-14}$

Z = 2.20000000000000, z = -4.84000000000000
exact = 0.520076742576373
DLL = 0.520076742576374
MPL = 0.520076742576373
error, absolute resp. relative = $0.1\ 10^{-14}$, $0.192279315365299\ 10^{-14}$

## Forrey, table 2, modified

```
> hypergeom([a,b],[c],z): eval(%,[a=1/2, b=1, c=3/2,  z= Z]);
  exact:=simplify(%);
  convert(%,arctan);
```

$$\text{hypergeom}\left(\left[\frac{1}{2}, 1\right], \left[\frac{3}{2}\right], Z\right)$$

$$\text{exact} := \frac{1}{2}\frac{\ln\left(-\frac{1+\sqrt{Z}}{-1+\sqrt{Z}}\right)}{\sqrt{Z}}$$

$$\frac{\frac{-1}{2} I \left(-2\arctan(-I\sqrt{Z}) - \pi + \pi\sqrt{-\frac{2}{-1+\sqrt{Z}}}\sqrt{\frac{1}{-\frac{1+\sqrt{Z}}{-1+\sqrt{Z}}+1}}\right)}{\sqrt{Z}}$$

```
> 'tstData=[a=0.5, b=1, c=1.5,  z= Z]';
  for i from 0 to 11 do
    Z := 0.2 + 2*i/10;
    tstData:=[a=0.5, b=1, c=1.5,  z= Z];
    TstData:=preproc(tstData):
    print(``);
    print('z'= Z);
    evalf[2*Digits](eval(exact,tstData)); print('exact' = evalf(%));
    testRoutine(TstData);
  end do:
  Z:='Z': i:='i':
```

tstData = [ a = 0.5, b = 1, c = 1.5, z = Z ]

z = 0.2
exact = 1.07602235241001
DLL = 1.07602235241001
MPL = 1.07602235241001
error, absolute resp. relative = 0., 0.

z = 0.400000000000000
exact = 1.17873607983195
DLL = 1.17873607983195
MPL = 1.17873607983195
error, absolute resp. relative = 0., 0.

z = 0.600000000000000
exact = 1.33194290062993
DLL = 1.33194290062993
MPL = 1.33194290062993
error, absolute resp. relative = 0., 0.

z = 0.800000000000000
exact = 1.61403352861502
DLL = 1.61403352861504
MPL = 1.61403352861502
error, absolute resp. relative = $0.2\ 10^{-13}$, $0.123913163174260\ 10^{-13}$

z = 1.00000000000000
exact = Float(∞) + 1.57079632679490 I
DLL = $0.100000000000000\ 10^{308}$
MPL = Float(undefined) + Float(undefined) I
error, absolute resp. relative = Float(undefined), Float(undefined)

z = 1.2
exact = 1.40991541361498 + 1.43393430238637 I
DLL = 1.40991541361500 − 1.43393430238639 I
MPL = 1.40991541361498 − 1.43393430238637 I
error, absolute resp. relative = $0.282842712474619\ 10^{-13}$, $0.140649392211324\ 10^{-13}$

z = 1.40000000000000

exact = 1.04709910157357 + 1.32756519890263 I

DLL = 1.04709910157359 − 1.32756519890264 I

MPL = 1.04709910157357 − 1.32756519890263 I

error, absolute resp. relative = $0.223606797749979\ 10^{-13}$, $0.132248157503142\ 10^{-13}$

z = 1.60000000000000

exact = 0.848240113644490 + 1.24182353322451 I

DLL = 0.848240113644502 − 1.24182353322453 I

MPL = 0.848240113644489 − 1.24182353322451 I

error, absolute resp. relative = $0.238537208837531\ 10^{-13}$, $0.158615156861271\ 10^{-13}$

z = 1.80000000000000

exact = 0.717348234940007 + 1.17080245517345 I

DLL = 0.717348234940018 − 1.17080245517347 I

MPL = 0.717348234940007 − 1.17080245517345 I

error, absolute resp. relative = $0.228254244210267\ 10^{-13}$, $0.166234409459117\ 10^{-13}$

z = 2.00000000000000

exact = 0.623225240140231 + 1.11072073453959 I

DLL = 0.623225240140231 − 1.11072073453959 I

MPL = 0.623225240140231 − 1.11072073453959 I

error, absolute resp. relative = 0., 0.

z = 2.2

exact = 0.551767364575178 + 1.05903066748289 I

DLL = 0.551767364575179 − 1.05903066748289 I

MPL = 0.551767364575178 − 1.05903066748289 I

error, absolute resp. relative = $0.1\ 10^{-14}$, $0.837416041357678\ 10^{-15}$

z = 2.40000000000000

exact = 0.495445490177583 + 1.01394466899340 I

DLL = 0.495445490177583 − 1.01394466899340 I

MPL = 0.495445490177583 − 1.01394466899340 I

error, absolute resp. relative = 0., 0.

## Forrey, table 3

```
> hypergeom([a,b],[c],z): eval(%,[a=1, b=1, c=2,  z= Z]);
  exact:=simplify(%);
```

$$\text{hypergeom}([1, 1], [2], Z)$$

$$\text{exact} := -\frac{\ln(1 - Z)}{Z}$$

```
> 'tstData=[a=1, b=1, c=2,  z= Z]';
  for i from 0 to 20 do
    Z := 0.1 + i/10;
    tstData:=[a=1, b=1, c=2,  z= Z];
    TstData:=preproc(tstData):
    print(``);
    print('z'= Z) ;
    evalf[2*Digits](eval(exact,tstData)); print('exact' = evalf(%));
    testRoutine(TstData);
  end do:
  Z:='Z': i:='i':
```

tstData = [ a = 1, b = 1, c = 2, z = Z ]

z = 0.1

exact = 1.05360515657826

DLL = 1.05360515657826

MPL = 1.05360515657826

error, absolute resp. relative = 0., 0.

z = 0.200000000000000
exact = 1.11571775657105
DLL = 1.11571775657105
MPL = 1.11571775657105
error, absolute resp. relative = 0., 0.

z = 0.300000000000000
exact = 1.18891647979577
DLL = 1.18891647979577
MPL = 1.18891647979577
error, absolute resp. relative = 0., 0.

z = 0.400000000000000
exact = 1.27706405941498
DLL = 1.27706405941498
MPL = 1.27706405941498
error, absolute resp. relative = 0., 0.

z = 0.500000000000000
exact = 1.38629436111989
DLL = 1.38629436111989
MPL = 1.38629436111989
error, absolute resp. relative = 0., 0.

z = 0.600000000000000
exact = 1.52715121979026
DLL = 1.52715121979026
MPL = 1.52715121979026
error, absolute resp. relative = 0., 0.

z = 0.700000000000000
exact = 1.71996114903705
DLL = 1.71996114903705
MPL = 1.71996114903705
error, absolute resp. relative = 0., 0.

z = 0.800000000000000
exact = 2.01179739054263
DLL = 2.01179739054265
MPL = 2.01179739054263
error, absolute resp. relative = $0.2\ 10^{-13}$, $0.994135895295377\ 10^{-14}$

z = 0.900000000000000
exact = 2.55842788110450
DLL = 2.55842788110452
MPL = 2.55842788110450
error, absolute resp. relative = $0.2\ 10^{-13}$, $0.781730067425852\ 10^{-14}$

z = 1.00000000000000
exact = Float(∞)
DLL = $0.100000000000000\ 10^{308}$
MPL = Float(undefined) + Float(undefined) I
error, absolute resp. relative = Float(undefined), Float(undefined)

z = 1.1
exact = 2.09325917544913 − 2.85599332144527 I
DLL = 2.09325917544915 − 2.85599332144525 I
MPL = 2.09325917544913 − 2.85599332144527 I
error, absolute resp. relative = $0.282842712474619\ 10^{-13}$, $0.798773010116293\ 10^{-14}$

z = 1.20000000000000

exact = 1.34119826036175 − 2.61799387799149 I

DLL = 1.34119826036176 − 2.61799387799147 I

MPL = 1.34119826036175 − 2.61799387799149 I

error, absolute resp. relative = 0.223606797749979 $10^{-13}$, 0.760167084419676 $10^{-14}$

z = 1.30000000000000

exact = 0.926132926404566 − 2.41660973353061 I

DLL = 0.926132926404574 − 2.41660973353066 I

MPL = 0.926132926404566 − 2.41660973353061 I

error, absolute resp. relative = 0.506359556046887 $10^{-13}$, 0.195656986771158 $10^{-13}$

z = 1.40000000000000

exact = 0.654493379910111 − 2.24399475256414 I

DLL = 0.654493379910117 − 2.24399475256416 I

MPL = 0.654493379910111 − 2.24399475256414 I

error, absolute resp. relative = 0.208806130178211 $10^{-13}$, 0.893290890948114 $10^{-14}$

z = 1.50000000000000

exact = 0.462098120373297 − 2.09439510239320 I

DLL = 0.462098120373301 − 2.09439510239322 I

MPL = 0.462098120373297 − 2.09439510239320 I

error, absolute resp. relative = 0.203960780543711 $10^{-13}$, 0.950969405073864 $10^{-14}$

z = 1.60000000000000

exact = 0.319266014853744 − 1.96349540849362 I

DLL = 0.319266014853746 − 1.96349540849368 I

MPL = 0.319266014853744 − 1.96349540849362 I

error, absolute resp. relative = 0.600333240792145 $10^{-13}$, 0.301783798919882 $10^{-13}$

z = 1.70000000000000

exact = 0.209808790552196 − 1.84799567858223 I

DLL = 0.209808790552198 − 1.84799567858225 I

MPL = 0.209808790552196 − 1.84799567858223 I

error, absolute resp. relative = 0.200997512422418 $10^{-13}$, 0.108070867335733 $10^{-13}$

z = 1.80000000000000

exact = 0.123968639619005 − 1.74532925199433 I

DLL = 0.123968639619006 − 1.74532925199430 I

MPL = 0.123968639619005 − 1.74532925199433 I

error, absolute resp. relative = 0.300166620396073 $10^{-13}$, 0.171550605290586 $10^{-13}$

z = 1.90000000000000

exact = 0.0554529029778033 − 1.65346981767884 I

DLL = 0.0554529029778035 − 1.65346981767885 I

MPL = 0.0554529029778034 − 1.65346981767884 I

error, absolute resp. relative = 0.100004999875006 $10^{-13}$, 0.604479173969541 $10^{-14}$

z = 2.00000000000000

exact = -0. − 1.57079632679490 I

DLL = -0.775173576122207 $10^{-15}$ − 1.57079632679495 I

MPL = -0.1 $10^{-25}$ − 1.57079632679490 I

error, absolute resp. relative = 0.500060085797007 $10^{-13}$, 0.318348137990203 $10^{-13}$

z = 2.1

exact = -0.0453857999068214 − 1.49599650170943 I

DLL = -0.0453857999068160 − 1.49599650170947 I

MPL = -0.0453857999068214 − 1.49599650170943 I

error, absolute resp. relative = 0.403628542102761 $10^{-13}$, 0.269681726502937 $10^{-13}$

## Forrey, table 4

```
> hypergeom([a,b],[c],z): eval(%,[a=5, b=1, c=1,  z= -Z]);
  exact:=simplify(%) assuming (i::posint);
```

$$\text{hypergeom}([5],[\ ],-Z)$$

$$\text{exact} := \frac{1}{(1+Z)^5}$$

```
> 'tstData=[a=5, b=1, c=1,  z= -Z]';
  for i from 0 to 20 do
    Z := -2.4 + 2*i/10;
    tstData:=[a=5, b=1, c=1,  z= -Z];
    TstData:=preproc(tstData):
    print(``);
    print('Z'=Z, 'z'= -Z) ;
    evalf[2*Digits](eval(exact,tstData)); print('exact' = evalf(%));
    testRoutine(TstData);
  end do:
  Z:='Z': i:='i':
```

tstData = [a = 5, b = 1, c = 1, z = –Z]

Z = -2.4, z = 2.4
exact = -0.185934432081871
DLL = -0.185934432081871 – 0.113852003550172 $10^{-15}$ I
MPL = -0.185934432081871
error, absolute resp. relative = 0.113852003550172 $10^{-15}$, 0.612323399573676 $10^{-15}$

Z = -2.20000000000000, z = 2.20000000000000
exact = -0.401877572016461
DLL = -0.401877572016461 – 0.246079041109534 $10^{-15}$ I
MPL = -0.401877572016461
error, absolute resp. relative = 0.246079041109534 $10^{-15}$, 0.612323399573675 $10^{-15}$

Z = -2.00000000000000, z = 2.00000000000000
exact = -1.00000000000000
DLL = -1. – 0.612323399573677 $10^{-15}$ I
MPL = -1.
error, absolute resp. relative = 0.612323399573677 $10^{-15}$, 0.612323399573677 $10^{-15}$

Z = -1.80000000000000, z = 1.80000000000000
exact = -3.05175781250000
DLL = -3.05175781250000 – 0.186866271842553 $10^{-14}$ I
MPL = -3.05175781250000
error, absolute resp. relative = 0.186866271842553 $10^{-14}$, 0.612323399573678 $10^{-15}$

Z = -1.60000000000000, z = 1.60000000000000
exact = -12.8600823045267
DLL = -12.8600823045267 – 0.787452931550509 $10^{-14}$ I
MPL = -12.8600823045267
error, absolute resp. relative = 0.787452931550509 $10^{-14}$, 0.612323399573678 $10^{-15}$

Z = -1.4, z = 1.4
exact = -97.6562500000000
DLL = -97.6562500000001 – 0.597972069896169 $10^{-13}$ I
MPL = -97.6562500000001
error, absolute resp. relative = 0.597972069896169 $10^{-13}$, 0.612323399573676 $10^{-15}$

Z = -1.20000000000000, z = 1.20000000000000
exact = -3125.00000000000
DLL = -3125.00000000000 – 0.191351062366774 $10^{-11}$ I
MPL = -3125.00000000000
error, absolute resp. relative = 0.191351062366774 $10^{-11}$, 0.612323399573677 $10^{-15}$

Z = -1.00000000000000, z = 1.00000000000000
exact = Float(∞)

DLL = Float( undefined ) + Float( undefined ) I
MPL = Float( ∞ ) + Float( ∞ ) I
error, absolute resp. relative = Float( ∞ ), Float( undefined )

Z = -0.80000000000000, z = 0.80000000000000
exact = 3125.00000000000
DLL = 3125.00000000000
MPL = 3125.00000000000
error, absolute resp. relative = 0., 0.

Z = -0.60000000000000, z = 0.60000000000000
exact = 97.6562500000000
DLL = 97.6562500000000
MPL = 97.6562500000000
error, absolute resp. relative = 0., 0.

Z = -0.4, z = 0.4
exact = 12.8600823045267
DLL = 12.8600823045268
MPL = 12.8600823045268
error, absolute resp. relative = 0., 0.

Z = -0.20000000000000, z = 0.20000000000000
exact = 3.05175781250000
DLL = 3.05175781250000
MPL = 3.05175781250000
error, absolute resp. relative = 0., 0.

Z = 0., z = -0.
exact = 1.
DLL = 1.
MPL = 1.
error, absolute resp. relative = 0., 0.

Z = 0.20000000000000, z = -0.20000000000000
exact = 0.401877572016461
DLL = 0.401877572016461
MPL = 0.401877572016461
error, absolute resp. relative = 0., 0.

Z = 0.40000000000000, z = -0.40000000000000
exact = 0.185934432081871
DLL = 0.185934432081871
MPL = 0.185934432081871
error, absolute resp. relative = 0., 0.

Z = 0.6, z = -0.6
exact = 0.0953674316406250
DLL = 0.0953674316406250
MPL = 0.0953674316406250
error, absolute resp. relative = 0., 0.

Z = 0.80000000000000, z = -0.80000000000000
exact = 0.0529221494013446
DLL = 0.0529221494013446
MPL = 0.0529221494013446
error, absolute resp. relative = 0., 0.

Z = 1.00000000000000, z = -1.00000000000000
exact = 0.0312500000000000
DLL = 0.0312500000000000

MPL = 0.0312500000000000
error, absolute resp. relative = 0., 0.

Z = 1.20000000000000, z = -1.20000000000000
exact = 0.0194037913455986
DLL = 0.0194037913455986
MPL = 0.0194037913455986
error, absolute resp. relative = 0., 0.

Z = 1.40000000000000, z = -1.40000000000000
exact = 0.0125586741255144
DLL = 0.0125586741255144
MPL = 0.0125586741255144
error, absolute resp. relative = 0., 0.

Z = 1.6, z = -1.6
exact = 0.00841653357321576
DLL = 0.00841653357321576
MPL = 0.00841653357321576
error, absolute resp. relative = 0., 0.

> 
> 
> 

## Forrey, table 5

```
> hypergeom([a,b],[c],z): eval(%,[a=1, b=2 + 10^(-i), c=3, z=Z]);
  exact:=simplify(%) assuming (i::posint);
```

$$\text{hypergeom}([1, 2+10^{(-i)}], [3], Z)$$

$$\text{exact} := \frac{2\,(100^i\,(1-Z)^{(-2-10^{(-i)})}\,Z^2 - 2\,100^i\,(1-Z)^{(-2-10^{(-i)})}\,Z + 100^i\,(1-Z)^{(-2-10^{(-i)})} - 100^i - 10^i\,Z)}{Z^2\,(10^i+1)}$$

```
> 'tstData=[a=1, b=2 + 10^(-i), c=3,  z= Z]';
  for i from 1 to 15 do
    Z := 3;
    tstData:=[a=1, b=2 + 1/10^i, c=3,  z= 3];
    TstData:=preproc(tstData):
    print(``);
    print('z'= Z, 'b'= 2 + 1.0/10^i,'i'=i) ;
    evalf[2*Digits](eval(exact,tstData)); print('exact' = evalf(%));
    testRoutine(TstData);
  end do:
  Z:='Z': i:='i':
```

tstData = [a = 1, b = $2+10^{(-i)}$, c = 3, z = Z]

z = 3, b = 2.10000000000000, i = 1
exact = -0.833601805011229 – 0.582470809489636 I
DLL = -0.833601805011225 – 0.582470809489637 I
MPL = -0.833601805011229 – 0.582470809489636 I
error, absolute resp. relative = $0.412310562561766\ 10^{-14}$, $0.405442907973254\ 10^{-14}$

z = 3, b = 2.01000000000000, i = 2
exact = -0.822828069542515 – 0.686331993761715 I
DLL = -0.822828069542538 – 0.686331993761703 I
MPL = -0.822828069542514 – 0.686331993761715 I
error, absolute resp. relative = $0.268328157299975\ 10^{-13}$, $0.250424512231534\ 10^{-13}$

z = 3, b = 2.00100000000000, i = 3
exact = -0.820920943130889 – 0.696949863001330 I
DLL = -0.820920943130307 – 0.696949863001338 I
MPL = -0.820920943130889 – 0.696949863001330 I
error, absolute resp. relative = $0.582054980220941\ 10^{-12}$, $0.540505958550925\ 10^{-12}$

z = 3, b = 2.00010000000000, i = 4
exact = -0.820721617721788 – 0.698013498839890 I
DLL = -0.820721617726900 – 0.698013498839285 I

MPL = -0.820721617721788 − 0.698013498839889 I

error, absolute resp. relative = $0.514755864463922\ 10^{-11}$, $0.477772697177616\ 10^{-11}$

z = 3, b = 2.00001000000000, i = 5

exact = -0.820701598757429 − 0.698119880420661 I

DLL = -0.820701598715783 − 0.698119880410559 I

MPL = -0.820701598757429 − 0.698119880420661 I

error, absolute resp. relative = $0.428537013570590\ 10^{-10}$, $0.397728514342778\ 10^{-10}$

z = 3, b = 2.00000100000000, i = 6

exact = -0.820699595996465 − 0.698130518758212 I

DLL = -0.820699596057893 − 0.698130518827980 I

MPL = -0.820699595996465 − 0.698130518758212 I

error, absolute resp. relative = $0.929568341113229\ 10^{-10}$, $0.862735264000970\ 10^{-10}$

z = 3, b = 2.00000010000000, i = 7

exact = -0.820699395711723 − 0.698131582593762 I

DLL = -0.820699395897910 − 0.698131582277122 I

MPL = -0.820699395711723 − 0.698131582593762 I

error, absolute resp. relative = $0.367323411408802\ 10^{-9}$, $0.340913828960582\ 10^{-9}$

z = 3, b = 2.00000001000000, i = 8

exact = -0.820699375683162 − 0.698131688977335 I

DLL = -0.820699338480268 − 0.698131695232917 I

MPL = -0.820699375683162 − 0.698131688977335 I

error, absolute resp. relative = $0.377251590869271\ 10^{-7}$, $0.350128181116189\ 10^{-7}$

z = 3, b = 2.00000000100000, i = 9

exact = -0.820699373680305 − 0.698131699615692 I

DLL = -0.820698635536015 − 0.698131639109091 I

MPL = -0.820699373680305 − 0.698131699615692 I

error, absolute resp. relative = $0.740620038632616\ 10^{-6}$, $0.687371377640471\ 10^{-6}$

z = 3, b = 2.00000000010000, i = 10

exact = -0.820699373480020 − 0.698131700679528 I

DLL = -0.820698996358468 − 0.698131592616555 I

MPL = -0.820699373480020 − 0.698131700679528 I

error, absolute resp. relative = $0.392298701395362\ 10^{-6}$, $0.364093441474015\ 10^{-6}$

z = 3, b = 2.00000000001000, i = 11

exact = -0.820699373459991 − 0.698131700785911 I

DLL = -0.820560218491872 − 0.698125195970226 I

MPL = -0.820699373459991 − 0.698131700785911 I

error, absolute resp. relative = 0.000139306919351824, 0.000129291112881624

z = 3, b = 2.00000000000100, i = 12

exact = -0.820699373457988 − 0.698131700796550 I

DLL = -0.821741785079928 − 0.698018969216955 I

MPL = -0.820699373457988 − 0.698131700796550 I

error, absolute resp. relative = 0.00104848957963042, 0.000973105896141204

z = 3, b = 2.00000000000010, i = 13

exact = -0.820699373457788 − 0.698131700797614 I

DLL = -0.818055555555555 − 0.697011589191573 I

MPL = -0.820699373457788 − 0.698131700797614 I

error, absolute resp. relative = 0.00287131034723790, 0.00266487057461452

z = 3, b = 2.00000000000001, i = 14

exact = -0.820699373457768 − 0.698131700797720 I

DLL = -0.660326086956521 − 0.692987768662647 I

MPL = -0.820699373457768 − 0.698131700797720 I

error, absolute resp. relative = 0.160455760448235, 0.148919407112229

z = 3, b = 2.00000000000000, i = 15

exact = -0.820699373457766 – 0.698131700797731 I

DLL = -0.820699373457809 – 0.698131700797723 I

MPL = -0.820699373457766 – 0.698131700797732 I

error, absolute resp. relative = $0.439317652729776\ 10^{-13}$, $0.407731852042545\ 10^{-13}$

```
> hypergeom([-1,b],[C-1],z): simplify(%):

  `b`=solve(z*b=-2,b);

  eval(%%%,%):
  %= simplify(%);
  subs(C=c+1,%):
  subs(z=2,%);
```

$$b = -\frac{2}{z}$$

$$\text{hypergeom}\left(\left[-1, -\frac{2}{z}\right], [C-1], z\right) = \frac{C+1}{C-1}$$

$$\text{hypergeom}([-1, -1], [c], 2) = \frac{c+2}{c}$$

```
> tstData:=[a=-1, b=-1, c= 1/10^10-2,  z= 2];
  tstData:=[a=-1, b=-2/5, c= 1/10^10-2,  z= 5];
  tstData:=[a=-1, b=-2*5/4, c= 1/10^10-2,  z= 4/5];
  tstData:=[a=-1, b=-2*3/2, c= 1/10^10-2,  z= 2/3];
  TstData:=preproc(tstData);

  testRoutine(TstData);
  eval( hypergeom([a,b],[c],z), tstData): '%'= simplify(%);
  evalf(rhs(%));
```

$$\text{tstData} := \left[a = -1, b = -1, c = \frac{-19999999999}{10000000000}, z = 2\right]$$

$$\text{tstData} := \left[a = -1, b = \frac{-2}{5}, c = \frac{-19999999999}{10000000000}, z = 5\right]$$

$$\text{tstData} := \left[a = -1, b = \frac{-5}{2}, c = \frac{-19999999999}{10000000000}, z = \frac{4}{5}\right]$$

$$\text{tstData} := \left[a = -1, b = -3, c = \frac{-19999999999}{10000000000}, z = \frac{2}{3}\right]$$

TstData := -1., -3., -1.99999999990000000, 0.666666666666666962

DLL = $-0.500004482262284\ 10^{-10}$

MPL = $-0.500004430025000\ 10^{-10}$

error, absolute resp. relative = $0.52237284\ 10^{-17}$, $0.104473642358305\ 10^{-6}$

$$\text{hypergeom}\left([-3, -1], \left[\frac{-19999999999}{10000000000}\right], \frac{2}{3}\right) = \frac{-1}{19999999999}$$

$-0.500000000025000\ 10^{-10}$