The four double-hypergeometric Appell functions,

a complete implementation in a computer algebra system

Edgardo S. Cheb-Terrab
Physics, Differential Equations and Mathematical Functions, Maplesoft

Abstract:
The four multi-parameter Appell functions, AppellF1, AppellF2, AppellF3 and AppellF4 are doubly hypergeometric functions that include as particular cases the 2F1 hypergeometric and some cases of the MeijerG function, and with them most of the known functions of mathematical physics. Appell functions have been popping up with increasing frequency in applications in quantum mechanics, molecular physics, and general relativity. In this talk, a full implementation of these functions in the Maple computer algebra system, including, for the first time, their numerical evaluation over the whole complex plane, is presented, with details about the symbolic and numerical strategies used.

Appell Functions (symbolic)

The main references:

• P. Appel, J.Kamke de Feriet, "*Fonctions hypergeometriques et Hyperspheriques*", 1926

• H. Srivastava, P.W. Karlsson, "*Multiple Gaussian Hypergeometric Series*", 1985

• 24 papers in the literature, ranging from 1882 to 2015

Definition and Symmetries

The definition of the four Appell series and the corresponding domains of convergence can be seen through the FunctionAdvisor. For example, for AppellF1,

> *restart*;

> *FunctionAdvisor*(*definition*, *AppellF1*)

$$\left[F_1(a, b_1, b_2, c, z_1, z_2) = \sum_{\_k1=0}^{\infty} \sum_{\_k2=0}^{\infty} \frac{(a)_{\_k1+\_k2} (b_1)_{\_k1} (b_2)_{\_k2} z_1^{\_k1} z_2^{\_k2}}{(c)_{\_k1+\_k2}\, \_k1!\, \_k2!}, |z_1| < 1 \wedge |z_2| < 1\right] \quad \textbf{(1.1.1)}$$

From this definition, by swapping the AppellF1 variables subscripted with the numbers 1 and 2, the function remains the same; hence

> *FunctionAdvisor*(*symmetries*, *AppellF1*)

$$\left[F_1(a, b_2, b_1, c, z_2, z_1) = F_1(a, b_1, b_2, c, z_1, z_2)\right] \quad \textbf{(1.1.2)}$$

Analogously, the definition and symmetry of AppellF2 are

> $FunctionAdvisor(definition, AppellF2)$

$$\left[F_2(a, b_1, b_2, c_1, c_2, z_1, z_2) = \sum_{\_k1=0}^{\infty} \sum_{\_k2=0}^{\infty} \frac{(a)_{\_k1+\_k2} (b_1)_{\_k1} (b_2)_{\_k2} z_1^{\_k1} z_2^{\_k2}}{(c_1)_{\_k1} (c_2)_{\_k2} \_k1! \, \_k2!}, |z_1| + |z_2| < 1\right] \quad \textbf{(1.1.3)}$$

> $FunctionAdvisor(symmetries, AppellF2)$

$$[F_2(a, b_2, b_1, c_2, c_1, z_2, z_1) = F_2(a, b_1, b_2, c_1, c_2, z_1, z_2)] \quad \textbf{(1.1.4)}$$

The cases of AppellF3 and AppellF4 are more general in that, from their definition, besides the symmetry under a swap of the subscripted variables, these two functions have additional symmetries under exchange of positions of the function's parameters

> $FunctionAdvisor(definition, AppellF3)$

$$\left[F_3(a_1, a_2, b_1, b_2, c, z_1, z_2) = \sum_{\_k1=0}^{\infty} \sum_{\_k2=0}^{\infty} \frac{(a_1)_{\_k1} (a_2)_{\_k2} (b_1)_{\_k1} (b_2)_{\_k2} z_1^{\_k1} z_2^{\_k2}}{(c)_{\_k1+\_k2} \_k1! \, \_k2!}, |z_1| < 1 \wedge |z_2| < 1\right] \quad \textbf{(1.1.5)}$$

> $FunctionAdvisor(symmetries, AppellF3)$

$$[F_3(a_2, a_1, b_2, b_1, c, z_2, z_1) = F_3(a_1, a_2, b_1, b_2, c, z_1, z_2), F_3(b_1, a_2, a_1, b_2, c, z_1, z_2) = F_3(a_1, a_2, b_1, b_2, c, z_1, z_2), F_3(a_1, b_2, b_1, a_2, c, z_1, z_2) = F_3(a_1, a_2, b_1, b_2, c, z_1, z_2)] \quad \textbf{(1.1.6)}$$

> $FunctionAdvisor(definition, AppellF4)$

$$\left[F_4(a, b, c_1, c_2, z_1, z_2) = \sum_{\_k1=0}^{\infty} \sum_{\_k2=0}^{\infty} \frac{(a)_{\_k1+\_k2} (b)_{\_k1+\_k2} z_1^{\_k1} z_2^{\_k2}}{(c_1)_{\_k1} (c_2)_{\_k2} \_k1! \, \_k2!}, \sqrt{|z_1|} + \sqrt{|z_2|} < 1\right] \quad \textbf{(1.1.7)}$$

> $FunctionAdvisor(symmetries, AppellF4)$

$$[F_4(a, b, c_2, c_1, z_2, z_1) = F_4(a, b, c_1, c_2, z_1, z_2), F_4(b, a, c_1, c_2, z_1, z_2) = F_4(a, b, c_1, c_2, z_1, z_2)] \quad \textbf{(1.1.8)}$$

Polynomial and Singular Cases

- From these four definitions, the Appell functions are singular (division by zero) when the *c* parameters entering the pochhammer functions in the denominators of these series are non-positive integers: these pochhammer functions will be equal to zero when the summation indices of these series are bigger than the absolute values of the *c* parameters.

- For an analogous reason, when the *a* and/or *b* parameters entering the pochhammer functions in the numerators of the series are non-positive integers, the series will truncate and the Appell functions will be polynomial.
- When both the pochhammers in the numerators and denominators have non-positive integer arguments, the Appell functions are polynomial when the absolute values of the non-positive integers in the numerators are smaller than or equal to the absolute values of the non-positive integers in the denominators, and singular otherwise.
- The combinatorial of all these conditions can also be consulted using the FunctionAdvisor. For example, for AppellF1, the singular cases happen when any of the following conditions hold

```
> FunctionAdvisor(singularities, AppellF1)
```

$$\left[F_1(a, b_1, b_2, c, z_1, z_2), \left(c::\mathbb{Z}^{(0,-)} \wedge a::\mathbb{Z}^{(0,-)} \wedge b_1::(\neg\, \mathbb{Z}^{(0,-)}) \wedge a < c\right) \vee \left(c::\mathbb{Z}^{(0,-)} \wedge a::\mathbb{Z}^{(0,-)} \wedge b_2::(\neg\, \mathbb{Z}^{(0,-)}) \wedge a < c\right) \vee \left(c::\mathbb{Z}^{(0,-)} \wedge a::\mathbb{Z}^{(0,-)} \wedge b_1::\mathbb{Z}^{(0,-)} \wedge b_2::\mathbb{Z}^{(0,-)} \wedge a < c \wedge b_1 + b_2 < c\right) \vee \left(c::\mathbb{Z}^{(0,-)} \wedge a::(\neg\, \mathbb{Z}^{(0,-)}) \wedge b_1::(\neg\, \mathbb{Z}^{(0,-)})\right) \vee \left(c::\mathbb{Z}^{(0,-)} \wedge a::(\neg\, \mathbb{Z}^{(0,-)}) \wedge b_2::(\neg\, \mathbb{Z}^{(0,-)})\right) \vee \left(c::\mathbb{Z}^{(0,-)} \wedge a::(\neg\, \mathbb{Z}^{(0,-)}) \wedge b_1::\mathbb{Z}^{(0,-)} \wedge b_2::\mathbb{Z}^{(0,-)} \wedge b_1 + b_2 < c\right)\right] \quad (1.2.1)$$

```
> %AppellF1(1/2, 3/2, 1/4, -2, z1, z2)
```

$$F_1\left(\frac{1}{2}, \frac{3}{2}, \frac{1}{4}, -2, z1, z2\right) \quad (1.2.2)$$

```
> value(%)
Error, (in AppellF1:-Singularities) numeric exception: division by zero
```

A polynomial example: b_1 and b_2 are negative integers

```
> AppellF1(3/2, -2, -1, 1/2, z1, z2)
```

$$F_1\left(\frac{3}{2}, -2, -1, \frac{1}{2}, z1, z2\right) \quad (1.2.3)$$

```
> simplify(series((1.2.3), z1))
```

$$1 - 3\, z2 + (-6 + 10\, z2)\, z1 + (5 - 7\, z2)\, z1^2 + \mathrm{O}\left(z1^6\right) \quad (1.2.4)$$

```
>
```

Single Power Series with Hypergeometric Coefficients

```
> FunctionAdvisor(definition, AppellF1(a, b_1, b_2, c, z_1, z_2))
```

$$\left[F_1(a, b_1, b_2, c, z_1, z_2) = \sum_{\_k1=0}^{\infty} \sum_{\_k2=0}^{\infty} \frac{(a)_{\_k1+\_k2} (b_1)_{\_k1} (b_2)_{\_k2} z_1^{\_k1} z_2^{\_k2}}{(c)_{\_k1+\_k2}\, \_k1!\, \_k2!}, |z_1| \quad (1.3.1)$$

$$< 1 \wedge |z_2| < 1\Big]$$

> *subsindets*((**1.3.1**), *specfunc*(pochhammer), *expand*)

$$\left[F_1(a, b_1, b_2, c, z_1, z_2) = \sum_{\_k1=0}^{\infty} \sum_{\_k2=0}^{\infty} \frac{(a+\_k1)_{\_k2}\,(a)_{\_k1}\,(b_1)_{\_k1}\,(b_2)_{\_k2}\,z_1^{\_k1}\,z_2^{\_k2}}{(c+\_k1)_{\_k2}\,(c)_{\_k1}\,\_k1!\,\_k2!}, |z_1| < 1 \wedge |z_2| < 1\right] \quad \textbf{(1.3.2)}$$

> *expand*((**1.3.2**))

$$\left[F_1(a, b_1, b_2, c, z_1, z_2) = \sum_{\_k1=0}^{\infty} \frac{(a)_{\_k1}\,(b_1)_{\_k1}\,z_1^{\_k1}\left(\displaystyle\sum_{\_k2=0}^{\infty} \frac{(a+\_k1)_{\_k2}\,(b_2)_{\_k2}\,z_2^{\_k2}}{(c+\_k1)_{\_k2}\,\_k2!}\right)}{(c)_{\_k1}\,\_k1!}, |z_1| < 1 \wedge |z_2| < 1\right] \quad \textbf{(1.3.3)}$$

> *value*(%) assuming abs$(z_2) < 1$

$$\left[F_1(a, b_1, b_2, c, z_1, z_2) = \sum_{\_k1=0}^{\infty} \frac{(a)_{\_k1}\,(b_1)_{\_k1}\,z_1^{\_k1}\,{}_2F_1(b_2, a+\_k1; c+\_k1; z_2)}{(c)_{\_k1}\,\_k1!}, |z_1| < 1 \wedge |z_2| < 1\right] \quad \textbf{(1.3.4)}$$

> *FunctionAdvisor*(*sum_form*, *AppellF1*$(a, b_1, b_2, c, z_1, z_2)$)

$$\left[F_1(a, b_1, b_2, c, z_1, z_2) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{(a)_{m+n}\,(b_1)_m\,(b_2)_n\,z_1^m\,z_2^n}{(c)_{m+n}\,m!\,n!}, |z_1| < 1 \wedge |z_2| < 1\right], \left[F_1(a, b_1, b_2, c, z_1, z_2) = \sum_{k=0}^{\infty} \frac{(a)_k\,(b_1)_k\,{}_2F_1(a+k, b_2; c+k; z_2)\,z_1^k}{(c)_k\,k!}, |z_1| < 1\right], \left[F_1(a, b_1, b_2, c, z_1, z_2) = \sum_{k=0}^{\infty} \frac{(a)_k\,(b_2)_k\,{}_2F_1(a+k, b_1; c+k; z_1)\,z_2^k}{(c)_k\,k!}, |z_2| \right. \quad \textbf{(1.3.5)}$$

< 1

> FunctionAdvisor(sum_form, AppellF3)

$$\left[F_3(a_1, a_2, b_1, b_2, c, z_1, z_2) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{(a_1)_m (a_2)_n (b_1)_m (b_2)_n z_1^m z_2^n}{(c)_{m+n} m! n!}, |z_1| < 1 \wedge |z_2| < 1\right], \left[F_3(a_1, a_2, b_1, b_2, c, z_1, z_2) = \sum_{k=0}^{\infty} \frac{(a_1)_k (b_1)_k {}_2F_1(a_2, b_2; c+k; z_2) z_1^k}{(c)_k k!}, |z_1| < 1\right], \left[F_3(a_1, a_2, b_1, b_2, c, z_1, z_2) = \sum_{k=0}^{\infty} \frac{(a_2)_k (b_2)_k {}_2F_1(a_1, b_1; c+k; z_1) z_2^k}{(c)_k k!}, |z_2| < 1\right] \quad \textbf{(1.3.6)}$$

As indicated in the formulas above, for these two Appell functions the domain of convergence of the single sum with hypergeometric coefficients is larger than the domain of convergence of the double series, because the hypergeometric coefficient in the single sum - say the one in z_2 - analytically extends the series with regards to the other variable - say z_1 - entering the hypergeometric coefficient. Hence, for AppellF1 and AppellF3, the case where one of the two variables, z_1 or z_2, is equal to 1, is convergent only when the corresponding hypergeometric coefficient in the single sum form is convergent. For instance, for AppellF1 the convergent case at $z_1 = 1$ requires that $0 < -\mathrm{Re}(-c + a + b_1)$.

The situation is different for AppellF2 and AppellF4, where the domain of convergence with regards to the two variables z_1 and z_2 is ***entangled***, i.e. it intrinsically depends on a combination of the two variables:

> FunctionAdvisor(sum_form, AppellF2)

$$\left[F_2(a, b_1, b_2, c_1, c_2, z_1, z_2) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{(a)_{m+n} (b_1)_m (b_2)_n z_1^m z_2^n}{(c_1)_m (c_2)_n m! n!}, |z_2| + |z_1| < 1\right], \left[F_2(a, b_1, b_2, c_1, c_2, z_1, z_2) = \sum_{k=0}^{\infty} \frac{(a)_k (b_1)_k {}_2F_1(a+k, b_2; c_2; z_2) z_1^k}{(c_1)_k k!}, |z_2| + |z_1| < 1\right], \left[F_2(a, b_1, b_2, c_1, c_2, z_1, z_2)\right. \quad \textbf{(1.3.7)}$$

$$= \sum_{k=0}^{\infty} \frac{(a)_k (b_2)_k \, {}_2F_1(a+k, b_1; c_1; z_1) \, z_2^k}{(c_2)_k \, k!}, |z_2| + |z_1| < 1 \Bigg]$$

> $FunctionAdvisor(sum\_form, AppellF4)$

$$\left[F_4(a, b, c_1, c_2, z_1, z_2) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{(a)_{m+n} (b)_{m+n} z_1^m z_2^n}{(c_1)_m (c_2)_n \, m! \, n!}, \sqrt{|z_2|} + \sqrt{|z_1|} < 1\right], \tag{1.3.8}$$

$$\left[F_4(a, b, c_1, c_2, z_1, z_2) = \sum_{k=0}^{\infty} \frac{(a)_k (b)_k \, {}_2F_1(a+k, b+k; c_2; z_2) \, z_1^k}{(c_1)_k \, k!}, \sqrt{|z_2|} + \sqrt{|z_1|} < 1\right], \left[F_4(a, b, c_1, c_2, z_1, z_2)\right.$$

$$\left.= \sum_{k=0}^{\infty} \frac{(a)_k (b)_k \, {}_2F_1(a+k, b+k; c_1; z_1) \, z_2^k}{(c_2)_k \, k!}, \sqrt{|z_2|} + \sqrt{|z_1|} < 1\right]$$

so the hypergeometric coefficient in one variable in the single sum form does not extend the domain of convergence of the double sum but for particular cases, and from the formulas above one cannot conclude about the value of the function when one of z_1 or z_2 is equal to 1 unless the other one is exactly equal to 0.

Analytic Extension from the Appell Series to the Appell Functions

In the literature, the Appell series are analytically extended by integral representations in terms of Eulerian double integrals. With the exception of AppellF4, one of the two iterated integrals can always be computed resulting in a single integral with hypergeometric integrand. For example, for AppellF3

> $FunctionAdvisor(integral\_form, AppellF3)$

$$\Bigg[F_3(a_1, a_2, b_1, b_2, c, z_1, z_2) \tag{1.4.1}$$

$$= \frac{\Gamma(c) \left(\int_0^1 \frac{(1-u)^{-1+b_1} \, {}_2F_1(a_2, b_2; c-b_1; z_2 u)}{u^{-c+b_1+1} (1+(u-1) z_1)^{a_1}} \, \mathrm{d}u \right)}{\Gamma(b_1) \, \Gamma(c-b_1)}, 0 < \Re(b_1) \wedge 0$$

$$< \Re(c) \wedge 0 < -\Re\left(-c+b_1\right)\Big], \Big[F_3\left(a_1, a_2, b_1, b_2, c, z_1, z_2\right)$$

$$= \frac{\Gamma(c)\left(\int_0^1 \frac{(1-u)^{b_2-1}\ {}_2F_1\left(a_1, b_1; c-b_2; u\, z_1\right)}{u^{-c+b_2+1}\left(1+(u-1)\, z_2\right)^{a_2}}\, \mathrm{d}u\right)}{\Gamma\left(b_2\right)\Gamma\left(c-b_2\right)}, 0 < \Re\left(b_2\right) \wedge 0$$

$$< \Re(c) \wedge 0 < -\Re\left(-c+b_2\right)\Big], \Big[F_3\left(a_1, a_2, b_1, b_2, c, z_1, z_2\right)$$

$$= \frac{1}{\Gamma(\rho)\,\Gamma(c-\rho)}\left(\Gamma(c)\left(\int_0^1 u^{\rho-1}\,(1-u)^{c-\rho-1}\ {}_2F_1\left(a_1, b_1; \rho; u\, z_1\right)\right.\right.$$

$$\left.\left.{}_2F_1\left(a_2, b_2; c-\rho; -(u-1)\, z_2\right)\, \mathrm{d}u\right)\right), 0 < \Re(c)\Big], \Big[F_3\left(a_1, a_2, b_1, b_2, c, z_1, z_2\right)$$

$$= \frac{1}{\Gamma\left(b_1\right)\Gamma\left(b_2\right)\Gamma\left(c-b_1-b_2\right)}\left(\Gamma(c)\left(\int_0^1\right.\right.$$

$$\left.\left.\int_0^{1-v} \frac{u^{-1+b_1}\, v^{b_2-1}}{(1-u-v)^{-c+b_1+b_2+1}\left(-u\, z_1+1\right)^{a_1}\left(-z_2\, v+1\right)^{a_2}}\, \mathrm{d}u\, \mathrm{d}v\right)\right), 0$$

$$< \Re\left(b_1\right) \wedge 0 < \Re\left(b_2\right) \wedge 0 < -\Re\left(-c+b_1+b_2\right)\Big]$$

In the case of AppellF4, single integral representation exists only for particular values of the function's parameters, for example two cases are

> FunctionAdvisor(integral_form, AppellF4)[1 ..2]

$$\left[F_4(a,b,c_1,c_2,z_1,z_2) = \frac{\Gamma(c_1)\left(\displaystyle\int_0^1 \frac{u^{b-1}\,{}_2F_1\left(\frac{a}{2},\frac{1}{2}+\frac{a}{2};c_1;\frac{4u^2 z_1 z_2}{(-1+(z_1+z_2)u)^2}\right)}{(1-u)^{b-c_1+1}(1+(-z_1-z_2)u)^a}\,\mathrm{d}u\right)}{\Gamma(b)\,\Gamma(c_1-b)}, \left(\sqrt{|z_1|}+\sqrt{|z_2|}<1 \wedge c_1=c_2 \wedge 0<\Re(b) \wedge 0<\Re(c_1) \wedge 0<-\Re(-c_1+b)\right) \vee \left(\sqrt{|z_1|}+\sqrt{|z_2|}<1 \wedge c_1=c_2 \wedge 0<\Re(b) \wedge 0<\Re(c_2) \wedge 0<-\Re(-c_2+b)\right)\right], \left[F_4(a,b,c_1,c_2,z_1,z_2) = \frac{\displaystyle\int_0^\infty \frac{u^{2a-1}\,{}_0F_1\left(;c_1;\frac{z_1 u^2}{4}\right)\,{}_0F_1\left(;c_2;\frac{z_2 u^2}{4}\right)}{\mathrm{e}^u}\,\mathrm{d}u}{\Gamma(2a)}, \sqrt{|z_1|}+\sqrt{|z_2|}<1 \wedge b = a+\frac{1}{2} \wedge 0<\Re(a) \wedge \Re(\sqrt{z_1}+\sqrt{z_2})<1 \wedge \Re(\sqrt{z_1}-\sqrt{z_2})<1 \wedge -\Re(\sqrt{z_1}-\sqrt{z_2})<1 \wedge -\Re(\sqrt{z_1}+\sqrt{z_2})<1\right] \tag{1.4.2}$$

These integral representations are the starting point for the derivation of many of the identities known for the four Appell functions.

Euler-Type and Contiguity Identities

For the purpose of numerically evaluation the four Appell functions over the whole complex plane, instead of numerically evaluating the integral representations, it is simpler, when possible, to evaluate the function using identities. For example, with the exception of AppellF3, the Appell functions admit identities analogous to Euler identities for the hypergeometric function.

These Euler-type identities, as well as contiguity identities for the four Appell functions, are visible using the FunctionAdvisor or directly from the function. For AppellF4, for instance, provided that none of a, b, $a-b$, c_2-a is a non-positive integer,

```
> AppellF4(a, b, c[1], c[2], z[1], z[2]) = AppellF4:-Transformations["Euler"][1](a, b, c[1], c[2], z[1], z[2]);
```

$$F_4(a,b,c_1,c_2,z_1,z_2) = \frac{\Gamma(c_2)\,\Gamma(b-a)\,(-z_2)^{-a}\,F_4\left(a, a-c_2+1, a-b+1, c_1, \frac{1}{z_2}, \frac{z_1}{z_2}\right)}{\Gamma(c_2-a)\,\Gamma(b)} + \frac{\Gamma(c_2)\,\Gamma(a-b)\,(-z_2)^{-b}\,F_4\left(b, 1+b-c_2, b-a+1, c_1, \frac{1}{z_2}, \frac{z_1}{z_2}\right)}{\Gamma(c_2-b)\,\Gamma(a)} \tag{1.5.1}$$

This identity can be used to evaluate AppellF4 at $z_1 = 1$ over the whole complex plane since, in that case, the two variables of the Appell Functions on right-hand side become equal, and that is a special case of AppellF4, expressible in terms of hypergeometric 4F3 functions

```
> eval((1.5.1), z[1] = 1);
```

$$F_4(a,b,c_1,c_2,1,z_2) = \frac{1}{\Gamma(c_2-a)\,\Gamma(b)}\left(\Gamma(c_2)\,\Gamma(b-a)\,(-z_2)^{-a}\,{}_4F_3\left(a, a-c_2+1, \frac{a}{2}-\frac{b}{2}+\frac{c_1}{2}, \frac{a}{2}-\frac{b}{2}+\frac{1}{2}+\frac{c_1}{2}; c_1, a-b+1, a-b+c_1; \frac{4}{z_2}\right)\right) + \frac{1}{\Gamma(c_2-b)\,\Gamma(a)}\left(\Gamma(c_2)\,\Gamma(a-b)\,(-z_2)^{-b}\,{}_4F_3\left(b, 1+b-c_2, \frac{b}{2}-\frac{a}{2}+\frac{c_1}{2}, \frac{b}{2}-\frac{a}{2}+\frac{1}{2}+\frac{c_1}{2}; c_1, b-a+1, b-a+c_1; \frac{4}{z_2}\right)\right) \tag{1.5.2}$$

A contiguity transformation for AppellF4

```
> AppellF4(a, b, c[1], c[2], z[1], z[2]) = AppellF4:-Transformations["Contiguity"][2](a, b, c[1], c[2], z[1], z[2]);
```

$$F_4(a,b,c_1,c_2,z_1,z_2) = F_4(b, a+n, c_1, c_2, z_1, z_2) - \frac{b\,z_1\left(\sum_{k=1}^{n} F_4(a+k, b+1, c_1+1, c_2, z_1, z_2)\right)}{c_1} \tag{1.5.3}$$

$$- \frac{b\, z_2 \left(\sum_{k=1}^{n} F_4(a+k, b+1, c_1, 1+c_2, z_1, z_2) \right)}{c_2}$$

Appell Differential Equations

Each of the four Appell functions satisfy a linear system of partial differential equations, for example for AppellF1

> $FunctionAdvisor(DE, AppellF1)$

$$\left[f(a, b_1, b_2, c, z_1, z_2) = F_1(a, b_1, b_2, c, z_1, z_2), \left[\frac{\partial^2}{\partial z_1^2} f(a, b_1, b_2, c, z_1, z_2) = - \frac{z_2 \left(\frac{\partial^2}{\partial z_1 \partial z_2} f(a, b_1, b_2, c, z_1, z_2) \right)}{z_1} + \frac{((-a - b_1 - 1) z_1 + c) \left(\frac{\partial}{\partial z_1} f(a, b_1, b_2, c, z_1, z_2) \right)}{z_1 (z_1 - 1)} - \frac{b_1 z_2 \left(\frac{\partial}{\partial z_2} f(a, b_1, b_2, c, z_1, z_2) \right)}{z_1 (z_1 - 1)} - \frac{f(a, b_1, b_2, c, z_1, z_2)\, a\, b_1}{z_1 (z_1 - 1)}, \frac{\partial^2}{\partial z_1 \partial z_2} f(a, b_1, b_2, c, z_1, z_2) = - \frac{z_2 \left(\frac{\partial^2}{\partial z_2^2} f(a, b_1, b_2, c, z_1, z_2) \right)}{z_1} - \frac{b_2 \left(\frac{\partial}{\partial z_1} f(a, b_1, b_2, c, z_1, z_2) \right)}{z_2 - 1} + \frac{((-a - b_2 - 1) z_2 + c) \left(\frac{\partial}{\partial z_2} f(a, b_1, b_2, c, z_1, z_2) \right)}{z_1 (z_2 - 1)} - \frac{f(a, b_1, b_2, c, z_1, z_2)\, a\, b_2}{z_1 (z_2 - 1)} \right] \right] \tag{1.6.1}$$

By using differential algebra techniques, this PDE system, as well as the ones corresponding to each of the other Appell functions, can be transformed into an equivalent PDE system where one of the equations is a linear ODE in

z_2 parametrized by z_1. In the case of AppellF1 this linear ODE is of third order and can be computed as follows

```
> F1(z[1], z[2]) = AppellF1(a, b[1], b[2], c, z[1], z[2]);
```

$$F1(z_1, z_2) = F_1(a, b_1, b_2, c, z_1, z_2) \qquad (1.6.2)$$

```
> simplify(op([1, 2], PDEtools:-casesplit(PDEtools:-dpolyform((1.6.2), no_Fn),
      [lex[F1]], ivars = [z[1], z[2]], diffalg)))
```

$$\frac{\partial^3}{\partial z_2^3} F1(z_1, z_2) = \frac{1}{z_2 (z_2 - 1)(z_1 - z_2)} \Big(\big((a + 2 b_2 + 4) z_2^2 + ((-a + b_1 - b_2 - 3) z_1 - c - b_2 - 2) z_2 + z_1 (c - b_1 + 1) \big) \left(\frac{\partial^2}{\partial z_2^2} F1(z_1, z_2) \right) + \big(((2 a + b_2 + 2) z_2 + (-a + b_1 - 1) z_1 - c) \left(\frac{\partial}{\partial z_2} F1(z_1, z_2) \right) + F1(z_1, z_2)\, a\, b_2 \big) (b_2 + 1) \Big) \qquad (1.6.3)$$

This is a linear ODE with four regular singularities, one of which is located at $z_2 = z_1$

```
> DEtools[singularities](subs(F1(z[1], z[2]) = F1(z[2]), (1.6.3)))
```

$$regular = \{0, 1, z_1, \infty\},\ irregular = \varnothing \qquad (1.6.4)$$

When applying the same procedure to the other Appell functions, the result is a fourth order linear ODE with singularities of increasing complexity. The singularities of those fourth order linear ODES behind AppellF2, AppellF3 and AppellF4 can be viewed directly using the Singularities command of the MathematicalFunctions:-Evalf package; for instance for AppellF4 the singularities of the underlying ODE are

```
> fa(syntax, AppellF2)
```

$$F_2(a, b_1, b_2, c_1, c_2, z_1, z_2) \qquad (1.6.5)$$

```
> F2(z[1], z[2]) = AppellF2(a, b[1], b[2], c[1], c[2], z[1], z[2])
```

$$F2(z_1, z_2) = F_2(a, b_1, b_2, c_1, c_2, z_1, z_2) \qquad (1.6.6)$$

```
> simplify(op([1, 2], PDEtools:-casesplit(PDEtools:-dpolyform((1.6.6), no_Fn),
      [lex[F2]], ivars = [z[1], z[2]], diffalg)))
```

$$\frac{\partial^4}{\partial z_2^4} F2(z_1, z_2) = \frac{1}{z_2^2 (z_2 - 1)(z_2 + z_1 - 1)} \Big(-\big((2 a + 2 b_2 - c_1 + 8) z_2^2 + ((z_1 - 2) b_2 + (z_1 - 2) a + (z_1 - 2) c_2 + (-b_1 + 5) z_1 + c_1 - 10) z_2 - 2 (z_1 - 1)(1 + c_2) \big) z_2 \left(\frac{\partial^3}{\partial z_2^3} F2(z_1, z_2) \right) + \big((-b_2^2 + (-4 a + 2 c_1 \qquad (1.6.7)$$

$$-9) b_2 - a^2 + (c_1 - 9) a + 4 c_1 - 14) z_2^2 + (((-z_1 + 2) a + (-z_1 + 2) c_2$$

$$+ (b_1 - 2) z_1 - c_1 + 4) b_2 - (z_1 - 2) (c_2 + 2) a + ((b_1 - 3) z_1 - c_1$$

$$+ 6) c_2 + (2 b_1 - 4) z_1 - 2 c_1 + 8) z_2 + c_2 (z_1 - 1) (1 + c_2)) \left(\frac{\partial^2}{\partial z_2^2}\right.$$

$$\left. F2(z_1, z_2)\right) - \left((((2 a - c_1 + 2) b_2 + 2 (a + 1) (a - c_1 + 2)) z_2 + ((z_1\right.$$

$$- 2) a + (-b_1 + 1) z_1 + c_1 - 2) c_2) \left(\frac{\partial}{\partial z_2} F2(z_1, z_2)\right) + F2(z_1, z_2) a b_2 (a$$

$$\left. - c_1 + 1)\right) (b_2 + 1)\Big)$$

> *MathematicalFunctions:-Evalf:-Singularities*(*AppellF4*(a, b, c_1, c_2, z_1, z_2));

$$\left[0, \frac{(z_1 - 1) (a + b - c_1 + 1) (a + b - c_1 - 2 c_2 + 3)}{(c_1 - 1 - b + a) (-c_1 + 1 - b + a)}, z_1 + 1 - 2 \sqrt{z_1}, z_1 + 1\right. \tag{1.6.8}$$

$$\left. + 2 \sqrt{z_1}, \infty + \infty \mathrm{I}\right]$$

Putting all together

> *FunctionAdvisor*(*AppellF1*);

AppellF1

describe

AppellF1 = *Appell 2-variable hypergeometric function F1*

definition

$$F_1(a, b_1, b_2, c, z_1, z_2) = \sum_{\_k1=0}^{\infty} \sum_{\_k2=0}^{\infty} \frac{(a)_{\_k1 + \_k2} (b_1)_{\_k1} (b_2)_{\_k2} z_1^{\_k1} z_2^{\_k2}}{(c)_{\_k1 + \_k2} \_k1! \_k2!} \qquad |z_1| < 1 \wedge |z_2| < 1$$

classify function

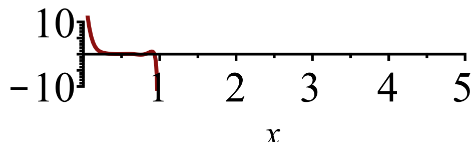
Appell

symmetries

$F_1(a, b_2, b_1, c, z_2, z_1) = F_1(a, b_1, b_2, c, z_1, z_2)$

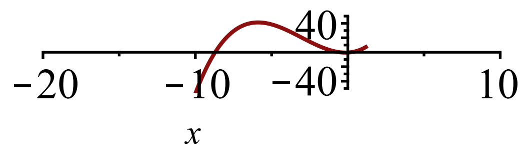
plot

Appell 2-variable hypergeometric function F1



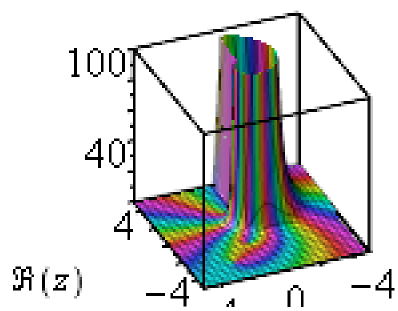
$F_1\left(-\frac{13}{2}, 5, 12, \frac{17}{9}, -\frac{3}{10}, x\right)$

Appell 2-variable hypergeometric function F1

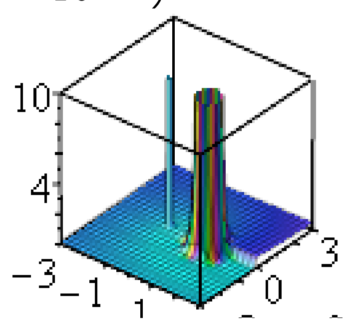


$F_1\left(-\frac{16}{5}, -\frac{5}{2}, \frac{7}{2}, 1, x, \frac{3}{10}\right)$

$F_1\left(\frac{16}{5} + 2\,I, -3, \frac{51}{10}, \frac{17}{10}, -2 + \frac{I}{10}, z\right)$



$F_1\left(\frac{1}{5} - \frac{2\,I}{5}, 5, 7, -\frac{3\,I}{4}, -\frac{7}{10}, z\right)$



singularities

$F_1(a, b_1, b_2, c, z_1, z_2)$

$$\left(c::\mathbb{Z}^{(0,-)} \land a::\mathbb{Z}^{(0,-)} \land b_1::(\neg\, \mathbb{Z}^{(0,-)}) \land a < c\right) \lor \left(c::\mathbb{Z}^{(0,-)} \land a::\mathbb{Z}^{(0,-)} \land b_2::(\neg\, \mathbb{Z}^{(0,-)}) \land a < c\right) \lor \left(c::\mathbb{Z}^{(0,-)} \land a::\mathbb{Z}^{(0,-)} \land b_1::\mathbb{Z}^{(0,-)} \land b_2::\mathbb{Z}^{(0,-)} \land a < c \land b_1 + b_2 < c\right) \lor \left(c::\mathbb{Z}^{(0,-)} \land a::(\neg\, \mathbb{Z}^{(0,-)}) \land b_1::(\neg\, \mathbb{Z}^{(0,-)})\right) \lor \left(c::\mathbb{Z}^{(0,-)} \land a::(\neg\, \mathbb{Z}^{(0,-)}) \land b_2::(\neg\, \mathbb{Z}^{(0,-)})\right) \lor \left(c::\mathbb{Z}^{(0,-)} \land a::(\neg\, \mathbb{Z}^{(0,-)}) \land b_1::\mathbb{Z}^{(0,-)} \land b_2::\mathbb{Z}^{(0,-)} \land b_1 + b_2 < c\right)$$

branch points

$F_1(a, b_1, b_2, c, z_1, z_2)$

$$\left(a::(\neg\, \mathbb{Z}^{(0,-)}) \land b_1::(\neg\, \mathbb{Z}^{(0,-)}) \land z_1 \in [1, \infty + \infty\, \mathrm{I}]\right) \lor \left(a::(\neg\, \mathbb{Z}^{(0,-)}) \land b_2::(\neg\, \mathbb{Z}^{(0,-)}) \land z_2 \in [1, \infty + \infty\, \mathrm{I}]\right)$$

branch cuts

$F_1(a, b_1, b_2, c, z_1, z_2)$

$$\left(a::(\neg\, \mathbb{Z}^{(0,-)}) \land b_1::(\neg\, \mathbb{Z}^{(0,-)}) \land 1 < z_1\right) \lor \left(a::(\neg\, \mathbb{Z}^{(0,-)}) \land b_2::(\neg\, \mathbb{Z}^{(0,-)}) \land 1 < z_2\right)$$

special values

$F_1(a, b_1, b_2, c, z_1, z_2) = 1$	$z_1 = 0 \wedge z_2 = 0$
$F_1(a, b_1, b_2, c, z_1, z_2) = 1$	$a = 0$
$F_1(a, b_1, b_2, c, z_1, z_2) = 1$	$b_1 = 0 \wedge b_2 = 0$
$F_1(a, b_1, b_2, c, z_1, z_2) = {}_2F_1(a, b_2; c; z_2)$	$z_1 = 0$
$F_1(a, b_1, b_2, c, z_1, z_2) = {}_2F_1(a, b_2; c; z_2)$	$b_1 = 0$
$F_1(a, b_1, b_2, c, z_1, z_2) = {}_2F_1(a, b_1; c; z_1)$	$z_2 = 0$
$F_1(a, b_1, b_2, c, z_1, z_2) = {}_2F_1(a, b_1; c; z_1)$	$b_2 = 0$
$F_1(a, b_1, b_2, c, z_1, z_2) = {}_2F_1(a, b_1; c; 1)\ {}_2F_1(a, b_2; c - b_1; z_2)$	$z_1 = 1$
$F_1(a, b_1, b_2, c, z_1, z_2) = {}_2F_1(a, b_2; c; 1)\ {}_2F_1(a, b_1; c - b_2; z_1)$	$z_2 = 1$
$F_1(a, b_1, b_2, c, z_1, z_2) = {}_2F_1(a, b_1 + b_2; c; z_1)$	$z_1 = z_2$
$F_1(a, b_1, b_2, c, z_1, z_2) = {}_3F_2\left(b_1, \frac{a}{2}, \frac{a}{2} + \frac{1}{2}; \frac{c}{2}, \frac{c}{2} + \frac{1}{2}; z_1^2\right)$	$z_1 = -z_2 \wedge b_1 = b_2$
$F_1(a, b_1, b_2, c, z_1, z_2) = {}_1F_0(b_1;; z_1)\ {}_1F_0(b_2;; z_2)$	$c = a \wedge a \neq 0$
$F_1(a, b_1, b_2, c, z_1, z_2) = \frac{z_1\, {}_2F_1(1, a; c; z_1) - z_2\, {}_2F_1(1, a; c; z_2)}{-z_2 + z_1}$	$b_1 = 1 \wedge b_2 = 1 \wedge z_1 \neq z_2$

identities

$F_1(a, b_1, b_2, c, z_1, z_2) = (1 - z_1)^{-b_1} (1 - z_2)^{-b_2} F_1\left(c - a, b_1, b_2, c, \frac{z_1}{-1 + z_1}, \frac{z_2}{-1 + z_2}\right)$	$z_1 \neq 1 \land z_2 \neq 1 \land (a::\mathbb{Z}^{(0,-)} \lor (b_1::\mathbb{Z}^{(0,-)} \land b_2::\mathbb{Z}^{(0,-)}) \lor \neg(1 < z_1 \lor 1 < z_2))$
$F_1(a, b_1, b_2, c, z_1, z_2) = (1 - z_1)^{-a} F_1\left(a, c - b_1 - b_2, b_2, c, \frac{z_1}{-1 + z_1}, \frac{z_1 - z_2}{-1 + z_1}\right)$	$z_1 \neq 1 \land (a::\mathbb{Z}^{(0,-)} \lor (b_1::\mathbb{Z}^{(0,-)} \land b_2::\mathbb{Z}^{(0,-)}) \lor \neg(1 < z_1 \lor 1 < z_2))$
$F_1(a, b_1, b_2, c, z_1, z_2) = (1 - z_1)^{c - a - b_1} (1 - z_2)^{-b_2} F_1\left(c - a, c - b_1 - b_2, b_2, c, z_1, \frac{z_2 - z_1}{-1 + z_2}\right)$	$z_2 \neq 1 \land z_1 \neq 1$
$F_1(a, b_1, b_2, c, z_1, z_2) = \frac{1}{(z_1 - z_2)(-1 + a)}((-1 + c)(F_1(-1 + a, b_1, b_2 - 1, -1 + c, z_1, z_2) - F_1(-1 + a, b_1 - 1, b_2, -1 + c, z_1, z_2)))$	$z_1 \neq z_2 \land a \neq 1 \land c \neq 1$
$F_1(a, b_1, b_2, c, z_1, z_2) = \frac{(a)_n F_1(n + a, b_1, b_2, c, z_1, z_2)}{(a - c + 1)_n} - \left(\sum_{k=1}^{n} \frac{1}{(a - c + 1)_k}\left((-1)^k \binom{n}{k} (1 - c)_k F_1(a, b_1, b_2, c - k, z_1, z_2)\right)\right)$	$c \neq 1 \land ((a - c + 1)::(\neg \mathbb{Z}^{(0,-)}) \lor n \leq \lvert a - c + 1 \rvert)$
$F_1(a, b_1, b_2, c, z_1, z_2) = \frac{1}{-b_1 - b_2 + a}(a F_1(a + 1, b_1, b_2, c, z_1, z_2) - F_1(a, b_1 + 1, b_2, c,$	$a \neq b_1 + b_2$

sum form

$F_1(a, b_1, b_2, c, z_1, z_2) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{(a)_{m+n} (b_1)_m (b_2)_n z_1^m z_2^n}{(c)_{m+n} m! \, n!}$	$\lvert z_1 \rvert < 1 \land \lvert z_2 \rvert < 1$
$F_1(a, b_1, b_2, c, z_1, z_2) = \sum_{k=0}^{\infty} \frac{1}{(c)_k k!} \left((a)_k (b_1)_k \, {}_2F_1(a + k, b_2; c + k; z_2) \, z_1^k \right)$	$\lvert z_1 \rvert < 1$
$F_1(a, b_1, b_2, c, z_1, z_2) = \sum_{k=0}^{\infty} \frac{1}{(c)_k k!} \left((a)_k (b_2)_k \, {}_2F_1(a + k, b_1; c + k; z_1) \, z_2^k \right)$	$\lvert z_2 \rvert < 1$

series

$$series\left(F_1\left(a, b_1, b_2, c, z_1, z_2\right), z_1, 4\right) = {}_2F_1\left(a, b_2; c; z_2\right) + \frac{a\, b_1\, {}_2F_1\left(b_2, a+1; 1+c; z_2\right)}{c} z_1 + \frac{1}{2} \frac{a\, b_1\, (a+1)\, \left(b_1+1\right) {}_2F_1\left(b_2, a+2; 2+c; z_2\right)}{c\,(1+c)} z_1^2 + \frac{1}{6} \frac{a\, b_1\, (a+1)\, \left(b_1+1\right)(a+2)\left(b_1+2\right) {}_2F_1\left(b_2, a+3; 3+c; z_2\right)}{c\,(1+c)\,(2+c)} z_1^3 + \mathrm{O}\left(z_1^4\right)$$

$$series\left(F_1\left(a, b_1, b_2, c, z_1, z_2\right), z_2, 4\right) = {}_2F_1\left(a, b_1; c; z_1\right) + \frac{a\, b_2\, {}_2F_1\left(b_1, a+1; 1+c; z_1\right)}{c} z_2 + \frac{1}{2} \frac{a\, b_2\, (a+1)\, \left(b_2+1\right) {}_2F_1\left(b_1, a+2; 2+c; z_1\right)}{c\,(1+c)} z_2^2 + \frac{1}{6} \frac{a\, b_2\, (a+1)\, \left(b_2+1\right)(a+2)\left(b_2+2\right) {}_2F_1\left(b_1, a+3; 3+c; z_1\right)}{c\,(1+c)\,(2+c)} z_2^3 + \mathrm{O}\left(z_2^4\right)$$

integral form

$F_1(a, b_1, b_2, c, z_1, z_2) = \left(\Gamma(c) \left(\int_0^1 \frac{1}{u^{-c+1+b_1}} \left((1-u)^{-1+b_1} (-u z_1 + 1)^{-c+a}\, {}_2F_1(a, b_2; c-b_1; z_2 u)\right) \mathrm{d}u\right)\right) \Big/ \left(\Gamma(b_1)\, \Gamma(c - b_1)\, (1 - z_1)^{-c+a+b_1}\right)$	$0 < \Re(b_1) \wedge 0 < -\Re(-c + b_1)$
$F_1(a, b_1, b_2, c, z_1, z_2) = \left(\Gamma(c) \left(\int_0^1 \frac{1}{u^{-c+1+b_2}} \left((1-u)^{b_2-1} (-z_2 u + 1)^{-c+a}\, {}_2F_1(a, b_1; c-b_2; u z_1)\right) \mathrm{d}u\right)\right) \Big/ \left(\Gamma(b_2)\, \Gamma(c - b_2)\, (1 - z_2)^{-c+a+b_2}\right)$	$0 < \Re(b_2) \wedge 0 < -\Re(-c + b_2)$
$F_1(a, b_1, b_2, c, z_1, z_2) = \frac{1}{\Gamma(a)\, \Gamma(c-a)} \left(\Gamma(c) \left(\int_0^1 u^{a-1} \ldots\right)\right)$	$0 < \Re(a) \wedge 0 < -\Re(-c + a)$

differentiation rule

$$\frac{\partial}{\partial z_1} F_1(a, b_1, b_2, c, z_1, z_2) = \frac{a\, b_1\, F_1(a+1, b_1+1, b_2, 1+c, z_1, z_2)}{c}$$

$$\frac{\partial^n}{\partial z_1^{\,n}} F_1(a, b_1, b_2, c, z_1, z_2) = \frac{(a)_n\, (b_1)_n\, F_1(n+a, n+b_1, b_2, n+c, z_1, z_2)}{(c)_n}$$

$$\frac{\partial}{\partial z_2} F_1(a, b_1, b_2, c, z_1, z_2) = \frac{a\, b_2\, F_1(a+1, b_1, b_2+1, 1+c, z_1, z_2)}{c}$$

$$\frac{\partial^n}{\partial z_2^{\,n}} F_1(a, b_1, b_2, c, z_1, z_2) = \frac{(a)_n\, (b_2)_n\, F_1(n+a, b_1, n+b_2, n+c, z_1, z_2)}{(c)_n}$$

DE

$f(a, b_1, b_2, c, z_1, z_2) = F_1(a, b_1, b_2, c, z_1, z_2)$	$\frac{\partial^2}{\partial z_1^2} f(a, b_1, b_2, c, z_1, z_2) = -\frac{1}{z_1}\left(z_2 \left(\frac{\partial^2}{\partial z_2 \partial z_1} f(a, b_1, b_2, c, z_1, z_2)\right)\right) + \left(\left((-a - b_1 - 1) z_1 + c\right)\left(\frac{\partial}{\partial z_1} f(a, b_1, b_2, c, z_1, z_2)\right)\right) / \left(z_1 (z_1 - 1)\right) - \left(b_1 z_2 \left(\frac{\partial}{\partial z_2} f(a, b_1, b_2, c, z_1, z_2)\right)\right) / \left(z_1 (z_1 - 1)\right) - \left(f(a, b_1, b_2, c, z_1, z_2)\, a\, b_1\right) / \left(z_1 (z_1 - 1)\right)$	$\frac{\partial^2}{\partial z_2 \partial z_1} f(a, b_1, b_2, c, z_1, z_2) = -\frac{1}{z_1}\left(z_2 \left(\frac{\partial^2}{\partial z_2^2} f(a, b_1, b_2, c, z_1, z_2)\right)\right) - 1 / (z_2 - 1) \left(b_2 \left(\frac{\partial}{\partial z_1} f(a, b_1, b_2, c, z_1, z_2)\right)\right) + \left(\left((-a - b_2 - 1) z_2 + c\right)\left(\frac{\partial}{\partial z_2} f(a, b_1, b_2, c, z_1, z_2)\right)\right) / \left(z_1 (z_2 - 1)\right) - \left(f(a, b_1, b_2, c, z_1, z_2)\, a\, b_2\right) / \left(z_1 (z_2 - 1)\right)$

>

Problem: some formulas in the literature are wrong or miss the conditions indicating when are they valid (exchange with the

Mathematics director of the DLMF - NIST)

16.16 Transformations of Variables

16.16(i) Reduction Formulas

16.16.1 $$F_1(\alpha;\beta,\beta';\beta+\beta';x,y)=(1-y)^{-\alpha}\,{}_2F_1\left({\alpha,\beta \atop \beta+\beta'};\frac{x-y}{1-y}\right),$$

16.16.2 $$F_2(\alpha;\beta,\beta';\gamma,\beta';x,y)=(1-y)^{-\alpha}\,{}_2F_1\left({\alpha,\beta \atop \gamma};\frac{x}{1-y}\right),$$

16.16.3 $$F_2(\alpha;\beta,\beta';\gamma,\alpha;x,y)=(1-y)^{-\beta'}\,F_1\left(\beta;\alpha-\beta',\beta';\gamma;x,\frac{x}{1-y}\right),$$

16.16.4 $$F_3(\alpha,\gamma-\alpha;\beta,\beta';\gamma;x,y)=(1-y)^{-\beta'}\,F_1\left(\alpha;\beta,\beta';\gamma;x,\frac{y}{y-1}\right),$$

16.16.5 $$F_3(\alpha,\gamma-\alpha;\beta,\gamma-\beta;\gamma;x,y)=(1-y)^{\alpha+\beta-\gamma}\,{}_2F_1\left({\alpha,\beta \atop \gamma};x+y-xy\right),$$

From the five formulas above three are wrong in that they are valid only *sometimes*. These problems became evident after we developed the numerical evaluation. The following is an excerpt of an exchange we had with the Mathematics Editor of the DLMF (Digital Library of Mathematical Functions)

```
> restart;
```

This is Appell F1

```
> F1 := %AppellF1(a, b1, b2, b1 + b2, z1, z2)
```

$$F1 := F_1(a, b1, b2, b1+b2, z1, z2) \quad \textbf{(1.8.1)}$$

From the symmetry of this function, swap $b1 \leftrightarrow b2$ and $z1 \leftrightarrow z2$ does not change the value of the function

```
> SF1 := %AppellF1(a, b2, b1, b1 + b2, z2, z1)
```

$$SF1 := F_1(a, b2, b1, b1+b2, z2, z1) \quad \textbf{(1.8.2)}$$

According to NIST page 414, 16.16.1, with no restrictions, if $c = b1 + b2$ we have

```
> formula := (a, b1, b2, c, z1, z2) → (1-z2)^(-a) * hypergeom([a, b1], [b1 + b2], (z1
    -z2)/(1-z2))
```

$$formula := (a, b1, b2, c, z1, z2) \mapsto (1-z2)^{-a}\,{}_2F_1\left(a, b1; b1+b2; \frac{z1-z2}{1-z2}\right) \quad \textbf{(1.8.3)}$$

So, applying 16.16.1 to F1 and SF1 we get

```
> e1 := formula(op(F1))
```

$$e1 := (1-z2)^{-a}\,{}_2F_1\left(a, b1; b1+b2; \frac{z1-z2}{1-z2}\right) \quad \textbf{(1.8.4)}$$

```
> e2 := formula(op(SF1))
```

$$e2 := (1-z1)^{-a}\,{}_2F_1\left(a, b2; b1+b2; \frac{z2-z1}{1-z1}\right) \quad \textbf{(1.8.5)}$$

and because F1 = SF1, we expect that $e1 = e2$.
Give some values to the parameters ... and recall we are evaluating only hypergeometric functions

> $val := [a, b1, b2, c, z1, z2] =\sim \left[\frac{17}{5}, \frac{37}{10}, -\frac{11}{5}, \frac{3}{2}, 2-I, 2+4\cdot I\right]$

$$val := \left[a = \frac{17}{5}, b1 = \frac{37}{10}, b2 = -\frac{11}{5}, c = \frac{3}{2}, z1 = 2 - \mathrm{I}, z2 = 2 + 4\,\mathrm{I}\right] \quad \textbf{(1.8.6)}$$

> $evalf(subs(val, e1))$

$$13.79273401 - 9.715912626\,\mathrm{I} \quad \textbf{(1.8.7)}$$

> $evalf(subs(val, e2))$

$$-16.86942635 - 0.2468272677\,\mathrm{I} \quad \textbf{(1.8.8)}$$

In other words, "*the special value of F1 is unexpectedly different from the special value of SF1 even when these two are the same function*". What happened?

We were expecting $e1 = e2$ so that

> $zero := e1 - e2 = 0$

$$zero := (1-z2)^{-a}\, {}_2F_1\left(a, b1; b1+b2; \frac{z1-z2}{1-z2}\right) - (1-z1)^{-a}\, {}_2F_1\left(a, b2; b1 + b2; \frac{z2-z1}{1-z1}\right) = 0 \quad \textbf{(1.8.9)}$$

We know at this point that *zero* is not actually 0 but

> $evalf(subs(val, zero))$

$$30.66216036 - 9.469085358\,\mathrm{I} = 0. \quad \textbf{(1.8.10)}$$

Change variables from $\{z1, z2\}$ to some $\{x, y\}$

> $x = 1 - z1, y = 1 - z2$

$$x = 1 - z1, y = 1 - z2 \quad \textbf{(1.8.11)}$$

> $solve(\{\%\}, \{z1, z2\})$

$$\{z1 = -x + 1, z2 = -y + 1\} \quad \textbf{(1.8.12)}$$

Substitute this in *zero*

> $subs(\%, zero)$

$$y^{-a}\, {}_2F_1\left(a, b1; b1+b2; \frac{-x+y}{y}\right) - x^{-a}\, {}_2F_1\left(a, b2; b1+b2; \frac{-y+x}{x}\right) = 0 \quad \textbf{(1.8.13)}$$

Now substitute $y = -x\cdot z + x$ in order to have a simpler argument for the 2F1 function

> $subs(y = -x\cdot z + x, \%)$

$$(-x\,z + x)^{-a}\, {}_2F_1\left(a, b1; b1+b2; -\frac{x\,z}{-x\,z+x}\right) - x^{-a}\, {}_2F_1(a, b2; b1+b2; z) = 0 \quad \textbf{(1.8.14)}$$

Isolate the simplest 2F1

> $isolate\big(normal(\%), {}_2F_1(a, b2; b1+b2; z)\big)$

$${}_2F_1(a, b2; b1+b2; z) = \frac{(-x\,z+x)^{-a}\, {}_2F_1\left(a, b1; b1+b2; \frac{z}{z-1}\right)}{x^{-a}} \quad \textbf{(1.8.15)}$$

Collect x in the power

> $(-x z+x)=x\cdot(1-z)$

$$-x z+x=x\,(1-z) \tag{1.8.16}$$

> $ee := subs(\mathbf{(1.8.16)}, \mathbf{(1.8.15)})$

$$ee := {}_2F_1(a, b2; b1+b2; z) = \frac{(x\,(1-z))^{-a}\, {}_2F_1\left(a, b1; b1+b2; \frac{z}{z-1}\right)}{x^{-a}} \tag{1.8.17}$$

And by inspection we arrived at something we can check in NIST: this is formula 15.8.1 on page 390, provided that we can split

$$(x\,(1-z))^{-a} = (1-z)^{-a}\, x^{-a}$$

The formula for distributing powers, however, involves an exponential not present in the above and that is not always equal to 1:

> $(A\,B)^{\alpha} = A^{\alpha}\, B^{\alpha}\, \mathrm{e}^{2\,I\,\pi\,\alpha \left\lfloor \frac{\pi - arg(A) - arg(B)}{2\,\pi} \right\rfloor}$

$$(A\,B)^{\alpha} = A^{\alpha}\, B^{\alpha}\, \mathrm{e}^{2\,I\,\pi\,\alpha \left\lfloor \frac{\pi - arg(A) - arg(B)}{2\,\pi} \right\rfloor} \tag{1.8.18}$$

In our case, the dummies A, B, α are

> $subs(A = x, B = (1-z), \text{alpha} = -a, \mathbf{(1.8.18)})$

$$(x\,(1-z))^{-a} = x^{-a}\,(1-z)^{-a}\, \mathrm{e}^{-2\,I\,\pi\,a \left\lfloor \frac{\pi - arg(x) - arg(1-z)}{2\,\pi} \right\rfloor} \tag{1.8.19}$$

So, substitute this value in ee

> $ee1 := subs(\mathbf{(1.8.19)}, ee)$

$$ee1 := {}_2F_1(a, b2; b1+b2; z) = (1-z)^{-a}\, \mathrm{e}^{-2\,I\,\pi\,a \left\lfloor \frac{\pi - arg(x) - arg(1-z)}{2\,\pi} \right\rfloor}\, {}_2F_1\left(a, b1; b1+b2; \frac{z}{z-1}\right) \tag{1.8.20}$$

So *"For 16.13.1 in NIST-DLMF to be correct, the exponential above should be equal to 1"*

Let's check the value of the exponential above for the values we used in **(1.8.6)**. For that purpose express *x, y* and *z* in terms of the original *z1* and *z2* variables of the Appell function

> $x = 1 - z1, y = 1 - z2, y = -x\cdot z + x$

$$x = 1 - z1, y = 1 - z2, y = -x\,z + x \tag{1.8.21}$$

> $solve(\{\%\}, \{x, y, z\})$

$$\left\{x = 1 - z1, y = 1 - z2, z = \frac{z1 - z2}{-1 + z1}\right\} \tag{1.8.22}$$

> $subs(\mathbf{(1.8.22)}, ee1)$:
> $ee2 := simplify(\%)$

$$ee2 := {}_2F_1\left(a, b2; b1+b2; \frac{z1-z2}{-1+z1}\right) = \left(\frac{-1+z2}{-1+z1}\right)^{-a} \mathrm{e}^{-2\,I\,\pi\,a \left\lfloor \frac{\pi - arg(1-z1) - arg\left(\frac{-1+z2}{-1+z1}\right)}{2\,\pi} \right\rfloor}\, {}_2F_1\Big(a, b1; b1+b2; \tag{1.8.23}$$

$$\left.\frac{z2 - z1}{-1 + z2}\right)$$

Check now the value of this exponential, if the NIST textbook formula were correct, this value should be equal to 1

> $E := indets(\textbf{(1.8.23)}, \text{'exp}(anything)\text{'})[1]$

$$E := \mathrm{e}^{-2\,\mathrm{I}\,\pi\,a \left\lfloor \frac{\pi - arg(1 - z1) - arg\left(\frac{-1 + z2}{-1 + z1}\right)}{2\,\pi} \right\rfloor} \quad \textbf{(1.8.24)}$$

> $subs(val, E)$

$$\mathrm{e}^{-\frac{34\,\mathrm{I}}{5}\,\pi \left\lfloor \frac{\pi - arg(-1 + \mathrm{I}) - arg\left(-\frac{3}{2} + \frac{5\,\mathrm{I}}{2}\right)}{2\,\pi} \right\rfloor} \quad \textbf{(1.8.25)}$$

> $evalf(\%)$

$$-0.8090169945 + 0.5877852522\,\mathrm{I} \quad \textbf{(1.8.26)}$$

It is not equal to 1. We conclude that $e1 \neq e2$, the formula NIST 16.13.1 is not correct for these values, and in fact we see this value **(1.8.26)** is precisely the ratio between $e1$ and $e2$ when evaluated at the numbers provided in **(1.8.6)**

> $e1$

$$(1 - z2)^{-a}\,{}_2F_1\left(a, b1; b1 + b2; \frac{z1 - z2}{1 - z2}\right) \quad \textbf{(1.8.27)}$$

> $e2$

$$(1 - z1)^{-a}\,{}_2F_1\left(a, b2; b1 + b2; \frac{z2 - z1}{1 - z1}\right) \quad \textbf{(1.8.28)}$$

> $subs\left(val, \frac{e1}{e2}\right)$

$$\frac{{}_2F_1\left(\frac{17}{5}, \frac{37}{10}; \frac{3}{2}; \frac{20}{17} + \frac{5\,\mathrm{I}}{17}\right)(-1 + \mathrm{I})^{17/5}}{(-1 - 4\,\mathrm{I})^{17/5}\,{}_2F_1\left(\frac{17}{5}, -\frac{11}{5}; \frac{3}{2}; \frac{5}{2} - \frac{5\,\mathrm{I}}{2}\right)} \quad \textbf{(1.8.29)}$$

> $evalf(\%)$

$$-0.8090169934 + 0.5877852558\,\mathrm{I} \quad \textbf{(1.8.30)}$$

In conclusion, formula 16.13.1 on page 413 of the DLMF misses stating that the formula is valid only when the exponential is equal to 1

> $E = 1$

$$\mathrm{e}^{-2\,\mathrm{I}\,\pi\,a \left\lfloor \frac{\pi - arg(1 - z1) - arg\left(\frac{-1 + z2}{-1 + z1}\right)}{2\,\pi} \right\rfloor} = 1 \quad \textbf{(1.8.31)}$$

A similar situation happens with the other two hypergeometric special values of AppellF2 and AppellF3, and in fact with several Appell formulas we got from the literature. Detecting the incorrect formulas in the literature and deriving a correction for them was the main source of slowdown in the project.

>

Appell Functions (numeric)

Goals

• Compute these Appell functions ***over the whole complex plane***

• Considering that ***this is a research problem***, implement different methods and flexible optional arguments to allow for:

a) comparison between methods (both performance and correctness),

b) investigation of a single method in different circumstances.

• Develop a computational structure that can be reused with other special functions (abstract code and provide the main options), and that could also be translated to C (so: only one numerical implementation, not 100 special function numerical implementations)

Limitation: the Maple original evalf command does not accept optional arguments

The cost of numerically evaluating an Appell function

• If it is a special hypergeometric case, then between 1 to 2 hypergeometric functions

• Next simplest case (series/recurrence below) 3 to 4 hypergeometric functions plus adding somewhat large formulas that involve only arithmetic operations up to 20,000 times (frequently less than 100 times)

• Next simplest case: the formulas themselves are power series with hypergeometric function coefficients; these cases frequently converge rapidly but may involve the numerical evaluation of up to hundreds of hypergeometric functions to get the value of a single Appell function.

Strategy for the numerical evaluation of Appell functions (or other functions ...)

The numerical evaluation flows orderly according to:
1) check whether it is a singular case
2) check whether it is a special value
3) compute the value using a series derived from a recurrence related to the underlying ODE
4) perform an sum using an infinite sum formula, checking for convergence
5) perform the numerical integration of the ODE underlying the given Appell function
6) perform a sequence of concatenated Taylor series expansions

Examples

```
> restart; with(MathematicalFunctions, Evalf); with(Evalf);
```

$$[Evalf] \tag{2.1.1.1}$$

$\{Add, Evalb, Zoom, QuadrantNumbers, Singularities, GenerateRecurrence, PairwiseSummation\}$ **(2.1.1.1)**

> $\%AppellF1\left(-4, -\frac{6}{5}+I, -\frac{3 I}{10}, -3, \frac{1}{2}, 1+\frac{3 I}{5}\right)$

$$F_1\left(-4, -\frac{6}{5}+I, -\frac{3 I}{10}, -3, \frac{1}{2}, 1+\frac{3 I}{5}\right) \quad \textbf{(2.1.1.2)}$$

> $Evalf(\%)$

```
case: abs(z1) < 1, abs(z2) > 1; swapping parameters and z1
<-> z2
-> Numerical evaluation of AppellF1(-4., -.3000000000*I,
-1.200000000+1.*I, -3., 1.+.6000000000*I, .5000000000)
   -> AppellF1, checking singular cases
   <- AppellF1, singular case detected
```

CPU time elapsed during evaluation: .7e-2 seconds

$$\mathrm{Float}(\infty) + \mathrm{Float}(\infty)\, I \quad \textbf{(2.1.1.3)}$$

> $AppellF4\left(-4, -\frac{6}{5}+I, -\frac{3 I}{10}, \frac{I}{2}, \frac{1}{2}, 1+\frac{3 I}{5}\right)$

$$F_4\left(-4, -\frac{6}{5}+I, -\frac{3 I}{10}, \frac{I}{2}, \frac{1}{2}, 1+\frac{3 I}{5}\right) \quad \textbf{(2.1.1.4)}$$

> $Evalf(\%)$

```
case: abs(z1) < 1, abs(z2) > 1; swapping parameters and z1
<-> z2
-> Numerical evaluation of AppellF4(-4., -1.200000000+1.*
I, .5000000000*I, -.3000000000*I, 1.+.6000000000*I,
.5000000000)
-> AppellF4, checking singular cases
   -> AppellF4, trying special values
   <- special values of AppellF4 successful
```

CPU time elapsed during evaluation: .67e-1 seconds

$$-38.99294789 + 14.17214182\, I \quad \textbf{(2.1.1.5)}$$

> $F3 := AppellF3\left(\frac{-4}{3}, -\frac{6}{5}+I, -\frac{3 I}{10}, \frac{I}{2}, \frac{1}{2}, 1+\frac{3 I}{5}, \frac{1}{10}\right)$

$$F3 := F_3\left(\frac{I}{2}, -\frac{4}{3}, -\frac{6}{5}+I, -\frac{3 I}{10}, \frac{1}{2}, \frac{1}{10}, 1+\frac{3 I}{5}\right) \quad \textbf{(2.1.1.6)}$$

> $Evalf(F3)$

```
case: abs(z1) < 1, abs(z2) > 1; swapping parameters and z1
<-> z2
-> Numerical evaluation of AppellF3(-1.333333333,
.5000000000*I, -.3000000000*I, -1.200000000+1.*I,
.5000000000, 1.+.6000000000*I, .1000000000)
   -> AppellF3, checking singular cases
   -> AppellF3, trying special values
   -> AppellF3, trying series based on recurrence
```

CPU time elapsed during evaluation: .118 seconds

$$0.5978088090 + 0.5841674146\, I \quad \textbf{(2.1.1.7)}$$

> $Evalf(F3, formulas)$

```
case: abs(z1) < 1, abs(z2) > 1; swapping parameters and z1
<-> z2
-> Numerical evaluation of AppellF3(-1.333333333,
```

```
.5000000000*I, -.3000000000*I, -1.200000000+1.*I,
.5000000000, 1.+.6000000000*I, .1000000000)
   -> AppellF3, checking singular cases
   -> AppellF3, trying formulas
      case 1: abs(z2) < 1, using single sum with
hypergeometric coefficients
          -> entering Add with: 'formula' = (-1)^k*
pochhammer(-1.333333333,k)*pochhammer(.5000000000*I,k)*
pochhammer(-.3000000000*I,k)*pochhammer(-1.200000000+1.*I,
k)/k!/pochhammer(-.5000000000+k,k)/pochhammer(.5000000000,
2*k)*hypergeom([-1.333333333+k, -.3000000000*I+k],
[.5000000000+2*k],1.+.6000000000*I)*hypergeom(
[.5000000000*I+k, -1.200000000+1.*I+k],[.5000000000+2*k],
.1000000000)*(.1000000000+.6000000000e-1*I)^k
          <- exiting Add with .597808808973+.584167414632*
I; after adding 6 terms
```

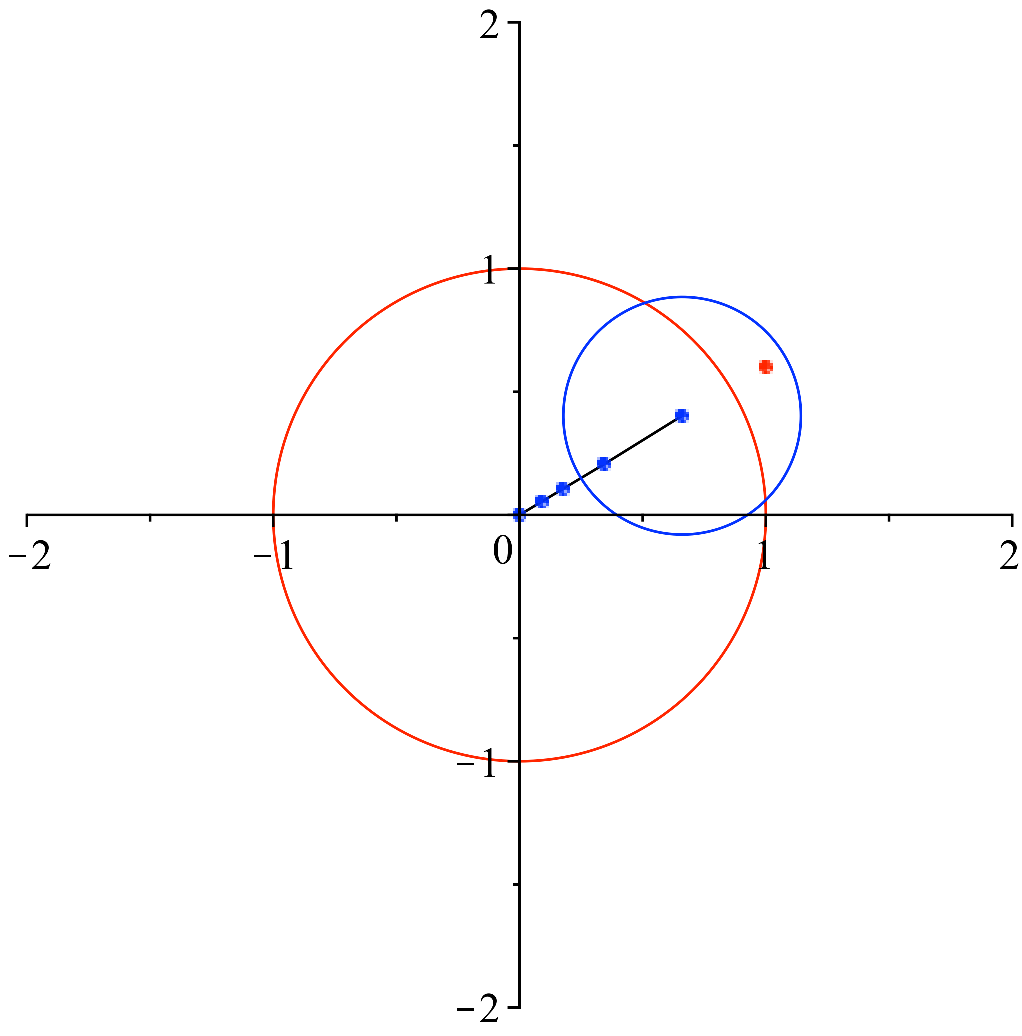
CPU time elapsed during evaluation: .263 seconds

$$0.5978088090 + 0.5841674146\, I \tag{2.1.1.8}$$

> *Evalf*(*F3*, *differentialequation*)

Error, (in MathematicalFunctions:-Evalf) invalid input: too many and/or wrong type of arguments passed to EvalfAppell:-Evalf; first unused argument is differentialequation

> *Evalf*(*F3*, *allowswapping* = *false*, *taylor*, *plot*, *time*, *quiet*)



CPU time elapsed during evaluation: .221 seconds

$$0.5978088090 + 0.5841674146\,I \quad \textbf{(2.1.1.9)}$$

> *map*(abs, *Singularities*(*F3*))

$$\left[0, \frac{1}{9}, 1, \infty\right] \quad \textbf{(2.1.1.10)}$$

> $F33 := F_3\left(1, 2, -3, 1, \frac{180}{17}, \frac{2}{5} - \frac{I}{5}, 60\right)$

$$F33 := F_3\left(1, 2, -3, 1, \frac{180}{17}, \frac{2}{5} - \frac{I}{5}, 60\right) \quad \textbf{(2.1.1.11)}$$

> *map*(abs, *Singularities*(*F33*))

$$\left[0, \frac{\sqrt{2}}{2}, 1, \infty\right] \quad \textbf{(2.1.1.12)}$$

> *Evalf*(*F33*, *taylor*)

```
case: abs(z1) < 1, abs(z2) > 1; swapping parameters and
z1 <-> z2
-> Numerical evaluation of AppellF3(2., 1., 1., -3.,
10.58823529, 60., .4000000000-.2000000000*I)
   -> AppellF3, checking singular cases
   -> AppellF3, trying concatenated Taylor series
expansions
      AppellF3(2.,1.,1.,-3.,10.58823529,60.,.4000000000
-.2000000000*I) is polynomial in z2, computing a simple
series around the origin
```

CPU time elapsed during evaluation: .15e-1 seconds

$$-0.1599786215 - 0.06501291731\,I \quad \textbf{(2.1.1.13)}$$

OK ... Make it not polynomial

> $F33 := \%AppellF3\left(1, 2, \frac{-3}{2}, 1, \frac{180}{17}, \frac{2}{5} - \frac{I}{5}, 60\right)$

$$F33 := F_3\left(1, 2, -\frac{3}{2}, 1, \frac{180}{17}, \frac{2}{5} - \frac{I}{5}, 60\right) \quad \textbf{(2.1.1.14)}$$

> *Evalf*(*F33*)

```
case: abs(z1) < 1, abs(z2) > 1; swapping parameters and
z1 <-> z2
-> Numerical evaluation of AppellF3(2., 1., 1.,
-1.500000000, 10.58823529, 60., .4000000000-.2000000000*
I)
   -> AppellF3, checking singular cases
   -> AppellF3, trying special values
   -> AppellF3, trying series based on recurrence
```

CPU time elapsed during evaluation: .2e-2 seconds

$$-0.1720188468 - 0.06454452081\,I \quad \textbf{(2.1.1.15)}$$

Do not allow swapping, it gets tackled solving the differential equation

> *Evalf*(*F33*, *allowswapping* = *false*)

```
-> Numerical evaluation of AppellF3(1., 2., -1.500000000,
1., 10.58823529, .4000000000-.2000000000*I, 60.)
   -> AppellF3, checking singular cases
   -> AppellF3, trying special values
   -> AppellF3, trying formulas
```

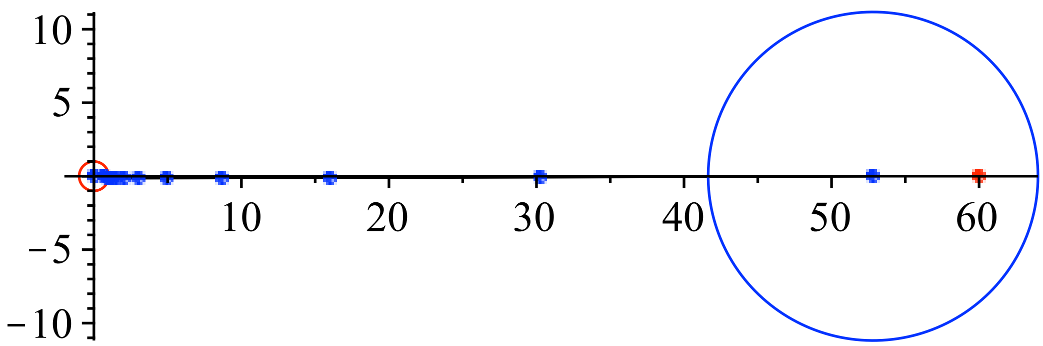
```
   -> AppellF3, trying the numerical integration of the
related ODE
      -> computing initial conditions: all of AppellF3,
AppellF3', AppellF3'' and AppellF3''' at z2 = .6717514421
         case: abs(z1) < abs(z2) < 1; swapping parameters
and z1 <-> z2
         -> Numerical evaluation of AppellF3(2., 1., 1.,
-1.500000000, 10.58823529, .6717514421, .4000000000
-.2000000000*I)
            -> AppellF3, checking singular cases
            -> AppellF3, trying special values
            -> AppellF3, trying series based on
recurrence
         case: abs(z1) < abs(z2) < 1; swapping parameters
and z1 <-> z2
         -> Numerical evaluation of AppellF3(3., 1., 2.,
-1.500000000, 11.58823529, .6717514421, .4000000000
-.2000000000*I)
            -> AppellF3, checking singular cases
            -> AppellF3, trying special values
            -> AppellF3, trying series based on
recurrence
         case: abs(z1) < abs(z2) < 1; swapping parameters
and z1 <-> z2
         -> Numerical evaluation of AppellF3(4., 1., 3.,
-1.500000000, 12.58823529, .6717514421, .4000000000
-.2000000000*I)
            -> AppellF3, checking singular cases
            -> AppellF3, trying special values
            -> AppellF3, trying series based on
recurrence
         case: abs(z1) < abs(z2) < 1; swapping parameters
and z1 <-> z2
         -> Numerical evaluation of AppellF3(5., 1., 4.,
-1.500000000, 13.58823529, .6717514421, .4000000000
-.2000000000*I)
            -> AppellF3, checking singular cases
            -> AppellF3, trying special values
            -> AppellF3, trying series based on
recurrence
      -> computing an extra step at z2 = .9506661558
-.1394573569*I
```

CPU time elapsed during evaluation: 3.825 seconds

$$-0.1720188468 - 0.06454452082\,I \qquad \textbf{(2.1.1.16)}$$

Use a concatenated sequence of Taylor series, no swapping, plot the path, tell the time and nothing more

> $Evalf(F33, taylor, allowswapping = false, plot, time, quiet)$



CPU time elapsed during evaluation: .71e-1 seconds

$$-0.1720188468 - 0.06454452081\, I \tag{2.1.1.17}$$

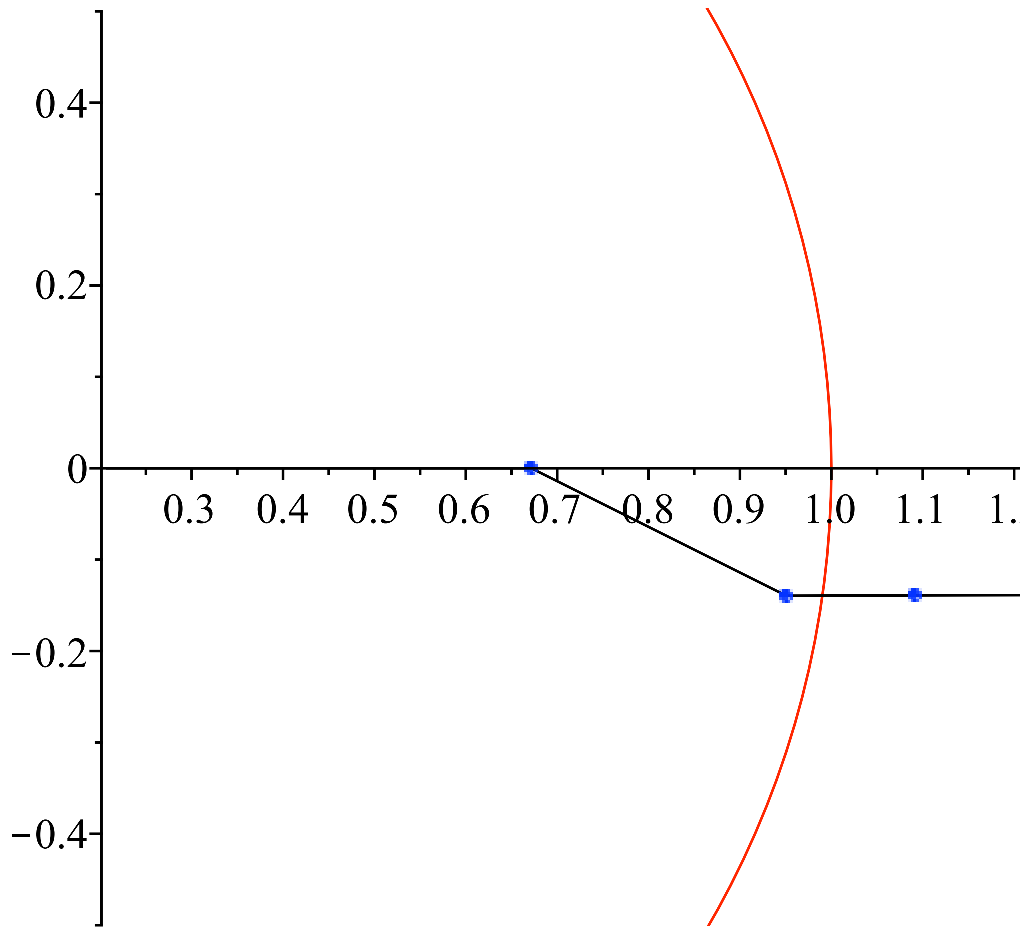
> *Singularities*(*F33*)

$$\left[0, -\frac{1}{2} + \frac{I}{2}, 1, \infty + \infty\, I\right] \tag{2.1.1.18}$$

> *evalf*(abs(%[2]))

$$0.7071067810 \tag{2.1.1.19}$$

> $Zoom\left(\left[\%, \frac{1}{2}\right]\right)$



>

Series/recurrence

Derivation of formulas for AppellF2

The underlying ODE is obtained using a differential algebra process on the linear PDE system satisfied by AppellF2

> $restart;\ with(MathematicalFunctions{:}\text{-}Evalf)$

$$\{Add, Evalb, Zoom, QuadrantNumbers, Singularities, GenerateRecurrence, PairwiseSummation\} \quad \textbf{(2.1.2.1)}$$

> $b1, b2, c1, c2, z1, z2 := b_1, b_2, c_1, c_2, z_1, z_2$

$$b1, b2, c1, c2, z1, z2 := b_1, b_2, c_1, c_2, z_1, z_2 \quad \textbf{(2.1.2.2)}$$

> $F2 := AppellF2(a, b1, b2, c1, c2, z1, z2)$

$$F2 := F_2(a, b_1, b_2, c_1, c_2, z_1, z_2) \quad \textbf{(2.1.2.3)}$$

How does this work?

```
> ode_F2 := diff(u(t), t$4) = (-((2 a + 2 b2-c1 + 8) t^2 + ((z1-2) b2 + (z1-2) a
      + (z1-2) c2 + (-b1 + 5) z1 + c1-10) t- (2 c2 + 2) (z1-1)) t·(diff(u(t), t
      $3)) + ((-b2^2 + (-4 a + 2 c1-9) b2-a^2 + (c1-9) a + 4 c1-14) t^2
      + (((-z1 + 2) a + (-z1 + 2) c2 + (b1-2) z1-c1 + 4) b2- (c2 + 2) (z1
      -2) a + ((b1-3) z1-c1 + 6) c2 + (2 b1-4) z1-2 c1 + 8) t + c2·(c2 + 1) (z1
      -1)) (diff(u(t), t$2))-((2·(((a-(1/2) c1 + 1) b2 + (1 + a) (a-c1 + 2)) t
      + (1/2·((z1-2) a + (-b1 + 1) z1 + c1-2)) c2)) (diff(u(t), t)) + u(t) a b2
      ·(a-c1 + 1)) (b2 + 1))/(t^2·(t-1) (t + z1-1))
```

$$ode\_F2 := \frac{d^4}{dt^4} u(t) = \frac{1}{t^2 (t-1) (t+z_1-1)} \Bigg(-\Big(\big((2a+2b_2-c_1+8) t^2 + \big((z_1-2) b_2 + (z_1-2) a + (z_1-2) c_2 + (-b_1+5) z_1 + c_1 - 10 \big) t - (2c_2+2)(z_1-1) \big) t \left(\frac{d^3}{dt^3} u(t) \right) + \Big(\big(-b_2^2 + (-4a+2c_1-9) b_2 - a^2 + (c_1-9) a + 4c_1 - 14 \big) t^2 + \Big(\big((-z_1+2) a + (-z_1+2) c_2 + (b_1-2) z_1 - c_1 + 4 \big) b_2 - (c_2+2)(z_1-2) a + \big((b_1-3) z_1 - c_1 + 6 \big) c_2 + (2b_1-4) z_1 - 2c_1 + 8 \Big) t + c_2 (c_2+1)(z_1-1) \Big) \left(\frac{d^2}{dt^2} u(t) \right) - \left(\left(2 \left(\left(a - \frac{c_1}{2} + 1 \right) b_2 + (1+a)(a-c_1+2) \right) t + 2 \left(\frac{(z_1-2) a}{2} + \frac{(-b_1+1) z_1}{2} + \frac{c_1}{2} - 1 \right) c_2 \right) \left(\frac{d}{dt} u(t) \right) + u(t)\, a\, b_2 (a-c_1+1) \right) (b_2+1) \Bigg) \tag{2.1.2.4}$$

The system with initial conditions: at this point we just say that u(0) = C0, D(u)(0) = C1, but as it makes sense (this is just a taylor expansion), we will see that C0 = Heun(z0) and C1 = HeunPrime(z0)

```
sys_z0 := {ode_F2, u(0) = C0, D(u)(0) = C1, D(D(u))(0) = C2, D(D(D(u)))(0)
      = C3}
```

$$sys\_z0 := \Bigg\{ \frac{d^4}{dt^4} u(t) = \frac{1}{t^2 (t-1) (t+z_1-1)} \Bigg(-\Big(\big((2a+2b_2-c_1+8) t^2 + \big((z_1-2) b_2 + (z_1-2) a + (z_1-2) c_2 + (-b_1+5) z_1 + c_1 - 10 \big) t - (2c_2+2)(z_1-1) \big) t \left(\frac{d^3}{dt^3} u(t) \right) + \Big(\big(-b_2^2 + (-4a+2c_1-9) b_2 \tag{2.1.2.5}$$

$$- a^2 + (c_1 - 9)\, a + 4\, c_1 - 14)\, t^2 + (((-z_1 + 2)\, a + (-z_1 + 2)\, c_2$$

$$+ (b_1 - 2)\, z_1 - c_1 + 4)\, b_2 - (c_2 + 2)\, (z_1 - 2)\, a + ((b_1 - 3)\, z_1 - c_1$$

$$+ 6)\, c_2 + (2\, b_1 - 4)\, z_1 - 2\, c_1 + 8)\, t + c_2\, (c_2 + 1)\, (z_1 - 1))\, \left(\frac{\mathrm{d}^2}{\mathrm{d}t^2}\right.$$

$$\left. u(t)\right) - \left(\left(2\left(\left(a - \frac{c_1}{2} + 1\right) b_2 + (1 + a)\, (a - c_1 + 2)\right) t\right.\right.$$

$$+ 2\left(\frac{(z_1 - 2)\, a}{2} + \frac{(-b_1 + 1)\, z_1}{2} + \frac{c_1}{2} - 1\right) c_2 \Big) \left(\frac{\mathrm{d}}{\mathrm{d}t}\, u(t)\right)$$

$$\left.\left. + u(t)\, a\, b_2\, (a - c_1 + 1)\right) (b_2 + 1)\right), u(0) = C0, \mathrm{D}(u)(0) = C1,$$

$$\mathrm{D}^{(2)}(u)(0) = C2, \mathrm{D}^{(3)}(u)(0) = C3 \Big\}$$

Compute the recurrence now

```
> C_z0_rec := simplify(gfun:-diffeqtorec(sys_z0, u(t), v(n)))
```

$$C\_z0\_rec := \Big\{((z_1 - 2)\, n + (a - b_1 + 1)\, z_1 - 2\, a + c_1 - 2)\, (n + c_2)\, (n \qquad \textbf{(2.1.2.6)}$$

$$+ 1)\, (n + b_2 + 1)\, v(n + 1) - (n + 2)\, (n + 1)\, (n + c_2 + 1)\, (n + c_2)\, (z_1$$

$$- 1)\, v(n + 2) + v(n)\, (n + b_2 + 1)\, (n + b_2)\, (a + n)\, (a + n - c_1 + 1),$$

$$v(0) = C0, v(1) = C1, v(2) = \frac{C2}{2}, v(3) = \frac{C3}{6} \Big\}$$

This recurrence can be transformed into a procedure that will *"compute the nth coefficient, departing from the previous one"*

```
> U := GenerateRecurrence(F2)
```

(2.1.2.7)

```
U := proc(i)
    option hfloat, cache;
    if(i::nonnegint, ((a + i - b[1] - 1) * z[1] - 2 * a - 2 * i + c[1] + 2) * (i
    + b[2] - 1) * thisproc(i - 1) / ((z[1] - 1) * (i + c[2] - 1) * i) + (i + b
    [2] - 1) * (i + b[2] - 2) * (a + i - 2) * (a + i - c[1] - 1)
    * thisproc(i - 2) / (i * (i - 1) * (i + c[2] - 1) * (i + c[2] - 2) * (z[1
    ] - 1)), 'procname(args)')
end
```

The formula we are exploring here is

```
> F2 = Sum(U(j)z2^j, j = 0 ..infinity);
```

$$F_2(a, b_1, b_2, c_1, c_2, z_1, z_2) = \sum_{j=0}^{\infty} U(j)\, z_2^{\,j} \quad \textbf{(2.1.2.8)}$$

Let's see

> $U(0)$

$$ {}_2F_1(a, b_1; c_1; z_1) \quad \textbf{(2.1.2.9)}$$

> $U(1)$

$$\frac{a\, b_2\, {}_2F_1(b_1, 1+a; c_1; z_1)}{c_2} \quad \textbf{(2.1.2.10)}$$

> $U(2)$

$$\frac{a\, b_2\, (1+a)\, (b_2+1)\, {}_2F_1(b_1, 2+a; c_1; z_1)}{2\, c_2\, (c_2+1)} \quad \textbf{(2.1.2.11)}$$

> $U(3)$

$$\frac{a\, b_2\, {}_2F_1(b_1, a+3; c_1; z_1)\, (b_2+2)\, (b_2+1)\, (2+a)\, (1+a)}{6\, c_2\, (c_2^{\,2}+3\, c_2+2)} \quad \textbf{(2.1.2.12)}$$

>

Numerical integration of an underlying differential equation (ODEs and dsolve/numeric)

The AppellF1 ODE

```
> restart;
> F1 := AppellF1(a, b1, b2, c, x, z);
```

$$F1 := F_1(a, b1, b2, c, x, z) \quad \textbf{(2.1.3.1)}$$

```
> AppellF1ODE := subs(z1 = x, y = f, diff(y(z), z, z, z) = (((a + 2*b2 + 4)*z^2
     + ((-a + b1-b2-3)*z1-c-b2-2)*z + z1*(c-b1 + 1))*(diff(y(z), z, z))
     - (b2 + 1)*(((-2*a-b2-2)*z + (a-b1 + 1)*z1 + c)*(diff(y(z), z))
     -y(z)*a*b2))/(z*(z-1)*(z1-z)));
```

$$AppellF1ODE := \frac{d^3}{dz^3} f(z) = \frac{1}{z\,(z-1)\,(x-z)}\Bigg(\Big((a+2\,b2+4)\,z^2+((-a+b1-b2-3)\,x-c-b2-2)\,z+x\,(c-b1+1)\Big)\left(\frac{d^2}{dz^2} f(z)\right)-(b2+1)\left(((-2\,a-b2-2)\,z+(a-b1+1)\,x+c)\left(\frac{d}{dz} f(z)\right)-f(z)\,a\,b2\right)\Bigg) \quad \textbf{(2.1.3.2)}$$

A first order ODE system version of this equation: we see YP[n] representing the nth derivative, plus adding the corresponding auxiliary ODE

```
> DEtools[convertsys](AppellF1ODE, [f(0) = Y0, D(f)(0) = DY0, D^(2)(f)(0)
     = DDY0], f(z), z)
```

(2.1.3.3)

$$\left[\left[YP_1 = Y_2, YP_2 = Y_3, YP_3 = \frac{1}{z\,(z-1)\,(x-z)}\Big(\big((a+2\,b2+4)\,z^2+((-a+b1-b2-3)\,x-c-b2-2)\,z+x\,(c-b1+1)\big)\,Y_3-(b2+1)\,\big(((-2\,a-b2-2)\,z+(a-b1+1)\,x+c)\,Y_2-Y_1\,a\,b2\big)\Big)\right], \left[Y_1 = f(z), Y_2 = \frac{\mathrm{d}}{\mathrm{d}z}f(z), Y_3 = \frac{\mathrm{d}^2}{\mathrm{d}z^2}f(z)\right], 0, [Y0, DY0, DDY0]\right] \quad \textbf{(2.1.3.3)}$$

Now we want to be able to perform the evaluation for a complex shift and scale, so we can integrate the DE along any line in the complex plane, so apply a change of co-ordinates:

```
> macro(S = z0, M = Delta) :
> DE0 := simplify(factor(isolate(PDEtools[dchange]({z = S + t·M, f(z) = F(t)},
    AppellF1ODE, [t, F(t)]), diff(F(t), t$3))), size);
```

$$DE0 := \frac{\mathrm{d}^3}{\mathrm{d}t^3}F(t) = -\left(2\,\Delta\left(\left(\left(\frac{t^2\,(a+2\,b2+4)\,\Delta^2}{2} - \frac{t\,((-2\,a-4\,b2-8)\,z0+(x+1)\,b2+(a-b1+3)\,x+c+2)\,\Delta}{2} + \left(2+\frac{a}{2}+b2\right)z0^2 + \left(\left(-\frac{1}{2}-\frac{x}{2}\right)b2 + \left(-\frac{3}{2}-\frac{a}{2}+\frac{b1}{2}\right)x - \frac{c}{2} - 1\right)z0 + \frac{x\,(c-b1+1)}{2}\right)\left(\frac{\mathrm{d}^2}{\mathrm{d}t^2}F(t)\right) + (b2+1)\,\Delta\left(\left(t\left(a+\frac{b2}{2}+1\right)\Delta + \left(a+\frac{b2}{2}+1\right)z0 + \left(-\frac{a}{2}+\frac{b1}{2}-\frac{1}{2}\right)x - \frac{c}{2}\right)\left(\frac{\mathrm{d}}{\mathrm{d}t}F(t)\right) + \frac{\Delta\,F(t)\,a\,b2}{2}\right)\right)\right)\right) \Big/ \big((t\,\Delta+z0)\,(t\,\Delta+z0-1)\,(t\,\Delta-x+z0)\big) \quad \textbf{(2.1.3.4)}$$

First optimization step

```
> newvars := {s1 = S + t*M};
```

$$newvars := \{s1 = t\,\Delta + z0\} \quad \textbf{(2.1.3.5)}$$

```
> DE1 := simplify(DE0, {S = s1 - t*M});
```

$$DE1 := \frac{\mathrm{d}^3}{\mathrm{d}t^3}F(t) = -\frac{1}{s1\,(s1-1)\,(-s1+x)}\left(\Delta\left(((-a-2\,b2-4)\,s1^2+((a-b1+b2+3)\,x+c+b2+2)\,s1-x\,(c-b1+1))\left(\frac{\mathrm{d}^2}{\mathrm{d}t^2}F(t)\right) + (b2+1)\,\Delta\left(((-2\,a-b2-2)\,s1+(a-b1+1)\,x+c)\left(\frac{\mathrm{d}}{\mathrm{d}t}F(t)\right)\right.\right.\right. \quad \textbf{(2.1.3.6)}$$

$$- \Delta F(t)\, a\, b2 \Big) \Big) \Big)$$

Next step is performed with this little program

```
> OptimizedSequence := proc(EE)
  local Objects1, j, s, s_in_ee, s_used, S, N, Objects, ee,
  newvars, tmp_newvars, last_N;

  ee := EE;
  s_in_ee := indets(ee, 'suffixed(s)'):
  s_used := {};
  newvars := {};


  "______________________________________________________________
  _______________";
  do
            Objects := remove(u -> u::'`*`' and nops(indets(u,
  'name')) = 1, sort([op(indets(ee, 'Or(`+`, `*`, `^`)'))],
  'length'));
            if Objects = [] then break fi;
            N := nops(Objects);
            S := [`tools/_Qn`('s', [ee, newvars], N)];
            tmp_newvars := Objects[1 .. N] =~ S;
            ee := subs(op(tmp_newvars), ee);
            s_used := indets(ee, 'suffixed'(s)) minus indets
  (newvars, 'suffixed(s)') minus s_in_ee;
            newvars := newvars union map((u, NV) -> u = subs
  (NV, u), s_used, map(rhs = lhs, tmp_newvars));

  "______________________________________________________________
  _______________";
  od;
  last_N := max(map(parse, map(substring, indets(newvars,
  'suffixed(s)'), 2..-1)));
  s_used := select(member, [seq(cat(s, j), j=1..last_N)], map
  (lhs, newvars));
  newvars := s_used =~ subs(newvars, s_used);

  # redefine the sn to match their number

  S := [`tools/_Qn`('s', [EE], nops(newvars))];
  [op(subs(s_used =~ S, [op(newvars), ee]))];
  end:
> OptimizedSequence(DE1)
```

$$\Big[s2 = \frac{1}{s1}, s3 = s1^2, s4 = b2 + 1, s5 = -s1 + x, s6 = s1 - 1, s7 = -2\, a - b2 - 2, \quad \textbf{(2.1.3.7)}$$

$$s8 = -a - 2\, b2 - 4, s9 = a - b1 + 1, s10 = c - b1 + 1, s11 = a - b1 + b2$$

$$+ 3, s12 = -\Delta F(t)\, a\, b2, s13 = \frac{1}{s5}, s14 = \frac{1}{s6}, s15 = s9\, x, s16 = s11\, x, s17$$

$$= s7\, s1, s18 = s8\, s3, s19 = -x\, s10, s20 = c + s15 + s17, s21 = b2 + c + 2$$

$+ s16, s22 = s21\, s1, s23 = s20 \left(\frac{d}{dt} F(t)\right), s24 = s23 + s12, s25 = s18 + s19$

$+ s22, s26 = s4\, \Delta\, s24, s27 = s25 \left(\frac{d^2}{dt^2} F(t)\right), s28 = s27 + s26, s29 =$

$-\Delta\, s28\, s2\, s14\, s13, \frac{d^3}{dt^3} F(t) = s29\Big]$

Incorporate now the first equation not handled via OptimizedSequence

> $OS := [s1 = S + t * M, op((\mathbf{2.1.3.7}))]$

$$OS := \Big[s1 = t\,\Delta + z0, s2 = \frac{1}{s1}, s3 = s1^2, s4 = b2 + 1, s5 = -s1 + x, s6 = s1 - 1, \quad \mathbf{(2.1.3.8)}$$

$$s7 = -2\,a - b2 - 2, s8 = -a - 2\,b2 - 4, s9 = a - b1 + 1, s10 = c - b1 + 1,$$

$$s11 = a - b1 + b2 + 3, s12 = -\Delta\, F(t)\, a\, b2, s13 = \frac{1}{s5}, s14 = \frac{1}{s6}, s15 = s9\, x,$$

$$s16 = s11\, x, s17 = s7\, s1, s18 = s8\, s3, s19 = -x\, s10, s20 = c + s15 + s17, s21$$

$$= b2 + c + 2 + s16, s22 = s21\, s1, s23 = s20 \left(\frac{d}{dt} F(t)\right), s24 = s23 + s12,$$

$$s25 = s18 + s19 + s22, s26 = s4\, \Delta\, s24, s27 = s25 \left(\frac{d^2}{dt^2} F(t)\right), s28 = s27$$

$$+ s26, s29 = -\Delta\, s28\, s2\, s14\, s13, \frac{d^3}{dt^3} F(t) = s29\Big]$$

> $parameters := op(1..-2, F1)$

$$parameters := a, b1, b2, c, x \quad \mathbf{(2.1.3.9)}$$

> $auxiliaryvariables := indets([(\mathbf{2.1.3.7})], symbol)$ **minus** $\{parameters, t\}$ **union** $\{z0\}$

$$auxiliaryvariables := \{\Delta, s1, s10, s11, s12, s13, s14, s15, s16, s17, s18, s19, s2, \quad \mathbf{(2.1.3.10)}$$

$$s20, s21, s22, s23, s24, s25, s26, s27, s28, s29, s3, s4, s5, s6, s7, s8, s9, z0\}$$

> $auxiliaryvariables$ **minus** $indets(auxiliaryvariables, suffixed(s))$

$$\{\Delta, z0\} \quad \mathbf{(2.1.3.11)}$$

> $riauxiliaryfunctions := \{$
> $F(t) = Y[1] + I \cdot Y[5],$
> $diff(F(t), t) = Y[2] + I \cdot Y[6],$
> $diff(F(t), t, t) = Y[3] + I \cdot Y[7],$
> $diff(F(t), t, t, t) = YP[3] + I \cdot YP[7]\}$

$$riauxiliaryfunctions := \Big\{F(t) = Y_1 + I\, Y_5, \frac{d}{dt} F(t) = Y_2 + I\, Y_6, \frac{d^2}{dt^2} F(t) = Y_3 \quad \mathbf{(2.1.3.12)}$$

$$+ \mathrm{I}\, Y_7, \frac{d^3}{dt^3} F(t) = YP_3 + \mathrm{I}\, YP_7 \}$$

> $risymbols := map(u \to u = cat(r, u) + I \cdot cat(i, u), \{parameters\}$ **union** $auxiliaryvariables)$

$$risymbols := \{\Delta = rDelta + \mathrm{I}\, iDelta, a = ra + \mathrm{I}\, ia, b1 = rb1 + \mathrm{I}\, ib1, b2 = rb2 + \mathrm{I}\, ib2, c = rc + \mathrm{I}\, ic, s1 = rs1 + \mathrm{I}\, is1, s10 = rs10 + \mathrm{I}\, is10, s11 = rs11 + \mathrm{I}\, is11, s12 = rs12 + \mathrm{I}\, is12, s13 = rs13 + \mathrm{I}\, is13, s14 = rs14 + \mathrm{I}\, is14, s15 = rs15 + \mathrm{I}\, is15, s16 = rs16 + \mathrm{I}\, is16, s17 = rs17 + \mathrm{I}\, is17, s18 = rs18 + \mathrm{I}\, is18, s19 = rs19 + \mathrm{I}\, is19, s2 = rs2 + \mathrm{I}\, is2, s20 = rs20 + \mathrm{I}\, is20, s21 = rs21 + \mathrm{I}\, is21, s22 = rs22 + \mathrm{I}\, is22, s23 = rs23 + \mathrm{I}\, is23, s24 = rs24 + \mathrm{I}\, is24, s25 = rs25 + \mathrm{I}\, is25, s26 = rs26 + \mathrm{I}\, is26, s27 = rs27 + \mathrm{I}\, is27, s28 = rs28 + \mathrm{I}\, is28, s29 = rs29 + \mathrm{I}\, is29, s3 = rs3 + \mathrm{I}\, is3, s4 = rs4 + \mathrm{I}\, is4, s5 = rs5 + \mathrm{I}\, is5, s6 = rs6 + \mathrm{I}\, is6, s7 = rs7 + \mathrm{I}\, is7, s8 = rs8 + \mathrm{I}\, is8, s9 = rs9 + \mathrm{I}\, is9, x = rx + \mathrm{I}\, ix, z0 = rz0 + \mathrm{I}\, iz0\} \tag{2.1.3.13}$$

> $riOS := subs(risymbols$ **union** $riauxiliaryfunctions, OS)$

$$riOS := [rs1 + \mathrm{I}\, is1 = t\,(rDelta + \mathrm{I}\, iDelta) + rz0 + \mathrm{I}\, iz0, rs2 + \mathrm{I}\, is2 = \frac{1}{rs1 + \mathrm{I}\, is1}, rs3 + \mathrm{I}\, is3 = (rs1 + \mathrm{I}\, is1)^2, rs4 + \mathrm{I}\, is4 = rb2 + \mathrm{I}\, ib2 + 1, rs5 + \mathrm{I}\, is5 = -rs1 - \mathrm{I}\, is1 + rx + \mathrm{I}\, ix, rs6 + \mathrm{I}\, is6 = rs1 + \mathrm{I}\, is1 - 1, rs7 + \mathrm{I}\, is7 = -2\, ra - 2\,\mathrm{I}\, ia - rb2 - \mathrm{I}\, ib2 - 2, rs8 + \mathrm{I}\, is8 = -ra - \mathrm{I}\, ia - 2\, rb2 - 2\,\mathrm{I}\, ib2 - 4, rs9 + \mathrm{I}\, is9 = ra + \mathrm{I}\, ia - rb1 - \mathrm{I}\, ib1 + 1, rs10 + \mathrm{I}\, is10 = rc + \mathrm{I}\, ic - rb1 - \mathrm{I}\, ib1 + 1, rs11 + \mathrm{I}\, is11 = ra + \mathrm{I}\, ia - rb1 - \mathrm{I}\, ib1 + rb2 + \mathrm{I}\, ib2 + 3, rs12 + \mathrm{I}\, is12 = -(rDelta + \mathrm{I}\, iDelta)\,(Y_1 + \mathrm{I}\, Y_5)\,(ra + \mathrm{I}\, ia)\,(rb2 + \mathrm{I}\, ib2), rs13 + \mathrm{I}\, is13 = \frac{1}{rs5 + \mathrm{I}\, is5}, rs14 + \mathrm{I}\, is14 = \frac{1}{rs6 + \mathrm{I}\, is6}, rs15 + \mathrm{I}\, is15 = (rs9 + \mathrm{I}\, is9)\,(rx + \mathrm{I}\, ix), rs16 + \mathrm{I}\, is16 = (rs11 + \mathrm{I}\, is11)\,(rx + \mathrm{I}\, ix), rs17 + \mathrm{I}\, is17 = (rs7 + \mathrm{I}\, is7)\,(rs1 + \mathrm{I}\, is1), rs18 + \mathrm{I}\, is18 = (rs8 + \mathrm{I}\, is8)\,(rs3 + \mathrm{I}\, is3), rs19 + \mathrm{I}\, is19 = -(rx + \mathrm{I}\, ix)\,(rs10 + \mathrm{I}\, is10), rs20 + \mathrm{I}\, is20 = rc + \mathrm{I}\, ic + rs15 + \mathrm{I}\, is15 + rs17 + \mathrm{I}\, is17, rs21 + \mathrm{I}\, is21 = rb2 + \mathrm{I}\, ib2 + rc + \mathrm{I}\, ic + 2 + rs16 + \mathrm{I}\, is16, rs22 \tag{2.1.3.14}$$

$+ \mathrm{I}\, is22 = (rs21 + \mathrm{I}\, is21)\, (rs1 + \mathrm{I}\, is1), rs23 + \mathrm{I}\, is23 = (rs20 + \mathrm{I}\, is20)\, (Y_2 + \mathrm{I}\, Y_6), rs24 + \mathrm{I}\, is24 = rs23 + \mathrm{I}\, is23 + rs12 + \mathrm{I}\, is12, rs25 + \mathrm{I}\, is25 = rs18 + \mathrm{I}\, is18 + rs19 + \mathrm{I}\, is19 + rs22 + \mathrm{I}\, is22, rs26 + \mathrm{I}\, is26 = (rs4 + \mathrm{I}\, is4)\, (rDelta + \mathrm{I}\, iDelta)\, (rs24 + \mathrm{I}\, is24), rs27 + \mathrm{I}\, is27 = (rs25 + \mathrm{I}\, is25)\, (Y_3 + \mathrm{I}\, Y_7), rs28 + \mathrm{I}\, is28 = rs27 + \mathrm{I}\, is27 + rs26 + \mathrm{I}\, is26, rs29 + \mathrm{I}\, is29 = -(rDelta + \mathrm{I}\, iDelta)\, (rs28 + \mathrm{I}\, is28)\, (rs2 + \mathrm{I}\, is2)\, (rs14 + \mathrm{I}\, is14)\, (rs13 + \mathrm{I}\, is13), YP_3 + \mathrm{I}\, YP_7 = rs29 + \mathrm{I}\, is29]$

```
> rseq := NULL :
  for i in riOS do
     rseq := rseq, simplify(map(Re, i)), simplify(map(Im, i)) assuming real;
  end do:
  i := 'i':
  rseq;
```

$$rs1 = rDelta\, t + rz0, is1 = iDelta\, t + iz0, rs2 = \frac{rs1}{is1^2 + rs1^2}, is2 = -\frac{is1}{is1^2 + rs1^2}, rs3 = -is1^2 + rs1^2, is3 = 2\, rs1\, is1, rs4 = rb2 + 1, is4 = ib2, rs5 = -rs1 + rx, is5 = -is1 + ix, rs6 = rs1 - 1, is6 = is1, rs7 = -2\, ra - rb2 - 2, is7 = -2\, ia - ib2, rs8 = -ra - 2\, rb2 - 4, is8 = -ia - 2\, ib2, rs9 = ra - rb1 + 1, is9 = ia - ib1, rs10 = rc - rb1 + 1, is10 = ic - ib1, rs11 = ra - rb1 + rb2 + 3, is11 = ia - ib1 + ib2, rs12 = ((ia\, rb2 + ib2\, ra)\, iDelta + rDelta\, (ia\, ib2 - ra\, rb2))\, Y_1 - Y_5\, ((ia\, ib2 - ra\, rb2)\, iDelta - rDelta\, (ia\, rb2 + ib2\, ra)), is12 = ((ia\, ib2 - ra\, rb2)\, iDelta - rDelta\, (ia\, rb2 + ib2\, ra))\, Y_1 + Y_5\, ((ia\, rb2 + ib2\, ra)\, iDelta + rDelta\, (ia\, ib2 - ra\, rb2)), rs13 = \frac{rs5}{is5^2 + rs5^2}, is13 = -\frac{is5}{is5^2 + rs5^2}, rs14 = \frac{rs6}{is6^2 + rs6^2}, is14 = -\frac{is6}{is6^2 + rs6^2}, rs15 = -is9\, ix + rs9\, rx, is15 = is9\, rx + ix\, rs9, rs16 = -is11\, ix + rs11\, rx, is16 = is11\, rx + ix\, rs11, rs17 = -is1\, is7 + rs1\, rs7, is17 = is1\, rs7 + is7\, rs1, rs18 = -is3\, is8 + rs3\, rs8, is18 = is3\, rs8 + is8\, rs3, rs19 = is10\, ix - rs10\, rx, is19 = -is10\, rx - ix\, rs10, rs20 = rc + rs15 + rs17, is20 = ic + is15 + is17, rs21 = rb2 + rc + 2 + rs16, is21 = ib2 + ic + is16, rs22 = -is1\, is21 + rs1\, rs21, is22 = is1\, rs21 + is21\, rs1, rs23 = -is20\, Y_6 + rs20\, Y_2, is23 = is20\, Y_2 + rs20\, Y_6, rs24 = rs23 + rs12, is24 = is23 + is12, rs25 = rs18 + rs19 + rs22, is25 = is18 + is19 + is22, rs26 = rs4\, (-iDelta\, is24 + rDelta\, rs24) - is4\, (iDelta\, rs24 \tag{2.1.3.15}$$

$+ is24\ rDelta), is26 = rs4\ (iDelta\ rs24 + is24\ rDelta) - (iDelta\ is24 - rDelta\ rs24)\ is4, rs27 = -is25\ Y_7 + rs25\ Y_3, is27 = is25\ Y_3 + rs25\ Y_7,$
$rs28 = rs27 + rs26, is28 = is27 + is26, rs29 = (((-is2\ rs28 - is28\ rs2)\ is14 - rs14\ (is2\ is28 - rs2\ rs28))\ is13 - ((is2\ is28 - rs2\ rs28)\ is14 - rs14\ (is2\ rs28 + is28\ rs2))\ rs13)\ iDelta - (((is2\ is28 - rs2\ rs28)\ is14 - rs14\ (is2\ rs28 + is28\ rs2))\ is13 - ((is2\ rs28 + is28\ rs2)\ is14 + rs14\ (is2\ is28 - rs2\ rs28))\ rs13)\ rDelta, is29 = (((-is2\ is28 + rs2\ rs28)\ is14 + rs14\ (is2\ rs28 + is28\ rs2))\ is13 + ((is2\ rs28 + is28\ rs2)\ is14 + rs14\ (is2\ is28 - rs2\ rs28))\ rs13)\ iDelta + rDelta\ (((is2\ rs28 + is28\ rs2)\ is14 + rs14\ (is2\ is28 - rs2\ rs28))\ is13 + ((is2\ is28 - rs2\ rs28)\ is14 - rs14\ (is2\ rs28 + is28\ rs2))\ rs13), YP_3 = rs29, YP_7 = is29$

Add equations for primed variables

> YP_equations := YP[1] = Y[2], YP[5] = Y[6], YP[2] = Y[3], YP[6] = Y[7]

$$YP\_equations := YP_1 = Y_2, YP_5 = Y_6, YP_2 = Y_3, YP_6 = Y_7 \quad \textbf{(2.1.3.16)}$$

> RIOS := rseq, YP_equations

$$RIOS := rs1 = rDelta\ t + rz0, is1 = iDelta\ t + iz0, rs2 = \frac{rs1}{is1^2 + rs1^2}, is2 = -\frac{is1}{is1^2 + rs1^2}, rs3 = -is1^2 + rs1^2, is3 = 2\ rs1\ is1, rs4 = rb2 + 1, is4 = ib2, rs5 = -rs1 + rx, is5 = -is1 + ix, rs6 = rs1 - 1, is6 = is1, rs7 = -2\ ra - rb2 - 2, is7 = -2\ ia - ib2, rs8 = -ra - 2\ rb2 - 4, is8 = -ia - 2\ ib2, rs9 = ra - rb1 + 1, is9 = ia - ib1, rs10 = rc - rb1 + 1, is10 = ic - ib1, rs11 = ra - rb1 + rb2 + 3, is11 = ia - ib1 + ib2, rs12 = ((ia\ rb2 + ib2\ ra)\ iDelta + rDelta\ (ia\ ib2 - ra\ rb2))\ Y_1 - Y_5\ ((ia\ ib2 - ra\ rb2)\ iDelta - rDelta\ (ia\ rb2 + ib2\ ra)), is12 = ((ia\ ib2 - ra\ rb2)\ iDelta - rDelta\ (ia\ rb2 + ib2\ ra))\ Y_1 + Y_5\ ((ia\ rb2 + ib2\ ra)\ iDelta + rDelta\ (ia\ ib2 - ra\ rb2)), rs13 = \frac{rs5}{is5^2 + rs5^2}, is13 = -\frac{is5}{is5^2 + rs5^2}, rs14 = \frac{rs6}{is6^2 + rs6^2}, is14 = -\frac{is6}{is6^2 + rs6^2}, rs15 = -is9\ ix + rs9\ rx, is15 = is9\ rx + ix\ rs9, rs16 = -is11\ ix + rs11\ rx, is16 = is11\ rx + ix\ rs11, rs17 = -is1\ is7 + rs1\ rs7, is17 = is1\ rs7 + is7\ rs1, rs18 = -is3\ is8 + rs3\ rs8, is18 = is3\ rs8 + is8\ rs3, rs19 = is10\ ix - rs10\ rx, is19 = -is10\ rx - ix\ rs10, rs20 = rc + rs15 + rs17, is20 = ic + is15 + is17, rs21 = rb2 + rc + 2 + rs16, is21 = ib2 + ic + is16, rs22 = -is1\ is21 + rs1\ rs21, is22 = is1\ rs21 \quad \textbf{(2.1.3.17)}$$

$+ is21\ rs1, rs23 = -is20\ Y_6 + rs20\ Y_2, is23 = is20\ Y_2 + rs20\ Y_6, rs24 = rs23$
$+ rs12, is24 = is23 + is12, rs25 = rs18 + rs19 + rs22, is25 = is18 + is19$
$+ is22, rs26 = rs4\ (-iDelta\ is24 + rDelta\ rs24) - is4\ (iDelta\ rs24$
$+ is24\ rDelta), is26 = rs4\ (iDelta\ rs24 + is24\ rDelta) - (iDelta\ is24$
$- rDelta\ rs24)\ is4, rs27 = -is25\ Y_7 + rs25\ Y_3, is27 = is25\ Y_3 + rs25\ Y_7,$
$rs28 = rs27 + rs26, is28 = is27 + is26, rs29 = (((-is2\ rs28 - is28\ rs2)\ is14$
$- rs14\ (is2\ is28 - rs2\ rs28))\ is13 - ((is2\ is28 - rs2\ rs28)\ is14$
$- rs14\ (is2\ rs28 + is28\ rs2))\ rs13)\ iDelta - (((is2\ is28 - rs2\ rs28)\ is14$
$- rs14\ (is2\ rs28 + is28\ rs2))\ is13 - ((is2\ rs28 + is28\ rs2)\ is14$
$+ rs14\ (is2\ is28 - rs2\ rs28))\ rs13)\ rDelta, is29 = (((-is2\ is28$
$+ rs2\ rs28)\ is14 + rs14\ (is2\ rs28 + is28\ rs2))\ is13 + ((is2\ rs28$
$+ is28\ rs2)\ is14 + rs14\ (is2\ is28 - rs2\ rs28))\ rs13)\ iDelta$
$+ rDelta\ (((is2\ rs28 + is28\ rs2)\ is14 + rs14\ (is2\ is28 - rs2\ rs28))\ is13$
$+ ((is2\ is28 - rs2\ rs28)\ is14 - rs14\ (is2\ rs28 + is28\ rs2))\ rs13), YP_3$
$= rs29, YP_7 = is29, YP_1 = Y_2, YP_5 = Y_6, YP_2 = Y_3, YP_6 = Y_7$

Next there is a procedure, RIOSToProc, that takes the RIOS and returns the procedure used by dsolve/numeric to numerically evaluate the ODE behind the RIOS (in this case, the AppellF1 function)

```
> RIOSToProc := proc(RIOS)
  local previous_warnlevel;
      try
          previous_warnlevel := interface(warnlevel = 0):
          return(parse(sprintf("proc(N,t,Y,YP) %A end", cat(op
  (map(u -> sprintf("%a := %a;", op(u)), RIOS))))))
      finally:
          interface(warnlevel = previous_warnlevel)
      end:
  end:
```

Create dsolve's procedure from the Real and Imaginary parts of the Optimized Sequence

```
> dproc := RIOSToProc([RIOS])
```

$dproc := \textbf{proc}(N, t, Y, YP)$ **(2.1.3.18)**

local $rs1, is1, rs2, is2, rs3, is3, rs4, is4, rs5, is5, rs6, is6, rs7, is7, rs8, is8,$
$rs9, is9, rs10, is10, rs11, is11, rs12, is12, rs13, is13, rs14, is14, rs15, is15,$
$rs16, is16, rs17, is17, rs18, is18, rs19, is19, rs20, is20, rs21, is21, rs22, is22,$
$rs23, is23, rs24, is24, rs25, is25, rs26, is26, rs27, is27, rs28, is28, rs29, is29;$

$rs1 := rDelta * t + rz0;$

$is1 := iDelta * t + iz0;$

$rs2 := rs1 / (is1\verb|^|2 + rs1\verb|^|2);$

$is2 := -is1 / (is1\verb|^|2 + rs1\verb|^|2);$

$rs3 := -is1\verb|^|2 + rs1\verb|^|2;$

$is3 := 2 * is1 * rs1;$

```
rs4 := rb2 + 1;
is4 := ib2;
rs5 := -rs1 + rx;
is5 := -is1 + ix;
rs6 := rs1 - 1;
is6 := is1;
rs7 := -2*ra - rb2 - 2;
is7 := -2*ia - ib2;
rs8 := -ra - 2*rb2 - 4;
is8 := -ia - 2*ib2;
rs9 := ra - rb1 + 1;
is9 := ia - ib1;
rs10 := rc - rb1 + 1;
is10 := ic - ib1;
rs11 := ra - rb1 + rb2 + 3;
is11 := ia - ib1 + ib2;
rs12 := ((ia*rb2 + ib2*ra)*iDelta + rDelta*(ia*ib2 - ra*rb2))*Y
[1] - Y[5]*((ia*ib2 - ra*rb2)*iDelta - rDelta*(ia*rb2 + ib2
*ra));
is12 := ((ia*ib2 - ra*rb2)*iDelta - rDelta*(ia*rb2 + ib2*ra))*Y
[1] + Y[5]*((ia*rb2 + ib2*ra)*iDelta + rDelta*(ia*ib2 - ra*rb2));
rs13 := rs5/(is5^2 + rs5^2);
is13 := -is5/(is5^2 + rs5^2);
rs14 := rs6/(is6^2 + rs6^2);
is14 := -is6/(is6^2 + rs6^2);
rs15 := -is9*ix + rs9*rx;
is15 := is9*rx + ix*rs9;
rs16 := -is11*ix + rs11*rx;
is16 := is11*rx + ix*rs11;
rs17 := -is1*is7 + rs1*rs7;
is17 := is1*rs7 + is7*rs1;
rs18 := -is3*is8 + rs3*rs8;
is18 := is3*rs8 + is8*rs3;
rs19 := is10*ix - rs10*rx;
is19 := -is10*rx - ix*rs10;
rs20 := rc + rs15 + rs17;
is20 := ic + is15 + is17;
rs21 := rb2 + rc + 2 + rs16;
```

```
is21 := ib2 + ic + is16;
rs22 := - is1 * is21 + rs1 * rs21;
is22 := is1 * rs21 + is21 * rs1;
rs23 := - is20 * Y[6] + rs20 * Y[2];
is23 := is20 * Y[2] + rs20 * Y[6];
rs24 := rs23 + rs12;
is24 := is23 + is12;
rs25 := rs18 + rs19 + rs22;
is25 := is18 + is19 + is22;
rs26 := rs4 * ( - iDelta * is24 + rDelta * rs24) - is4 * (iDelta * rs24 + is24
 * rDelta);
is26 := rs4 * (iDelta * rs24 + is24 * rDelta) - (iDelta * is24 - rDelta
 * rs24) * is4;
rs27 := - is25 * Y[7] + rs25 * Y[3];
is27 := is25 * Y[3] + rs25 * Y[7];
rs28 := rs27 + rs26;
is28 := is27 + is26;
rs29 := ((( - is2 * rs28 - is28 * rs2) * is14 - rs14 * (is2 * is28 - rs2
 * rs28)) * is13 - ((is2 * is28 - rs2 * rs28) * is14 - rs14 * (is2 * rs28
 + is28 * rs2)) * rs13) * iDelta - ((((is2 * is28 - rs2 * rs28) * is14 - rs14
 * (is2 * rs28 + is28 * rs2)) * is13 - ((is2 * rs28 + is28 * rs2) * is14 + rs14
 * (is2 * is28 - rs2 * rs28)) * rs13) * rDelta;
is29 := ((( - is2 * is28 + rs2 * rs28) * is14 + rs14 * (is2 * rs28 + is28
 * rs2)) * is13 + ((is2 * rs28 + is28 * rs2) * is14 + rs14 * (is2 * is28 - rs2
 * rs28)) * rs13) * iDelta + rDelta * ((((is2 * rs28 + is28 * rs2) * is14 + rs14
 * (is2 * is28 - rs2 * rs28)) * is13 + ((is2 * is28 - rs2 * rs28) * is14 - rs14
 * (is2 * rs28 + is28 * rs2)) * rs13);
YP[3] := rs29;
YP[7] := is29;
YP[1] := Y[2];
YP[5] := Y[6];
YP[2] := Y[3];
YP[6] := Y[7]
end
>
```

Example 1: $F_1\left(1, 2, \frac{1}{2}, 1 + \mathrm{I}, \frac{5}{6}, \frac{3}{4} + \frac{\mathrm{I}}{20}\right)$

Example:

1) Pick values for $a, b1, b2, c, x$
2) Choose a start point and direction
3) Evaluate the relevant routine, and use dsolve/numeric

> $F1$

$$F_1(a, b1, b2, c, x, z) \tag{2.1.3.1.1}$$

1) Pick values for $a, b1, b2, c, x$

> $z\_val := \frac{3}{4} + \frac{I}{20}$

$$z\_val := \frac{3}{4} + \frac{\mathrm{I}}{20} \tag{2.1.3.1.2}$$

> $evalf(\mathrm{abs}(z\_val))$

$$0.7516648190 \tag{2.1.3.1.3}$$

For the purpose of this example, choose a z1 such that $|z2| < |z1|$

> $AF := \%AppellF1\left(1, 2, \frac{1}{2}, 1 + I, \frac{5}{6}, z\_val\right)$

$$AF := F_1\left(1, 2, \frac{1}{2}, 1 + \mathrm{I}, \frac{5}{6}, \frac{3}{4} + \frac{\mathrm{I}}{20}\right) \tag{2.1.3.1.4}$$

> $infolevel[AppellF1] := 5$

$$infolevel_{AppellF1} := 5 \tag{2.1.3.1.5}$$

dsolve/numeric works best with this value of Digits

> $Digits := \mathrm{trunc}(evalhf(Digits))$

$$Digits := 15 \tag{2.1.3.1.6}$$

> $AFval := evalf(AF)$

```
-> Numerical evaluation of AppellF1(1., 2.,
.500000000000000, 1.+1.*I, .833333333333333,
.750000000000000+.500000000000000e-1*I)
   -> AppellF1, checking singular cases
   -> AppellF1, trying special values
   -> AppellF1, trying series based on recurrence
```

$$AFval := -23.2785809735956 - 19.6532598979361\,\mathrm{I} \tag{2.1.3.1.7}$$

The parameters of F1, and their real (rP) and imaginary (iP) parts

> $P := [op(1..-2, F1)] =\sim [op(1..-2, AF)]$

$$P := \left[a = 1, b1 = 2, b2 = \frac{1}{2}, c = 1 + \mathrm{I}, x = \frac{5}{6}\right] \tag{2.1.3.1.8}$$

> $rP := map(u \rightarrow cat(r, lhs(u)) = \mathrm{Re}(rhs(u)), P)$

$$rP := \left[ra = 1, rb1 = 2, rb2 = \frac{1}{2}, rc = 1, rx = \frac{5}{6}\right] \tag{2.1.3.1.9}$$

> $iP := map(u \rightarrow cat(i, lhs(u)) = \mathrm{Im}(rhs(u)), P)$

$$iP := [ia = 0, ib1 = 0, ib2 = 0, ic = 1, ix = 0] \tag{2.1.3.1.10}$$

2) Choose a start point and direction. We aim at computing F1 at z = z2 = z_val = $\frac{3}{4} + \frac{\mathrm{I}}{20}$.

We start with deriving formulas for z0 and Delta and choose convenient values when possible (there is some arbitrariness in the choice of Delta and z0)

We have:

> $Z := z = t\,\Delta + z0$

$$Z := z = t\,\Delta + z0 \tag{2.1.3.1.11}$$

> $subs(risymbols, Z)$

$$z = t\,(rDelta + \mathrm{I}\,iDelta) + rz0 + \mathrm{I}\,iz0 \tag{2.1.3.1.12}$$

> $\mathrm{Re}(z) = (\mathrm{Re}(rhs(\%))$ assuming $real)$

$$\Re(z) = rDelta\,t + rz0 \tag{2.1.3.1.13}$$

> $\mathrm{Im}(z) = (\mathrm{Im}(rhs(\%\%))$ assuming $real)$

$$\Im(z) = iDelta\,t + iz0 \tag{2.1.3.1.14}$$

> $solve(\{\mathbf{(2.1.3.1.13)}, \mathbf{(2.1.3.1.14)}\}, \{t, iDelta\})$

$$\left\{iDelta = \frac{rDelta\,(\Im(z) - iz0)}{\Re(z) - rz0}, t = \frac{\Re(z) - rz0}{rDelta}\right\} \tag{2.1.3.1.15}$$

Then, iDelta is different from zero only if $\Im(z) - iz0 \neq 0$, the real variable t is given by $t = \frac{\Re(z) - rz0}{rDelta}$, and all of rDelta, rz0 and iz0 are arbitrary, which amounts to say that

- *"The initial point z0 could be anyone provided it is not a singularity, and the value of rDelta can be chosen at will - note however that this value will be taken into account when computing initial values at z0 for dsolve's procedure"*

Although any allowed value for z0 and Delta are OK and lead to the correct value of F1, if the Im(z) is small (within the radius of convergence of the F1 series) we can always choose iz0 = Im(z) so that iDelta = 0, then choose rDelta = 1 (or as big as possible?), and rz0 as big as possible (ay 3/4 of the radius of convergence) so that the computation of z0 is easy and the numerical integration over t is along a smaller path. In this example the radius of convergence around the origin is abs(z1) = 5/6, so

> $z0\_val := \frac{3}{4} \cdot \frac{5}{6} + I \cdot \mathrm{Im}(z\_val)$

$$z0\_val := \frac{5}{8} + \frac{\mathrm{I}}{20} \tag{2.1.3.1.16}$$

> $z0\_Delta := \{rDelta = 1, iz0 = \mathrm{Im}(z0\_val), rz0 = \mathrm{Re}(z0\_val)\}$

$$z0\_Delta := \left\{iz0 = \frac{1}{20}, rDelta = 1, rz0 = \frac{5}{8}\right\} \tag{2.1.3.1.17}$$

> $eval(\mathbf{(2.1.3.1.15)}, z0\_Delta\ \mathbf{union}\ \{z = z\_val\})$

$$\left\{iDelta = 0, t = \frac{1}{8}\right\} \tag{2.1.3.1.18}$$

> $z0\_Delta := z0\_Delta\ \mathbf{union}\ select(has, \mathbf{(2.1.3.1.18)}, iDelta)$

$$z0\_Delta := \left\{iDelta = 0, iz0 = \frac{1}{20}, rDelta = 1, rz0 = \frac{5}{8}\right\} \tag{2.1.3.1.19}$$

> $Delta\_val := subs(z0\_Delta, rDelta + I \cdot iDelta)$

(2.1.3.1.20)

$$Delta\_val := 1 \quad (2.1.3.1.20)$$

> t_val := select(has, (2.1.3.1.18), t)[1]

$$t\_val := t = \frac{1}{8} \quad (2.1.3.1.21)$$

> evalf(%)

$$t = 0.125000000000000 \quad (2.1.3.1.22)$$

So, the value of z0 is 1/10, the evaluating point is z = 3/4, that implies on integrating the t variable from t = 0 to t = 13/20, and the values of rz0, iz0, rDelta, iDelta are, respectively

The procedure to use with dsolve/numeric is the one we get from dproc by substituting the values of all the relevant variables rP, iP and *z0_Delta*

> thisproc := subs(rP, iP, z0_Delta, eval(dproc))

thisproc := **proc**(N, t, Y, YP) **(2.1.3.1.23)**

```
local rs1, is1, rs2, is2, rs3, is3, rs4, is4, rs5, is5, rs6, is6, rs7, is7, rs8,
is8, rs9, is9, rs10, is10, rs11, is11, rs12, is12, rs13, is13, rs14, is14,
rs15, is15, rs16, is16, rs17, is17, rs18, is18, rs19, is19, rs20, is20, rs21,
is21, rs22, is22, rs23, is23, rs24, is24, rs25, is25, rs26, is26, rs27, is27,
rs28, is28, rs29, is29;
rs1 := t + 5/8;
is1 := 1/20;
rs2 := rs1/(is1^2 + rs1^2);
is2 := − is1/(is1^2 + rs1^2);
rs3 := − is1^2 + rs1^2;
is3 := 2*is1*rs1;
rs4 := 3/2;
is4 := 0;
rs5 := − rs1 + 5/6;
is5 := − is1;
rs6 := rs1 − 1;
is6 := is1;
rs7 := − 9/2;
is7 := 0;
rs8 := − 6;
is8 := 0;
rs9 := 0;
is9 := 0;
rs10 := 0;
is10 := 1;
rs11 := 5/2;
is11 := 0;
```

```
rs12 := -1/2*Y[1];
is12 := -1/2*Y[5];
rs13 := rs5/(is5^2 + rs5^2);
is13 := -is5/(is5^2 + rs5^2);
rs14 := rs6/(is6^2 + rs6^2);
is14 := -is6/(is6^2 + rs6^2);
rs15 := 5/6*rs9;
is15 := 5/6*is9;
rs16 := 5/6*rs11;
is16 := 5/6*is11;
rs17 := -is1*is7 + rs1*rs7;
is17 := is1*rs7 + is7*rs1;
rs18 := -is3*is8 + rs3*rs8;
is18 := is3*rs8 + is8*rs3;
rs19 := -5/6*rs10;
is19 := -5/6*is10;
rs20 := 1 + rs15 + rs17;
is20 := 1 + is15 + is17;
rs21 := 7/2 + rs16;
is21 := 1 + is16;
rs22 := -is1*is21 + rs1*rs21;
is22 := is1*rs21 + is21*rs1;
rs23 := -is20*Y[6] + rs20*Y[2];
is23 := is20*Y[2] + rs20*Y[6];
rs24 := rs23 + rs12;
is24 := is23 + is12;
rs25 := rs18 + rs19 + rs22;
is25 := is18 + is19 + is22;
rs26 := rs4*rs24 - is4*is24;
is26 := rs4*is24 + rs24*is4;
rs27 := -is25*Y[7] + rs25*Y[3];
is27 := is25*Y[3] + rs25*Y[7];
rs28 := rs27 + rs26;
is28 := is27 + is26;
rs29 := -((is2*is28 - rs2*rs28)*is14 - rs14*(is2*rs28 + is28
 *rs2))*is13 + ((is2*rs28 + is28*rs2)*is14 + rs14*(is2
 *is28 - rs2*rs28))*rs13;
is29 := ((is2*rs28 + is28*rs2)*is14 + rs14*(is2*is28 - rs2
```

```
*rs28))*is13 + ((is2*is28 - rs2*rs28)*is14 - rs14*(is2*rs28
+ is28*rs2))*rs13;
YP[3] := rs29;
YP[7] := is29;
YP[1] := Y[2];
YP[5] := Y[6];
YP[2] := Y[3];
YP[6] := Y[7]
end
```

Now we need the values of F1 and its 1st and 2nd derivatives at z0, to be used as starting point by dsolve. Note however that, after changing variables z = S+Mt, we have

```
> [f(z), diff(f(z), z), diff(f(z), z, z)] : % =~ PDEtools[dchange]({z = S + t*M,
   f(z) = F(t)}, %, [t, F(t)], known = all)
```

$$\left[f(z) = F(t), \frac{d}{dz} f(z) = \frac{\frac{d}{dt} F(t)}{\Delta}, \frac{d^2}{dz^2} f(z) = \frac{\frac{d^2}{dt^2} F(t)}{\Delta^2} \right] \tag{2.1.3.1.24}$$

where on the lhs f(z) represent $F_1(a, b1, b2, c, x, z)$ and on the rhs F(t) represents $F_1(a, b1, b2, c, x, t)$ so to compute $[F(t), \dot{F}(t), \ddot{F}(t)]$ we compute

$$> convert\left(solve\left(convert\left(\left[f(z) = F(t), \frac{d}{dz} f(z) = \frac{\dot{F}(t)}{\Delta}, \frac{d^2}{dz^2} f(z) = \frac{\ddot{F}(t)}{\Delta^2} \right], D\right), convert([F(t), \dot{F}(t), \ddot{F}(t)], D)\right), diff\right)_1$$

$$\left[F(t) = f(z), \frac{d}{dt} F(t) = \left(\frac{d}{dz} f(z)\right)\Delta, \frac{d^2}{dt^2} F(t) = \left(\frac{d^2}{dz^2} f(z)\right)\Delta^2 \right] \tag{2.1.3.1.25}$$

In this particular example, Delta = 1 so this observation is not relevant.
We then have

$$> DF := subs\left(z = z0, \text{Delta} = Delta\_val, \left[F1, \text{Delta}\cdot\text{'}diff\text{'}(F1, z), \Delta^2\cdot\text{'}diff\text{'}(F1, z, z)\right]\right)$$

$$DF := \left[F_1(a, b1, b2, c, x, z0), \frac{d}{dz0} F_1(a, b1, b2, c, x, z0), \frac{d^2}{dz0^2} F_1(a, b1, b2, c, x, z0) \right] \tag{2.1.3.1.26}$$

```
> subs(z0 = z0_val, eval(DF, P))
```

$$\left[F_1\left(1, 2, \frac{1}{2}, 1 + I, \frac{5}{6}, \frac{5}{8} + \frac{I}{20}\right), \left(\frac{1}{4} - \frac{I}{4}\right) F_1\left(2, 2, \frac{3}{2}, 2 + I, \frac{5}{6}, \frac{5}{8} + \frac{I}{20}\right), \left(\frac{3}{20} - \frac{9\,I}{20}\right) F_1\left(3, 2, \frac{5}{2}, 3 + I, \frac{5}{6}, \frac{5}{8} + \frac{I}{20}\right) \right] \tag{2.1.3.1.27}$$

```
> DFval := evalf((2.1.3.1.27))
-> Numerical evaluation of AppellF1(1., 2.,
```

```
.500000000000000, 1.+1.*I, .833333333333333,
.625000000000000+.500000000000000e-1*I)
   -> AppellF1, checking singular cases
   -> AppellF1, trying special values
   -> AppellF1, trying series based on recurrence
-> Numerical evaluation of AppellF1(2., 2.,
1.50000000000000, 2.+1.*I, .833333333333333,
.625000000000000+.500000000000000e-1*I)
   -> AppellF1, checking singular cases
   -> AppellF1, trying special values
   -> AppellF1, trying series based on recurrence
-> Numerical evaluation of AppellF1(3., 2.,
2.50000000000000, 3.+1.*I, .833333333333333,
.625000000000000+.500000000000000e-1*I)
   -> AppellF1, checking singular cases
   -> AppellF1, trying special values
   -> AppellF1, trying series based on recurrence
```

$DFval := [-17.9388701408564 - 16.3439832562898\,I, -30.6892254450648 - 20.1904162659237\,I, -135.941143719457 - 73.3897112911057\,I]$ **(2.1.3.1.28)**

> $ini := Array([Re(DFval[1]), Re(DFval[2]), Re(DFval[3]), 0, Im(DFval[1]), Im(DFval[2]), Im(DFval[3]), 0], datatype = float[8]);$

$ini := [-17.9388701408564, -30.6892254450648, -135.941143719457, 0., -16.3439832562898, -20.1904162659237, -73.3897112911057, 0.]$ **(2.1.3.1.29)**

> $dsol := dsolve(numeric, procedure = thisproc, initial = ini, start = 0, procvars = [rF1(t), rDF1(t), rDDF1(t), rdummy(t), iF1(t), iDF1(t), iDDF1(t), idummy(t)]);$

$dsol :=$ **proc**(x_rkf45) ... **end** **(2.1.3.1.30)**

> $dsol(subs(t\_val, t))[[2, 6]]$

$[rF1(t) = -23.2785805599301, iF1(t) = -19.6532595475860]$ **(2.1.3.1.31)**

> $SFloat(rhs(dsol(subs(t\_val, t))[2] + I \cdot dsol(subs(t\_val, t))[6]))$

$-23.2785805599301163 - 19.6532595475859644\,I$ **(2.1.3.1.32)**

> $AFval$

$-23.2785809735956 - 19.6532598979361\,I$ **(2.1.3.1.33)**

> $evalf[10](\%\% - \%)$

$4.1\ 10^{-7} + 3.5\ 10^{-7}\,I$ **(2.1.3.1.34)**

Example 2: $F_1\left(1, 2 + I, \frac{3}{2} - \frac{I}{2}, 1, \frac{18}{25}, \frac{19}{20}\right)$

For the purpose of this example, choose a z1 such that $|z1| < |z2|$ and to integrate till z2 we need to "go over z1", which if course is not possible and we overcome the difficulty by using a z0 such that Im(z0) > 0, so we arrive at z2 without going over z1

> $convert([1, 2 + I, 3/2 - I*(1/2), 1, .72, .95], rational)$

$\left[1, 2 + I, \frac{3}{2} - \frac{I}{2}, 1, \frac{18}{25}, \frac{19}{20}\right]$ **(2.1.3.2.1)**

The F1 to be evaluated is actually the last example in evalf/tst/AppellF1_Table2.tst.

> $AF := \%AppellF1(op(\%))$

$$AF := F_1\left(1, 2 + \mathrm{I}, \frac{3}{2} - \frac{\mathrm{I}}{2}, 1, \frac{18}{25}, \frac{19}{20}\right) \quad \textbf{(2.1.3.2.2)}$$

> $z\_val := op(-1, AF)$

$$z\_val := \frac{19}{20} \quad \textbf{(2.1.3.2.3)}$$

> $evalf(\mathrm{abs}(z\_val))$

$$0.950000000000000 \quad \textbf{(2.1.3.2.4)}$$

Compare with abs(z1)

> $evalf(\mathrm{abs}(op(-2, AF)))$

$$0.720000000000000 \quad \textbf{(2.1.3.2.5)}$$

> $infolevel[AppellF1] := 5$

$$infolevel_{AppellF1} := 5 \quad \textbf{(2.1.3.2.6)}$$

> $Digits := \mathrm{trunc}(evalhf(Digits))$

$$Digits := 15 \quad \textbf{(2.1.3.2.7)}$$

The series code resolves the issue swapping z2 and z2

> $AFval := evalf(AF)$

```
case: abs(z1) < abs(z2) < 1; swapping parameters and
z1 <-> z2
-> Numerical evaluation of AppellF1(1.,
1.50000000000000-.500000000000000*I, 2.+1.*I, 1.,
.950000000000000, .720000000000000)
   -> AppellF1, checking singular cases
   -> AppellF1, trying special values
   <- special values of AppellF1 successful
```

$$AFval := 1112.12017081988 - 254.420420227346\,\mathrm{I} \quad \textbf{(2.1.3.2.8)}$$

Go now with dsolve/numeric without swapping. The parameters of F1, and their real (rP) and imaginary (iP) parts

> $F1$

$$F_1(a, b1, b2, c, x, z) \quad \textbf{(2.1.3.2.9)}$$

> $P := [op(1..-2, F1)] =\sim [op(1..-2, AF)]$

$$P := \left[a = 1, b1 = 2 + \mathrm{I}, b2 = \frac{3}{2} - \frac{\mathrm{I}}{2}, c = 1, x = \frac{18}{25}\right] \quad \textbf{(2.1.3.2.10)}$$

> $rP := map(u \rightarrow cat(r, lhs(u)) = \mathrm{Re}(rhs(u)), P)$

$$rP := \left[ra = 1, rb1 = 2, rb2 = \frac{3}{2}, rc = 1, rx = \frac{18}{25}\right] \quad \textbf{(2.1.3.2.11)}$$

> $iP := map(u \rightarrow cat(i, lhs(u)) = \mathrm{Im}(rhs(u)), P)$

$$iP := \left[ia = 0, ib1 = 1, ib2 = -\frac{1}{2}, ic = 0, ix = 0\right] \quad \textbf{(2.1.3.2.12)}$$

2) Choose a start point and direction. We aim at computing F1 at z = z2 = z_val = $\frac{3}{4} + \frac{\mathrm{I}}{20}$.

We start with deriving formulas for z0 and Delta and choose convenient values when possible (there is some arbitrariness in the choice of Delta and z0)

We have:

> $Z := z = t\,\Delta + z0$

$$Z := z = t\,\Delta + z0 \qquad \textbf{(2.1.3.2.13)}$$

> $subs(risymbols, Z)$

$$z = t\,(rDelta + \mathrm{I}\; iDelta) + rz0 + \mathrm{I}\; iz0 \qquad \textbf{(2.1.3.2.14)}$$

> $\mathrm{Re}(z) = (\mathrm{Re}(rhs(\%))$ assuming $real)$

$$\Re(z) = rDelta\; t + rz0 \qquad \textbf{(2.1.3.2.15)}$$

> $\mathrm{Im}(z) = (\mathrm{Im}(rhs(\%\%))$ assuming $real)$

$$\Im(z) = iDelta\; t + iz0 \qquad \textbf{(2.1.3.2.16)}$$

> $solve(\{\textbf{(2.1.3.2.15)}, \textbf{(2.1.3.2.16)}\}, \{t, iDelta\})$

$$\left\{iDelta = \frac{rDelta\,(\Im(z) - iz0)}{\Re(z) - rz0}, t = \frac{\Re(z) - rz0}{rDelta}\right\} \qquad \textbf{(2.1.3.2.17)}$$

Then, iDelta is different from zero only if $\Im(z) - iz0 \neq 0$, the real variable t is given by $t = \frac{\Re(z) - rz0}{rDelta}$, and all of rDelta, rz0 and iz0 are arbitrary, which amounts to say that

- *"The initial point z0 could be anyone provided it is not a singularity, and the value of rDelta can be chosen at will - note however that this value will be taken into account when computing initial values at z0 for dsolve's procedure"*

So take z0 within 1/2 of the radius of convergence, with imaginary part

> $z0\_val := \frac{1}{2}\cdot\min(\mathrm{abs}(op(-2, AF)), \mathrm{abs}(op(-1, AF))) + I\cdot\frac{1}{2}\cdot\min(\mathrm{abs}(op(-2, AF)), \mathrm{abs}(op(-1, AF)))$

$$z0\_val := \frac{9}{25} + \frac{9\,\mathrm{I}}{25} \qquad \textbf{(2.1.3.2.18)}$$

> $z0\_Delta := \{rDelta = 1, iz0 = \mathrm{Im}(z0\_val), rz0 = \mathrm{Re}(z0\_val)\}$

$$z0\_Delta := \left\{iz0 = \frac{9}{25}, rDelta = 1, rz0 = \frac{9}{25}\right\} \qquad \textbf{(2.1.3.2.19)}$$

> $eval(\textbf{(2.1.3.2.17)}, z0\_Delta\ \textbf{union}\ \{z = z\_val\})$

$$\left\{iDelta = -\frac{36}{59}, t = \frac{59}{100}\right\} \qquad \textbf{(2.1.3.2.20)}$$

> $z0\_Delta := z0\_Delta\ \textbf{union}\ select(has, \textbf{(2.1.3.2.20)}, iDelta)$

$$z0\_Delta := \left\{iDelta = -\frac{36}{59}, iz0 = \frac{9}{25}, rDelta = 1, rz0 = \frac{9}{25}\right\} \qquad \textbf{(2.1.3.2.21)}$$

> $Delta\_val := subs(z0\_Delta, rDelta + I\cdot iDelta)$

$$Delta\_val := 1 - \frac{36\,\mathrm{I}}{59} \qquad \textbf{(2.1.3.2.22)}$$

> $t\_val := select(has, \textbf{(2.1.3.2.20)}, t)[1]$

$$t\_val := t = \frac{59}{100} \qquad \textbf{(2.1.3.2.23)}$$

> $evalf(\%)$

(2.1.3.2.24)

$t = 0.590000000000000$ **(2.1.3.2.24)**

So, the value of z0 is 1/10, the evaluating point is z = 3/4, that implies on integrating the t variable from t = 0 to t = 13/20, and the values of rz0, iz0, rDelta, iDelta are, respectively

The procedure to use with dsolve/numeric is the one we get from dproc by substituting the values of all the relevant variables rP, iP and *z0_Delta*

```
> thisproc := subs(rP, iP, z0_Delta, eval(dproc))
thisproc := proc(N, t, Y, YP)                                    (2.1.3.2.25)
    local rs1, is1, rs2, is2, rs3, is3, rs4, is4, rs5, is5, rs6, is6, rs7, is7, rs8,
    is8, rs9, is9, rs10, is10, rs11, is11, rs12, is12, rs13, is13, rs14, is14,
    rs15, is15, rs16, is16, rs17, is17, rs18, is18, rs19, is19, rs20, is20, rs21,
    is21, rs22, is22, rs23, is23, rs24, is24, rs25, is25, rs26, is26, rs27, is27,
    rs28, is28, rs29, is29;
    rs1 := t + 9/25;
    is1 := - 36/59*t + 9/25;
    rs2 := rs1/(is1^2 + rs1^2);
    is2 := - is1/(is1^2 + rs1^2);
    rs3 := - is1^2 + rs1^2;
    is3 := 2*is1*rs1;
    rs4 := 5/2;
    is4 := - 1/2;
    rs5 := - rs1 + 18/25;
    is5 := - is1;
    rs6 := rs1 - 1;
    is6 := is1;
    rs7 := - 11/2;
    is7 := 1/2;
    rs8 := - 8;
    is8 := 1;
    rs9 := 0;
    is9 := - 1;
    rs10 := 0;
    is10 := - 1;
    rs11 := 7/2;
    is11 := - 3/2;
    rs12 := - 141/118*Y[1] - 167/118*Y[5];
    is12 := 167/118*Y[1] - 141/118*Y[5];
    rs13 := rs5/(is5^2 + rs5^2);
    is13 := - is5/(is5^2 + rs5^2);
    rs14 := rs6/(is6^2 + rs6^2);
```

```
is14 := -is6/(is6^2 + rs6^2);
rs15 := 18/25*rs9;
is15 := 18/25*is9;
rs16 := 18/25*rs11;
is16 := 18/25*is11;
rs17 := -is1*is7 + rs1*rs7;
is17 := is1*rs7 + is7*rs1;
rs18 := -is3*is8 + rs3*rs8;
is18 := is3*rs8 + is8*rs3;
rs19 := -18/25*rs10;
is19 := -18/25*is10;
rs20 := 1 + rs15 + rs17;
is20 := is15 + is17;
rs21 := 9/2 + rs16;
is21 := -1/2 + is16;
rs22 := -is1*is21 + rs1*rs21;
is22 := is1*rs21 + is21*rs1;
rs23 := -is20*Y[6] + rs20*Y[2];
is23 := is20*Y[2] + rs20*Y[6];
rs24 := rs23 + rs12;
is24 := is23 + is12;
rs25 := rs18 + rs19 + rs22;
is25 := is18 + is19 + is22;
rs26 := rs4*(36/59*is24 + rs24) - is4*(-36/59*rs24 + is24);
is26 := rs4*(-36/59*rs24 + is24) - (-36/59*is24 - rs24)
 *is4;
rs27 := -is25*Y[7] + rs25*Y[3];
is27 := is25*Y[3] + rs25*Y[7];
rs28 := rs27 + rs26;
is28 := is27 + is26;
rs29 := -36/59*((-is2*rs28 - is28*rs2)*is14 - rs14*(is2
 *is28 - rs2*rs28))*is13 + 36/59*((is2*is28 - rs2*rs28)
 *is14 - rs14*(is2*rs28 + is28*rs2))*rs13 - ((is2*is28 - rs2
 *rs28)*is14 - rs14*(is2*rs28 + is28*rs2))*is13 + ((is2*rs28
 + is28*rs2)*is14 + rs14*(is2*is28 - rs2*rs28))*rs13;
is29 := -36/59*((-is2*is28 + rs2*rs28)*is14 + rs14*(is2
 *rs28 + is28*rs2))*is13 - 36/59*((is2*rs28 + is28*rs2)*is14
 + rs14*(is2*is28 - rs2*rs28))*rs13 + ((is2*rs28 + is28*rs2)
```

```
*is14 + rs14*(is2*is28 - rs2*rs28))*is13 + ((is2*is28 - rs2
*rs28)*is14 - rs14*(is2*rs28 + is28*rs2))*rs13;
YP[3] := rs29;
YP[7] := is29;
YP[1] := Y[2];
YP[5] := Y[6];
YP[2] := Y[3];
YP[6] := Y[7]
end
```

Now we need the values of F1 and its 1st and 2nd derivatives at z0, to be used as starting point by dsolve. Note however that, after changing variables z = S+Mt, we have

```
> [f(z), diff(f(z), z), diff(f(z), z, z)] : % =~ PDEtools[dchange]({z = S + t*M,
    f(z) = F(t)}, %, [t, F(t)], known = all)
```

$$\left[f(z) = F(t),\ \frac{d}{dz} f(z) = \frac{\frac{d}{dt} F(t)}{\Delta},\ \frac{d^2}{dz^2} f(z) = \frac{\frac{d^2}{dt^2} F(t)}{\Delta^2} \right] \qquad \textbf{(2.1.3.2.26)}$$

where on the lhs f(z) represent $F_1(a, b1, b2, c, x, z)$ and on the rhs F(t) represents $F_1(a, b1, b2, c, x, t)$ so to compute $[F(t), \dot{F}(t), \ddot{F}(t)]$ we compute

$$> convert\left(solve\left(convert\left(\left[f(z) = F(t),\ \frac{d}{dz} f(z) = \frac{\dot{F}(t)}{\Delta},\ \frac{d^2}{dz^2} f(z) = \frac{\ddot{F}(t)}{\Delta^2} \right], D\right), convert([F(t), \dot{F}(t), \ddot{F}(t)], D)\right), diff\right)_1$$

$$\left[F(t) = f(z),\ \frac{d}{dt} F(t) = \left(\frac{d}{dz} f(z)\right)\Delta,\ \frac{d^2}{dt^2} F(t) = \left(\frac{d^2}{dz^2} f(z)\right)\Delta^2 \right] \qquad \textbf{(2.1.3.2.27)}$$

In this particular example, Delta = 1 so this observation is not relevant.
We then have

$$> DF := subs\left(z = z0, \mathrm{Delta} = Delta\_val, \left[F1, \mathrm{Delta}\cdot\text{'diff'}(F1, z), \Delta^2\cdot\text{'diff'}(F1, z, z)\right]\right)$$

$$DF := \left[F_1(a, b1, b2, c, x, z0),\ \left(1 - \frac{36\,\mathrm{I}}{59}\right) \frac{d}{dz0} F_1(a, b1, b2, c, x, z0),\ \left(\frac{2185}{3481} - \frac{72\,\mathrm{I}}{59}\right) \frac{d^2}{dz0^2} F_1(a, b1, b2, c, x, z0) \right] \qquad \textbf{(2.1.3.2.28)}$$

```
> subs(z0 = z0_val, eval(DF, P))
Hypergeometric: case: a = c
Hypergeometric: case: a = c
Hypergeometric: case: a = c
```

$$\left[{}_1F_0\left(2 + \mathrm{I};;\ \frac{18}{25}\right) {}_1F_0\left(\frac{3}{2} - \frac{\mathrm{I}}{2};;\ \frac{9}{25} + \frac{9\,\mathrm{I}}{25}\right), \left(\frac{141}{118} - \frac{167\,\mathrm{I}}{118}\right) {}_1F_0\left(2 + \mathrm{I};;\ \frac{18}{25}\right) {}_1F_0\left(\frac{5}{2} - \frac{\mathrm{I}}{2};;\ \frac{9}{25} + \frac{9\,\mathrm{I}}{25}\right), \left(-\frac{1697}{6962} - \frac{19238\,\mathrm{I}}{3481}\right) \right. \qquad \textbf{(2.1.3.2.29)}$$

$$ {}_1F_0\left(2+I;;\frac{18}{25}\right)\ {}_1F_0\left(\frac{7}{2}-\frac{I}{2};;\frac{9}{25}+\frac{9\,I}{25}\right)\Big] $$

> *DFval* := *evalf*((**2.1.3.2.29**))

$$ DFval := [-8.14736299180347 + 24.8904906590253\,I,\ 2.70043561611901 + 66.0074590509333\,I,\ 72.6426963778800 + 258.691598570864\,I] \quad \textbf{(2.1.3.2.30)} $$

> *ini* := *Array*([Re(*DFval*[1]), Re(*DFval*[2]), Re(*DFval*[3]), 0, Im(*DFval*[1]), Im(*DFval*[2]), Im(*DFval*[3]), 0], *datatype* = *float*[8]);

$$ ini := [-8.14736299180347,\ 2.70043561611901,\ 72.6426963778800,\ 0.,\ 24.8904906590253,\ 66.0074590509333,\ 258.691598570864,\ 0.] \quad \textbf{(2.1.3.2.31)} $$

> *dsol* := *dsolve*(*numeric*, *procedure* = *thisproc*, *initial* = *ini*, *start* = 0, *procvars* = [*rF1*(*t*), *rDF1*(*t*), *rDDF1*(*t*), *rdummy*(*t*), *iF1*(*t*), *iDF1*(*t*), *iDDF1*(*t*), *idummy*(*t*)]);

dsol := **proc**(*x_rkf45*) ... **end** **(2.1.3.2.32)**

> *dsol*(*subs*(*t_val*, *t*))[[2, 6]]

$$ [rF1(t) = 1112.12021236804,\ iF1(t) = -254.420480401349] \quad \textbf{(2.1.3.2.33)} $$

> *SFloat*(*rhs*(*dsol*(*subs*(*t_val*, *t*))[2] + *I*·*dsol*(*subs*(*t_val*, *t*))[6]))

$$ 1112.12021236803662 - 254.420480401348783\,I \quad \textbf{(2.1.3.2.34)} $$

> *AFval*

$$ 1112.12017081988 - 254.420420227346\,I \quad \textbf{(2.1.3.2.35)} $$

> *evalf*[10](%% -%)

$$ 0.000041 - 0.0000602\,I \quad \textbf{(2.1.3.2.36)} $$

>

Construct now the procedure used in the code by replacing a,b1,b2,c,x by p1, p2, p3, p4, p5

>

Concatenated Taylor series expansions covering the whole complex plane

>

Subproducts

Improvements in the numerical evaluation of hypergeometric functions

>

Evalf: an organized structure to implement the numerical evaluation of special functions in general

Description

- Evalf is both a command and a package of commands for the *numerical* evaluation of mathematical expressions and functions, *numerical* experimentation, and fast development of *numerical* algorithms, taking advantage of the advanced *symbolic* capabilities of the Maple computer algebra system. This kind of *numerical/symbolic* environment is increasingly relevant nowadays, when rather complicated mathematical expressions and advanced special functions, as for instance is the case of the Heun and Appell functions, appear more and more in the modeling of problems in science.

- The Evalf environment is also an excellent helper for *understanding how numerical algorithms work*, placing some of the typical numerical approaches used in the literature at the tip of your fingers, in a flexible and friendly manner.
- The Evalf command allows, among other things, for the indication of different numerical methods to evaluate the mathematical functions involved in an algebraic expression. In this version, Evalf implements optional arguments for the 10 Heun and 4 Appell functions. For anything else Evalf works the same as the standard evalf command. The options implemented for numerical evaluation, generally speaking, are divided into four groups:
  - – restrictive options;
  - – numerical methods options;
  - – management options;
  - – information options;
- As a package, Evalf contains the following commands:

Add	Evalb	GenerateRecurrence	PairwiseSummation
QuadrantNumbers	Singularities	Zoom	

Brief description of the commands of the Evalf package

- Add accepts a procedure with a formula that depends on one integer parameter, say n, or two of them, and numerically evaluates the formula adding it from $n = 0$ until the result converges with the current value of Digits, or the value $n = 10000$ *Digits* is reached.
- Evalb works as evalb but can handle boolean expressions involving the functions as And, Or, Not and interpreting the integer types integer, posint, negint, nonposint, nonnegint, even and odd in an extended sense to include floats, for example: 1. is considered of type integer and $-4.$ is considered of type negint and even.
- GenerateRecurrence accepts a mathematical function (currently working only with the four Appell functions) and returns a procedure to compute the nth coefficient of a power series expansion of the given function around the origin as a function of the $n - 1$ previous coefficients, where n depends on the function given.
- PairwiseSummation accepts a one-dimensional Array or a procedure of one argument, say U and performs a *pairwise summation* of $U(j)$ for j from given m to n, or from the lower to the upper bound of the given Array. Pairwise summation is a technique to add floating-point numbers that significantly reduces the accumulated round-off error if compared to adding the numbers one at a time.
- QuadrantNumbers accepts a complex number and returns the quadrant of the complex plane where the number is located, or it accepts a list of numbers, and returns an Array of four lists, corresponding to each quadrant, of random complex numbers that are around the numbers indicated.
- Singularities accepts a Heun or Appell function, for instance with numerical values for the function's parameters, and returns the singularities of the linear ODE satisfied by the given

function.

- Zoom is used to zoom within the last plot computed using Evalf for concatenated Taylor expansions used when performing numerical computations of Heun or Appell functions.

As usual, you can load the Evalf package using the with command, or invoke Evalf commands using the long form, e.g. as in Evalf:-Add.

>

To be done

- Simplification of Appell functions - use ladder differential operators
- Expressing integrals in terms of Appell functions
- Expressing solutions to differential equations in terms of Appell functions