Cauchy's integral formula and using it to compute the regularized hypergeometric function

```
> restart; interface(version);
  with(Student[Calculus1]):
  Digits:=18;
  unprotect(gamma): gamma:='gamma': # else it is Euler's constant 0.577... and protected against changing it
```

Classic Worksheet Interface, Maple 12.02, Windows, Dec 10 2008 Build ID 377066

$$Digits := 18$$

Let be γ a closed curve in the complex plane, defined on an interval $[a, ..., b]$, $\gamma(a) = \gamma(b)$. Then the integral along γ over f is defined as

```
> 'Int( f(t), t =`gamma` .. ``) = Int(f(gamma(t))*D(gamma)(t), t = a .. b)';
```

$$\int_{\gamma} f(t)\,dt = \int_a^b f(\gamma(t))\,D(\gamma)(t)\,dt$$

Cauchy's integral formula states, that for a holomorphic function under certain conditions one has

```
> '(2*Pi*I) *f(z0) = Int( f(z)/(z - z0), z =`gamma` .. ``)';
  ``='Int(f(gamma(t))/(gamma(t) - z0)*D(gamma)(t), t = a .. b)';
```

$$2\,I\,\pi\,f(z0) = \int_{\gamma} \frac{f(z)}{z - z0}\,dz$$

$$= \int_a^b \frac{f(\gamma(t))\,D(\gamma)(t)}{\gamma(t) - z0}\,dt$$

If $\gamma := t \rightarrow radius\,\mathbf{e}^{(2\,I\,\pi\,t)} + center$ is a circle, $z0$ lies inside and f is holomorphic on that open disk then the conditions are satisfied.

Here the factor $2\,I\,\pi$ occures on both sides and the formula reads as

```
> center:= 'center':
  radius:= 'radius':
  gamma:= t -> radius*exp(2*Pi*I*t) + center;
  ``;
  'f(z0) = Int( f(z)/(z - z0), z =`gamma` .. ``)';
  ``='Int(f(gamma(t))/(gamma(t) - z0)*(gamma(t)-center), t = 0 .. 1)',
  'diff(gamma(t),t)' = '2*I*Pi*(gamma(t) - center)'; #is(%);
```

$$\gamma := t \rightarrow radius\,\mathbf{e}^{(2\,I\,\pi\,t)} + center$$

$$f(z0) = \int_{\gamma} \frac{f(z)}{z - z0}\,dz$$

$$= \int_0^1 \frac{f(\gamma(t))\,(\gamma(t) - center)}{\gamma(t) - z0}\,dt,\ \frac{d}{dt}\gamma(t) = 2\,I\,\pi\,(\gamma(t) - center)$$

For the special choice $z0 = center$ the formula simplifies even more, since $\left.\frac{f(\gamma(t))\,(\gamma(t) - center)}{\gamma(t) - z0}\right|_{z0 = center} = f(\gamma(t))$ since terms cancel out:

```
> 'f(center) = Int(f(gamma(t)), t=0..1)';
```

$$f(center) = \int_0^1 f(\gamma(t))\,dt$$

That is a (complex valued) integral on the unit interval, one can approximate it by the trapezoid method, here 8 steps are used as an example.

```
> nSteps:=8;
  'nSteps*ApproximateInt(f(gamma(t)), t=0..1, iterations = 1,  partition = nSteps, output=sum, method=trapezoid)':
  'eval(%, z0=center)':
  TrapezSum:=value(%);
```

$$nSteps := 8$$

$$TrapezSum := f(radius + center) + f\left(radius\left(\frac{\sqrt{2}}{2} + \frac{1}{2} I \sqrt{2}\right) + center\right) + f(radius\ I + center) + f\left(radius\left(-\frac{\sqrt{2}}{2} + \frac{1}{2} I \sqrt{2}\right) + center\right) + f(-radius + center) + f\left(radius\left(-\frac{\sqrt{2}}{2} - \frac{1}{2} I \sqrt{2}\right) + center\right) + f(-radius\ I + center) + f\left(radius\left(\frac{\sqrt{2}}{2} - \frac{1}{2} I \sqrt{2}\right) + center\right)$$

Then $f(center)$ is numerical approximated by the formula

```
> 'f(center)' = 'TrapezSum/nSteps';
```

$$f(center) = \frac{TrapezSum}{nSteps}$$

That means: take the unit roots (scaled onto the circle), evaluate f in that points and take the average - then numerical one gets $f(center)$.

Simple example

For $f = exp$ and a point $zTst$ in the complex plane take a circle of radius 1/256 to compute $\mathbf{e}^{zTst}$ - just to see how large the errors are:

```
> zTst:=2.0001+1.001*I;
  tstCircle:= [center = zTst, radius=1/256];
```

$$zTst := 2.0001 + 1.001\ I$$

$$tstCircle := \left[center = 2.0001 + 1.001\ I,\ radius = \frac{1}{256}\right]$$

```
> 'exp(zTst)' = '1/nSteps*eval(eval(eval(TrapezSum, tstCircle), f =exp), z0=zTst)';;
  evalf(%);
  `error (absolute, relative)`  = abs(lhs(%) - rhs(%)), abs(1 - rhs(%)/lhs(%));
```

$$\mathbf{e}^{zTst} = \frac{eval(TrapezSum, tstCircle)\Big|_{f = exp}\Big|_{z0 = zTst}}{nSteps}$$

$$3.98650300737221150 + 6.22228772457518097\ I = 3.98650300737221149 + 6.22228772457518097\ I$$

$$error\ (absolute,\ relative) = 0.1\ 10^{-16},\ 0.151601012308512526\ 10^{-17}$$

If errors would be larger then one could increase the number of steps (adaptively).

Application: regularization for the hypergeometric 2F1

The function $\frac{hypergeom([a, b], [c], z)}{\Gamma(c)}$ is analytic in its parameters (for fixed and admissible z, c.f. Lebedev's book on special functions for a proof) and called the regularization. However Maple can not compute it, if c is a negative integer.

One has the following fact (see Abramowitz & Stegun, 15.1.2, p.556):

```
> m in `IN`, Limit( hypergeom([a, b],[c],z)/GAMMA(c), c = -m) = 'limF(a,b,c,z)';
  limF := (a, b, c, z) -> pochhammer(a,-c+1)*pochhammer(b,-c+1)/(-c+1)! *
      z^(-c+1)*hypergeom([b-c+1, a-c+1],[-c+2],z);
```

$$m \in IN,\quad \lim_{c \to (-m)} \frac{hypergeom([a, b], [c], z)}{\Gamma(c)} = limF(a, b, c, z)$$

$$limF := (a, b, c, z) \to \frac{pochhammer(a, -c + 1)\ pochhammer(b, -c + 1)\ z^{(-c+1)}\ hypergeom([b - c + 1, a - c + 1], [-c + 2], z)}{(-c + 1)!}$$

Check that by choosing some test data for a, b, z while for c some negative integer taken:

```
> cTst:=-3.0;
  tstData:=[a=0.1, b=-2.3, z =1.1 + 1.3*I];
```

$$cTst := -3.0$$

$$tstData := [a = 0.1, b = -2.3, z = 1.1 + 1.3\ I]$$

Maple can not calculate the value directly, not even numerically:

```
> eval(hypergeom([a,b],[c],z)/GAMMA(c), tstData):
  fTst:=unapply(%,c);
```

$$fTst := c \to \frac{hypergeom([-2.3, 0.1], [c], 1.1 + 1.3\ I)}{\Gamma(c)}$$

```
> 'fTst(cTst)': '%'=%;
```

$$fTst(cTst) = Float(undefined) + Float(undefined)\ I$$

However the method from above works:

```
> tstCircle:= [center = cTst, radius=1/256];
  ``;
  'eval(1/nSteps*TrapezSum, tstCircle)';
  eval(%, f =fTst):
  eval(%, z0 = cTst):
  ``=evalf(%);
  ``;
  'eval(limF(a,b,cTst,z), tstData)'; evalf(%):
  ``=%;
```

$$tstCircle := \left[center = -3.0, radius = \frac{1}{256}\right]$$

$$eval\left(\frac{TrapezSum}{nSteps}, tstCircle\right)$$

$$= -0.118486240610449468 + 0.0546427031746235087\, I$$

$$eval(limF(a, b, cTst, z), tstData)$$

$$= -0.118486240610449467 + 0.0546427031746235088\, I$$

Note that this also works, if one is not exactly in such a critical point, but only close to it (and a direct method would possibly lead to large numerical errors in usual precision).