

Contour Integration

or

what is still missing in Mathematica

Part 2 : Construction of sophisticated Contour Paths,
Location of Poles inside/outside Closed Contours,
Special Functions Representations by Contour Integrals,
Transformation of Improper Integrals into Contour Integrals and
Investigation of Action Integrals.

Prof. Dr. Robert Kragler

Weingarten University of Applied Sciences

kragler@hs-weingarten.de

Abstract :

While the *first part* was devoted primarily to the main procedures `calculateResidues` and `ContourIntegration` applied to a wide class of complex functions $f(z)$ which are rational polynomials, products of rational and trigonometric/ hyperbolic functions, rational functions consisting of trigonometric/hyperbolic functions. However, the investigations of the *second part* of this paper are special topics which occur in the context of contour integration and are of interest in themselves. The issues discussed in this paper are :

- (1) introduction of a language for creation and visualization of non-trivial integration paths consisting of polylines and circular arcs such as contours γ which exclude certain poles or branch cuts, or the sophisticated contour for Meijer G-functions meandering around integer singularities but avoiding half-integral ones, or the Pochhammer double-loop contour for the evaluation of the so-called Euler's integral etc. ;
- (2) criterium for the determination of poles inside/outside an arbitrary closed contour ;
- (3) symbolic evaluation of the integral representation for special functions such as Meijer G-function or Euler's integral for Beta function etc.
- (4) transformation of improper integrals (along the real axis) into exotic contour integrals with the help of change of variables, e.g. $\int_0^{+\infty} \frac{1}{x^3+1} dx$ with variables $z \rightarrow r e^{i\phi}$ where $\phi \rightarrow \frac{2\pi}{3}$ etc. ;
- (5) evaluation of action integrals such as $\oint p_r dr = \oint \sqrt{\mathcal{A} + 2 \frac{B}{r} - \frac{C}{r^2}} d\mathbf{r}$ by mapping the complex plane \mathbb{C} on the Riemann sphere \mathcal{R} ;

□ Initialization

In order to execute the subsequent *Mathematica* code with the examples given the *Mathematica* package **ContourIntegration`** must be loaded first. It should be located in the same subdirectory from where the current notebook **ContourIntegration_P2.nb** is revoked.

```
Clear["Global`*"];
SetDirectory[NotebookDirectory[]]
Get["ContourIntegration`"]
```

C:\eMail_Attachment\ContourIntegration_final

The *Mathematica* package comprises all definitions, procedures, replacement rules etc. which are required to run the main procedure **ContourIntegral** etc. . After successful execution of the package the *Mathematica* version, date and time are shown.

VersionDateTime

Mathematica V10..14.1 for Microsoft Windows (64-bit) (April 11, 2016)
date= August 13, 2016; time= 15:21h

Special notations

For sake of better readability some special notations will be introduced and used throughout the notebook which are given here.

Numerical contour integrals $\oint_{\gamma} f(z) dz$ and $\oint_a^b f(z) dz$,

Notation[$\oint_{\gamma} f dz \Rightarrow$ NIntegrate[f_, Evaluate[Join[{z_}, \gamma_]]] ,WorkingForm \rightarrow tF]

Notation[$\oint_{\theta=a}^{\theta=b} f dz_{z \rightarrow g} \Rightarrow$ NIntegrate[Evaluate[Simplify[$\frac{f_ Dt[z_] /. z_{\rightarrow} g_}{Dt[\theta_]}]$], {theta_, a_, b_}] ,WorkingForm \rightarrow tF]

Line integrals $\int_{\mathcal{L}(t)} f(R(t)) \cdot dt[R]$

Notation[$\int_{\mathcal{L}_-, p_-} f_ . dt[r_] \Leftrightarrow$ LineIntegral[f_ . Dt[r_] , \mathcal{L}_-, p_-, r_]]

ContourIntegration_P2.nb

Symbolic contour integrals $\oint_{\text{selPol}, \text{polRange}, \text{onoff}} f(z) dz$

Notation $\left[\oint_{\text{selectPoles}_-, \text{polesRange}_-, \text{onoff}_-} f_{-} dz_{-} \Leftrightarrow \text{ContourIntegral}[f_{-}, z_{-}, \text{selectPoles}_-, \text{polesRange}_-, \text{onoff}_-] \right]$

Replacement Rules and Shortcuts

This are *substitution rules* for $\{\sin(\theta), \cos(\theta)\}$ and $\{\sinh(\theta), \cosh(\theta)\}$ not included in the package **ContourIntegration**

```
z=. ;
trigRule:= {Sin[θ_] →  $\frac{1}{2i} \left( z - \frac{1}{z} \right)$ , Cos[θ_] →  $\frac{1}{2} \left( z + \frac{1}{z} \right)$ , Csc[θ_] →  $\frac{2i}{z - \frac{1}{z}}$ , Sec[θ_] →  $\frac{2}{z + \frac{1}{z}}$ ,
  Tan[θ_] →  $-i \frac{(z - \frac{1}{z})}{(z + \frac{1}{z})}$ , Cot[θ_] →  $i \frac{(z + \frac{1}{z})}{(z - \frac{1}{z})}$ , dθ →  $\frac{1}{i z} dz$  }; (* z = ei θ *)
hypRule:= {Sinh[θ_] →  $\frac{1}{2} \left( z - \frac{1}{z} \right)$ , Cosh[θ_] →  $\frac{1}{2} \left( z + \frac{1}{z} \right)$ , Csch[θ_] →  $\frac{2}{(z - \frac{1}{z})}$ , Sech[θ_] →  $\frac{2}{(z + \frac{1}{z})}$ ,
  Tanh[θ_] →  $\frac{(z - \frac{1}{z})}{(z + \frac{1}{z})}$ , Coth[θ_] →  $\frac{(z + \frac{1}{z})}{(z - \frac{1}{z})}$ , dθ →  $\frac{1}{z} dz$  }; (* z = eθ *)
```

In order to make the code more transparent some *shortcuts* will be used throughout this paper :

fl=Flatten, **sf=Simplify**, **fs=FullSimplify**, **sF=StandardForm**, **tF=TraditionalForm**, **cF=ColumnForm**, **ce=ComplexExpand**, **hF=HoldForm**, **th=AbsoluteThickness[2]** and **th1=AbsoluteThickness[1]** and **polyForm=PolynomialForm[#, TraditionalOrder→True]&** .

Some additional *rules* are defined :

AbsRule = $\left\{ \text{Abs}[x_ + i y_] \rightarrow \sqrt{x^2 + y^2} \right\}$, **reIm** = {Re[#],Im[#]}& , **ratChop[v_]** := Rationalize[Chop[expr]] .

Global variables \$<name> being used are : **\$κ**, **\$sing**, **\$branchCut**, **\$rootObj**, **\$poles**, **\$polesType**, **\$orderC**, **\$i**, **\$circ**.

In Mathematica V10 there are three useful procedures in the package **ComplexAnalysis** :

The following definitions circumvent the context **ComplexAnalysis`**

```
{branchCuts, branchPoints, holomorphicQ} =
  {ComplexAnalysis`BranchCuts, ComplexAnalysis`BranchPoints, ComplexAnalysis`HolomorphicQ };

? ComplexAnalysis`BranchCuts ComplexAnalysis`BranchPoints ComplexAnalysis`HolomorphicQ
```

BranchCuts[f, z] gives the branch cuts of f with respect to the variable z. BranchCuts[f] returns the branch cuts of the pure function f.

BranchPoints[f, z] gives the branch points of f with respect to the variable z. BranchPoints[f] returns the branch points of the pure function f.

HolomorphicQ[f, z] returns True if f is a holomorphic function. HolomorphicQ works best if there are no symbolic parameters.

■ Prolog

Contour integration is a method in complex analysis for the calculation of integrals along a closed path γ in the complex plane \mathbb{C} (see [1, 2]). In this second part some more sophisticated applications will be demonstrated.

Topics listed in Abstract :

(1) introduction of a language for creation and visualization of non-trivial integration paths consisting of polylines and circular arcs such as contours γ which exclude certain poles or branch cuts, or the sophisticated contour for Meijer G-functions meandering around integer singularities but avoiding half-integral ones, or the Pochhammer double-loop contour for the evaluation of the so-called Euler's integral etc. ;

(2) criterium for the determination of poles located inside/outside an arbitrary closed contour ;

(3) integral representations of Meijer G-functions $G_{p,q}^{m,n} \left(z \left| \begin{matrix} a_1, \dots, a_p \\ b_1, \dots, b_q \end{matrix} \right. \right)$;

(4) transformation of improper integrals (along the real axis) into 'exotic' contour integrals with the help of change of variables. It will be shown that due to a suitable *change of integration variable* certain types of *improper* integrals (along the real axis \mathbb{R}) can be transformed into contour integrals with a closed contour γ in \mathbb{C} .

(5) evaluation of action integrals such as $\oint \mathbf{p}_r d\mathbf{r} = \oint \sqrt{\mathcal{A} + 2 \frac{\mathcal{B}}{r} - \frac{\mathcal{C}}{r^2}} d\mathbf{r}$ by mapping the complex plane \mathbb{C} on the Riemann sphere \mathcal{R} ;

A detailed investigation of multi-valued functions $f(z)$ with branch cuts and evaluation of corresponding contour integrals will be given in a subsequent paper Part 3.

■ Motivation for Symbolic Contour Integration

Contour integration is a method in complex analysis for the calculation of certain integrals along a path γ in the complex plane \mathbb{C} ; this method is closely related to the calculus of residues as shown before. One use of contour integrals is the evaluation of integrals along the real axis that are not easily determined by using only methods with real variables. The main applications of contour integration are :

- direct integration of a complex-valued function $f(z)$ along a curve γ in \mathbb{C} ,
- application of *Cauchy's integral theorem*,
- application of the *residue theorem*.

- special functions often defined by *contour integral representations*

See, for example, the definition of the Γ -function in terms of Hankel's contour integral as found on *The Wolfram Functions Site* [4] with URL : <http://functions.wolfram.com/> .

$\Gamma(z) = \frac{1}{e^{2\pi iz} - 1} \int_{\mathcal{L}} e^{-t} t^{z-1} dt$. The path of integration \mathcal{L} starts at $\infty + i0$ above the real axis, goes to $\rho + i0$, encircles the origin in counter-clockwise direction with radius ρ to the point $\rho - i0$ below the real axis, and returns to the point $\infty - i0$.

$$\Gamma(z) := \frac{1}{e^{2\pi iz} - 1} \text{ContourIntegrate}[e^{-t} t^{z-1}, \{t, \mathcal{L}\}]$$

A *Mathematica* procedure **ContourIntegrate**[**f**[**z**,**t**],**{t,L}**] is suggested for the (symbolic) calculation of the contour integrals. However, this essential procedure is not yet implemented in *Mathematica* nor available elsewhere which is astonishing. There are several reason why the implementation has not been done (Private communications with M. Trott / WRI (2010)).

This was the motivation for the author to implement a corresponding procedure which covers many nontrivial contour integrals but does not claim to cope with all possible cases and situations occuring in the context of contour integrations.

Special notation for contour integrals

$$\oint_{\text{selectPoles_}, \text{polesRange_}, \text{onoff_}} f_ dz_ \Leftrightarrow \text{ContourIntegral}[f_ , z_ , \text{selectPoles_} , \text{polesRange_} , \text{onoff_}]$$

? ContourIntegral

ContourIntegral[f, z, selectPoles_All, polesRange_{}, onoff_"On"] evaluates contour integrals symbolically in the complex plane \mathbb{C} by means of the residues for the poles selected. 'f' denotes the integrand f(z) of the contour integral where 'z' is the integration variable $z \in \mathbb{C}$. The parameter 'selectPoles'={ i, j, ... } a subset of poles can be selected to be considered for the residues; default value for 'selectPoles' is All. The sum of residues is evaluated using calculateResidues which has the same parameter list as ContourIntegral. For the final result the sum of (selected) residues is only multiplied by $2\pi i$ and returned in the variable \mathcal{J} for the contour integral. With "No" intermediate printout is complete suppressed.

ContourIntegration_P2.nb

$$\text{ContourIntegral}\left[\frac{1}{\left(z^3+1\right)^2},z,\{2,3\},\{\},\text{"On"}\right]$$

The same result is obtained with the special notation with the symbol \oint ...

$$\mathcal{J}=\oint_{\{2,3\},\{\},\text{"No"}}\frac{1}{\left(z^3+1\right)^2}\mathrm{d}z$$

$$\text{all residues: } \sum_i \text{res}\Sigma = -\frac{2}{9}; \text{ selectPoles} = \{2,3\}$$

$$\text{Contour integral } \mathcal{J} = \oint \frac{1}{\left(1+z^3\right)^2}\mathrm{d}z = -\frac{4\,\mathrm{i}\,\pi}{9}$$

$$-\frac{4\,\mathrm{i}\,\pi}{9}$$

■ (1) Creation and Visualization of Contours

In order to evaluate real-valued integrals the integrand $f(x)$ is continued to the complex plane \mathbb{C} and the integration interval on the real axis $(a,b) \in \mathbb{R}$ is extended to a closed curve γ by attaching in the simplest case a semi-circle in the upper/lower half-plane $\mathbb{H}_{+/-}$. Often the contribution of the semi-circle to the integral vanishes if the radius $R \rightarrow \infty$ so that only the real-axis part of the integral remains.

□ Contour created by polylines with `showPolygonalContour`

In the simplest case a closed contour is made of a list γ_i of points $P_k = (x_k + i y_k) \in \mathbb{C}$ which are connected by polylines. This is achieved with the procedure

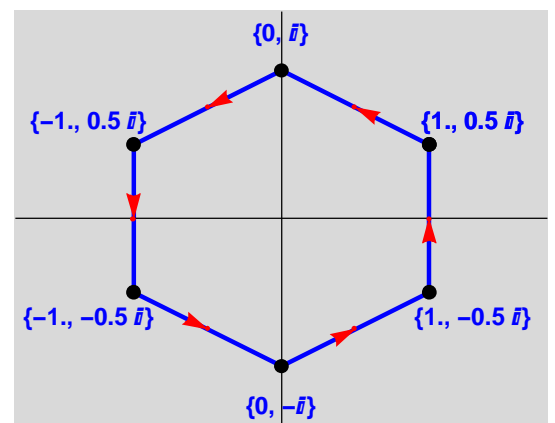
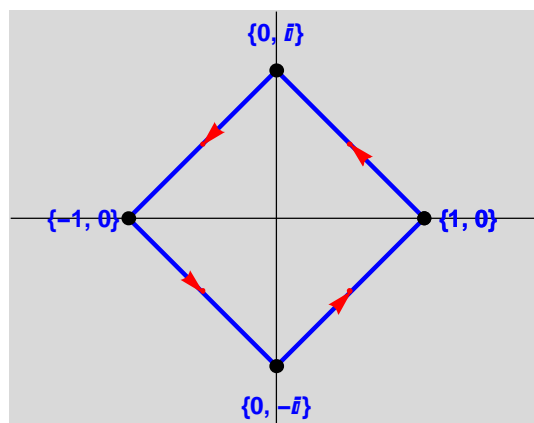
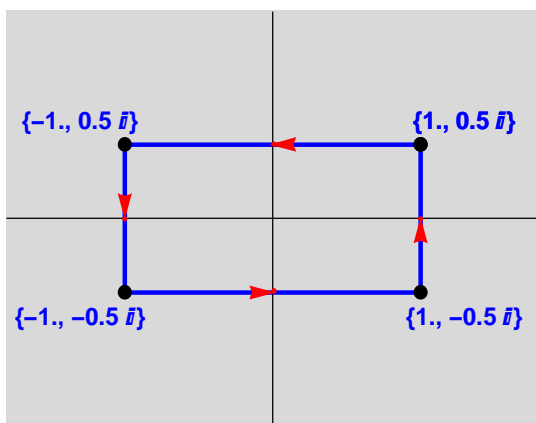
```
? showPolygonalContour
```

Rectangular and diamond-like shaped contours

Here, various contours (of rectangular and diamond-like shape) are shown

```
z = . ;
γ1 = {1 + i .5, -1 + i .5, -1 - i .5, 1 - i .5, 1 + i .5};
γ2 = {1, i, -1, -i, 1};
γ3 = {1 + i .5, i, -1 + i .5, -1 - i .5, -i, 1 - i .5, 1 + i .5};
showPolygonalContour[{{γ1}, {γ2}, {γ3}}]
```

ContourIntegration_P2.nb



Polygonal contour with vertex coordinates

Furthermore, the *coordinates* of vertex points (defining the polygonal contour) and additional *points* (such as roots, singularities etc.) given in the list **cmplxPts** can be displayed with **showPolygonalContour1** which is an improved version of the procedure above.

? **showPolygonalContour1**

showPolygonalContour[*γlist*:{{_}..}, *range_*:Full, *cmplxPts_*:Null] draws *n* (*n*=1|2|...) polygonal closed contours $\gamma_1, \gamma_2, \dots, \gamma_n$ within the '*range*'= {{ x_0, x_1 }, { y_0, y_1 }} (default is Full) where the coordinates of the vertex points $P_i \in \mathbb{C}$ are shown and the direction of the contour path is indicated by arrows centered halfway on each of the connecting (poly)lines between subsequent vertices P_i, P_{i+1} . '*cmplxPts*' is the set of points (default is Null) enclosed by the contour with coordinates given in \mathbb{C} .

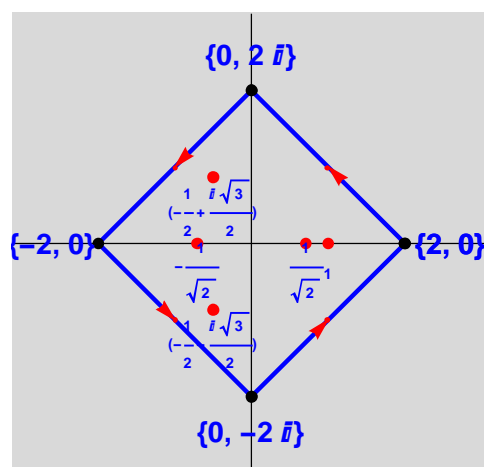
Here, for example the complex roots of the polynomial $\mathcal{P}_5(z) = 0$ are displayed in addition :

```

 $\mathcal{P}[z_] := z^5 - \frac{z^3}{2} - z^2 + \frac{1}{2};$ 
 $\xi = z /. \text{Solve}[\mathcal{P}[z] == 0, z];$ 
 $\text{cmplxRoots} = (\#[[1]] + i \#[[2]]) \& /@ (\{\text{Re}[\xi], \text{Im}[\xi]\}^T)$ 
 $\gamma_4 = 2 \{1, i, -1, -i, 1\};$ 
 $\text{showPolygonalContour1}[\{\{\gamma_4\}\}, \{-3.1, 3.1\}, \{-3, 3\}, \text{cmplxRoots}]$ 

```

$$\left\{1, -\frac{1}{2} - \frac{i\sqrt{3}}{2}, -\frac{1}{2} + \frac{i\sqrt{3}}{2}, -\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right\}$$

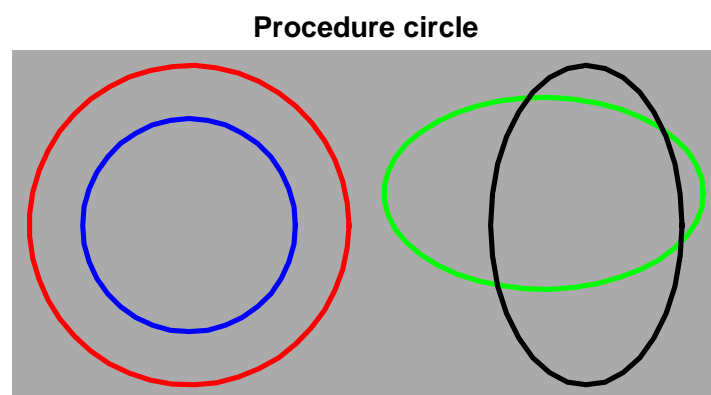


Circular contour

Because a contour path γ consists of polylines and arcs the built-in routine `Circle[{x,y},r,{theta1,theta2}]` should internally be represented by a polyline. But this seems not to be the case so that instead a user-defined version `circle[{x,y}...]` was created with an internal polyline representation which admits further processing of closed contours in the context of the *Point-In-Polygon* problem.

? circle

The subsequent test shows various circles and ellipsoids (the internal representation of which are polylines) using several instances of the procedure `circle`.



In practice, contours in the complex plane \mathbb{C} or in some half-planes $\mathbb{H}_{+/-}$ or $\mathbb{H}_{I/r}$ turn out to be more complicated if *poles/singularities* z_i are to be enclosed or excluded. Sometimes, in order to avoid *branch cuts* the contour encircling this discontinuity will be deformed into another equivalent contour.

- **Creation and visualization of contours more involved using `contourPathGeneration` and `contourPathGraphic`**

Thus, apart from trivial cases which are taken care by `showPolygonalContour1` the construction of these contours could become quite involved so that it is suitable to have procedures such as `contourPathGeneration` and `contourPathGraphic` for the generation and visualization

of more sophisticated contours γ .

? contourPathGeneration

"contourPathGeneration[pts,contour_:All,opts___] creates the contour in terms of polylines. 'pts' is the list of points $(x_k + i y_k)$ defining the contour in the complex plane \mathbb{C} . 'contour' is the index (connectivity) list numbering the sequence of points to give polylines and circles. For default value 'All' the complete index list is interpreted as a single polyline line[{1,2,3,...}]; circular arcs are defined as sublists {...,{i,j,k},...}. If an arc comprises more than 3 points, e.g. {i,j,k,l,m} \Rightarrow circ[{i,j,k,l,m}], then intermediate points (here k,l) are dropped for the calculation of the corresponding arc; if the index order is inverted to guarantee correct orientation of an arc, e.g. cir[{3,2,1}], then the coordinates of the resulting (circular) polyline will be reversed with adjacent duplicates of coordinate points being removed. If the global variable \$branchCut $\neq \{ \}$, e.g. $\{x_1 + i y_1, \dots, x_n + i y_n\}$, then an additional contour is created encircling the branch line spanned by P_1 and P_n . Two lists, {coordsContour,coordsBranchCut} are returned (which may be further used for the procedure PointInPolygonQ to determine which poles/singularities are inside the contour). 'coordsContour' describes the list of coordinate pairs of the resulting contour whereas 'coordsBranchCut' describes the contour encircling the branch cut."

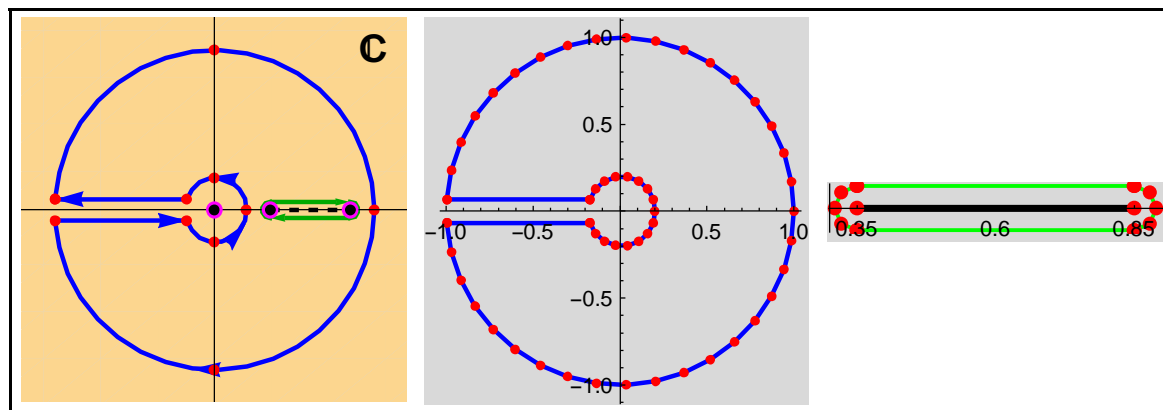
? contourPathGraphic

contourPathGraphic[pts,singPts,contour_:All,opts] shows a contour path which consists of arcs and polylines going through 'pts'. The list of points 'pts' which determines the blue contour are $P_i = x_i + i y_i$ in the complex plane \mathbb{C} . The points 'pts' are shown as red dots. 'singPts' denotes a list of singular points which could be included or excluded from the contour and are shown in black. Index list 'contour' (e.g. contour={{1,2,3},3,4,{4,5,6},6,7}) defines the sequence of circular arcs and polylines constituting the contour path. The sublists { ..., {i,j,k}, l, m, n, ... } \Rightarrow circle[{i,j,k}] specify arcs through 3 points $\{ P_i, P_j, P_k \}$. Interlaced indices (besides these sublists for arcs), e.g. {..., l, m, n, ... } \Rightarrow line[{..., l, m, n, ...}], specify polylines. The default value for 'contour' is All for which a single polyline line[{1,2,3,...,n}] results. For a closed contour the last point is equal to the first point : $P_n = P_1$. There is an optional parameter opts___ which specifies e.g. the coordinate range of the plot using PlotRange $\rightarrow \{ \{x_0, x_1\}, \{y_0, y_1\} \}$ respectively All. Moreover, with the global variable \$branchCut a branch cut can be drawn. Thus \$branchCut={0+i 0,a+i 0} denotes a branch cut along the positive real axis from $x=0$ to a .

For the description of an arbitrary contour γ which might be open or closed a proper language is provided as regards to the points P_1, \dots, P_n characterizing an arbitrary contour path :

- the *vertex list* **pts** is a list of points $\{P_1, P_2, P_3, \dots\}$ in the complex plane \mathbb{C} which define the contour;
- an *index list* **contour** (e.g. **contour** = $\{\{1, 2, 3, 4\}, 4, 5, \{7, 6, 5\}, 7, 8\}$) defines a sequence of *circular arcs* and *polylines* which constitute the contour path. The sublists $\{\{1, 2, 3, 4\}, \dots, \{7, 6, 5\}, \dots\}$ specify circular arcs, e.g. **circ** $[\{1, 2, 3, 4\}]$ and **circ** $[\{7, 6, 5\}]$, going through (at least) 3 points, for example $\{P_7, P_6, P_5\}$; reverse ordering of points changes orientation for the circle from counter-clockwise to clockwise direction. The interlaced indices such as $\{\dots, 4, 5, \dots, 7, 8\}$ specify polylines, i.e. **line** $[\{4, 5\}]$ and **line** $[\{7, 8\}]$. With the help of this kind of index list even very complicated contours can easily be defined.
- *Branch cuts* are taken into account through the global variable **\$branchCut** which is a list of branch points $\{P_1, P_2, \dots\}$ in \mathbb{C} .

In the following graph the contour is defined by the *index list* **contour** = $\{\{5, 3, 1\}, 5, 6, \{6, 7, 8\}, \{8, 9, 10\}, 10, 11\}$ and the *branch cut* **\$branchCut** is given by two *branch points* $\{(.35+0i), (.85+0i)\}$. The procedure **contourPathGeneration** generates polylines with intermediate points so that the arcs look smooth. The *branch cut* is given as a dashed black line which is encircled by a green closed contour line running above and below the branch line.



The first graph shows the result of **contourPathGraphic** $[\mathbf{pts}, \mathbf{singPts}, \mathbf{contour}]$, the second graph illustrates the auxiliary points

`coordsContour` being used for the contour and the third graph shows `coordsBranchCut` which displays the branch cut. These (auxiliary) point lists are provided by `{coordsContour, coordsBranchCut} = contourPathGeneration[pts, contour]`.

Branch cut is taken care by the procedures

? `branchCutContour` `showBranchCut`

`branchCutContour[branchPts]` creates a contour encircling a given branchcut which is defined by the global variable `$branchCut = {-a+b i, a+b i}`. A list of coordinate pairs 'bcPts' describing the branchcut contour is returned.

`showBranchCut[branchPts]` displays a branch cut spanned by 'branchPts'. It is shown as a dashed black line encircled clockwise by a green closed contour.

which create auxiliary points to define a smooth branch cut contour to be displayed as a green line encircling the dashed black branch line spanned between given branch points.

```
branchPts = {.35 + 0 i, .85 + 0 i};
bcPts = branchCutContour[branchPts]
Graphics[ showBranchCut[branchPts], ImageSize -> 150]

{{0.35, 0.04}, {0.321716, 0.0282843}, {0.31, 0.}, {0.321716, -0.0282843},
{0.35, -0.04}, {0.35, -0.04}, {0.85, -0.04}, {0.85, -0.04}, {0.878284, -0.0282843},
{0.89, 0.}, {0.878284, 0.0282843}, {0.85, 0.04}, {0.85, 0.04}, {0.35, 0.04}}
```



Auxiliary procedures for the creation of circular arcs are available in the package `ContourIntegration`.

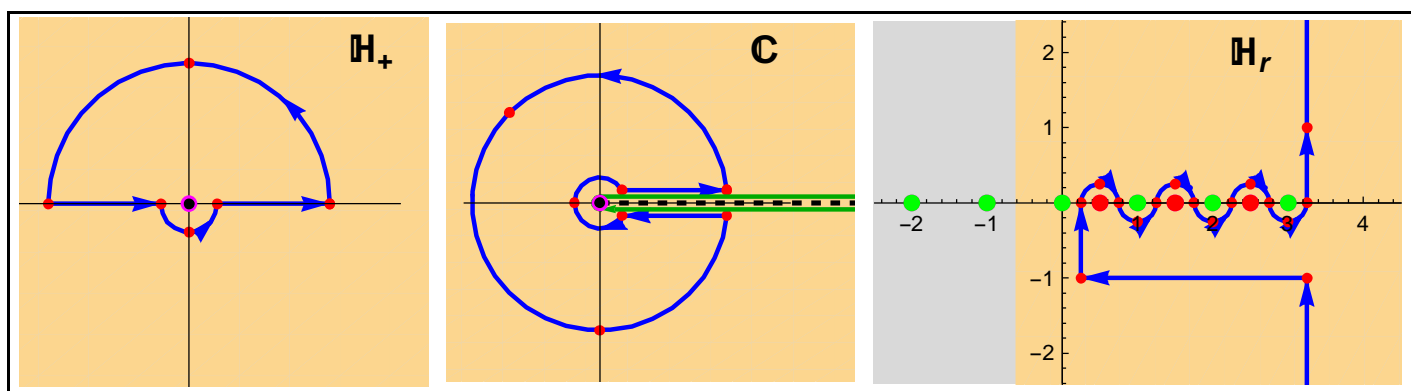
ContourIntegration_P2.nb

? ArcsAndLines Arc3 Arc circle

? vectAngles Orientation arrowList

Various types of contours

A selection of various types of contours is shown below :



Closing a contour at infinity

In order to close a contour in \mathbb{C} in most cases a semi-circle or full circle is added which contribution to the integral vanishes in the limit $R \rightarrow \infty$ so that only the real-axis part of the integral remains. But often there are situations encountered (such as the graph meandering around a number of points on the real axis) where the contour has to be closed either in the right or left half-plane $\mathbb{H}_{r/l}$. To cope with this situation it is suitable to introduce an additional parameter **close** = $\{+1, -1, +i, -i\}$ which indicates that the contour is closed either in the right/left half-plane $\mathbb{H}_{r/l}$ or in the upper/lower half-plane \mathbb{H}_{\pm} of the complex plane \mathbb{C} with a circular arc with radius $R \rightarrow \infty$.


```
close = +1; (* right half-plane  $\mathbb{H}_r$  :  $\{z \in \mathbb{C} \mid \operatorname{Re}(z) > 0\}$  *)
close = -1; (* left half-plane  $\mathbb{H}_l$  :  $\{z \in \mathbb{C} \mid \operatorname{Re}(z) < 0\}$  *)
close = +i; (* upper half-plane  $\mathbb{H}_+$  :  $\{z \in \mathbb{C} \mid \operatorname{Im}(z) > 0\}$  *)
close = -i; (* lower half-plane  $\mathbb{H}_-$  :  $\{z \in \mathbb{C} \mid \operatorname{Im}(z) < 0\}$  *)
```

Likewise, the parameter **inf** = $\{\infty, -\infty, i\infty, -i\infty\}$ could be introduced with the meaning that the contour γ will be *closed* (at infinity) by a circular arc $\mathbf{R} \rightarrow \infty$ in the right/left half-plane \mathbb{H}_r/l or in the upper/lower half-plane \mathbb{H}_\pm of \mathbb{C} .

```
inf = { $\infty$ ,  $-\infty$ ,  $i\infty$ ,  $-i\infty$ }; (*  $\mathbb{H}_r, \mathbb{H}_l, \mathbb{H}_+, \mathbb{H}_-$  *)
```

Mathematica has implemented two commands **DirectedInfinity** and **ComplexInfinity** for representing infinity.

```
? DirectedInfinity ComplexInfinity
```

DirectedInfinity[] represents an infinite numerical quantity whose direction in the complex plane is unknown.
 DirectedInfinity[z] represents an infinite numerical quantity that is a positive real multiple of the complex number z. >>

ComplexInfinity represents a quantity with infinite magnitude, but undetermined complex phase. >>

According to Weisstein [5] complex infinity is an infinite number in \mathbb{C} whose complex argument is undefined. In *Mathematica* this infinite number is represented symbolically by **ComplexInfinity** and the notation ∞ is used.

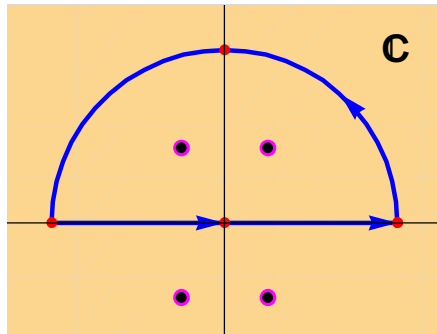
```
{ DirectedInfinity[1], DirectedInfinity[-1], DirectedInfinity[i], DirectedInfinity[-i],
  DirectedInfinity[1 + i], DirectedInfinity[1 - i], DirectedInfinity[] } /. {ComplexInfinity ->  $\infty$ }
```

$$\left\{ \infty, -\infty, i\infty, (-i)\infty, \frac{1+i}{\sqrt{2}}\infty, \frac{1-i}{\sqrt{2}}\infty, \infty \right\}$$

Examples below illustrate several contours which are created with **contourPathGraphic** :

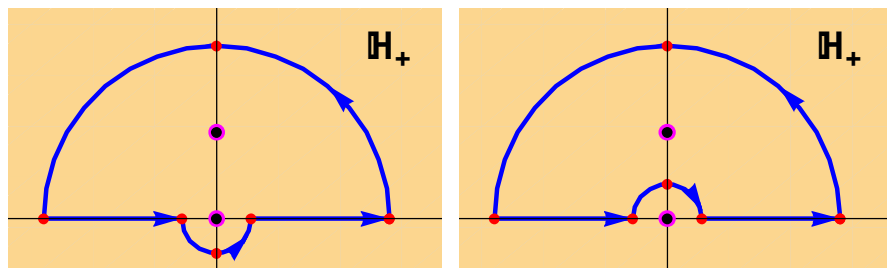
■ **Example 1: Path closed in \mathbb{H}_+**

Here, a contour γ is created which lies in the upper half-plane \mathbb{H}_+ . The function $f(z) = \frac{1}{1+z^2+z^4}$ possesses four poles in $\mathbb{C} : \left\{ \mp \frac{1}{2} - \frac{i\sqrt{3}}{2}, \mp \frac{1}{2} + \frac{i\sqrt{3}}{2} \right\}$ but only the poles $z_{3,4} = \pm \frac{1}{2} + \frac{i\sqrt{3}}{2}$ are located in \mathbb{H}_+ .



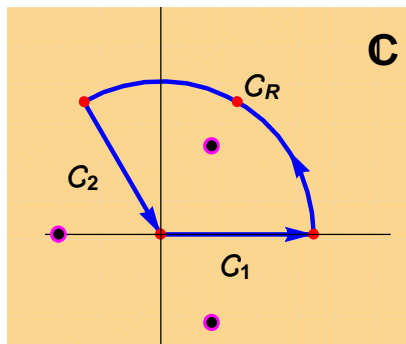
■ **Example 2: Path closed in \mathbb{H}_+ with/without inclusion of pole in 0**

The closed contour is (essentially) in \mathbb{H}_+ and includes/excludes the pole $z_0 = 0$ (by a small semi-circle around the origin) but contains the pole $z_1 = \frac{1}{2}i$.



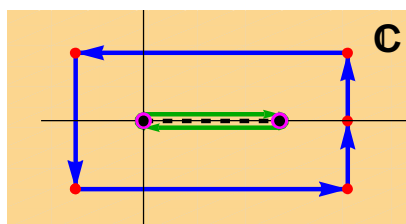
■ **Example 3: Sectional contour in \mathbb{H}_+**

Given are three poles $\left\{-1, \frac{1}{2} - \frac{i\sqrt{3}}{2}, \frac{1}{2} + \frac{i\sqrt{3}}{2}\right\}$ as solution of $z^3 + 1 = 0$. Only the pole $z_3 = \left\{\frac{1}{2} + \frac{i\sqrt{3}}{2}\right\}$ is located inside the sectional contour $\gamma = C_1 + C_R + C_2$



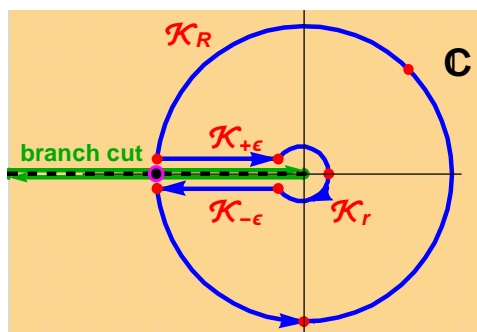
■ **Example 4: Rectangular contour around branch cut $\{0,1\}$**

The contour γ avoids the *branch cut* $\{0,1\}$. The orientation of the (green) branch cut contour is opposite to the orientation of the (blue) contour.



■ **Example 5: “Keyhole contour” in \mathbb{C} excluding branch cut $\{-\infty, 0\}$**

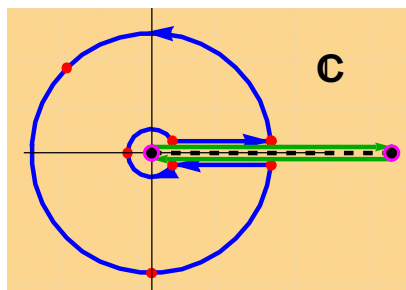
Here, the contour γ is constructed to avoid the branch cut on the negative real axis. Notice, the reverse ordering of points **{7, 6, 5}**, otherwise the circle is flipped. This type of contour avoiding either the negative or positive real axis and encircling the origin is colloquially called “keyhole contour”.



Obviously, the keyhole contour is equivalent to the contour around the branch cut.

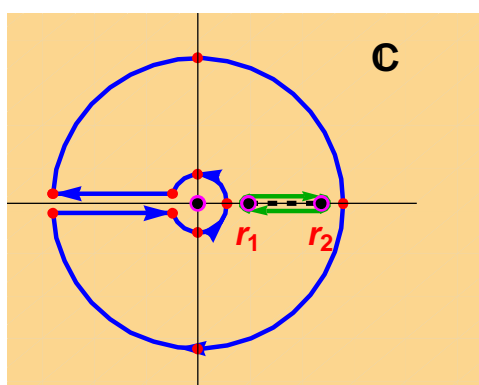
■ **Example 6: “Keyhole contour” in \mathbb{C} excluding branch cut $\{0, +\infty\}$**

Here the contour γ avoids the *branch cut* on the positive real axis, i.e. extending from the origin $\{0,0\}$ to $\{0, R\}$ (with $R \rightarrow \infty$). The origin is clockwise encircled by a small circular arc whereas the orientation of the large circular arc is anti-clockwise.



■ **Example 7: “Keyhole contour” in \mathbb{C} excluding branch cut $\{r_1, r_2\}$**

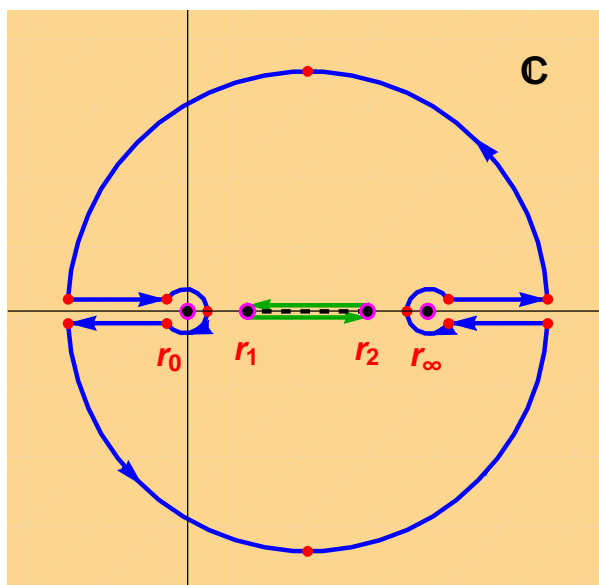
Here, the (green) contour around the *branch cut* (between r_1 and r_2) on the positive real axis is deformed (“blown up”) such that instead the *singularity* in $z_0 = 0$ is encircled counter-clockwise by a small circular arc whereas the branch cut is encircled clockwise by a large circular arc with radius $R \rightarrow \infty$.



■ **Example 8: “Double keyhole contour” in \mathbb{C} excluding branch cut $\{r_1, r_2\}$ in favor of poles $\{r_0, r_\infty\}$**

Action integrals of type $\oint p_r dr = \oint \sqrt{\mathcal{A} + 2\frac{\mathcal{B}}{r} - \frac{\mathcal{C}}{r^2}} dr$ can be casted into the form $\oint \sqrt{1 - \frac{r_1+r_2}{r} + \frac{r_1 r_2}{r^2}} dr = \oint \frac{1}{z} \sqrt{(z-r_1)(z-r_2)} dz$ with the transformation $\{\mathcal{A} \rightarrow 1, \mathcal{B} \rightarrow -(r_1+r_2)/2, \mathcal{C} \rightarrow -r_1 \cdot r_2\}$. There result a *branch cut* between $r_1 < z < r_2$ and two *singular points* at $r_0 = 0$ and $r_\infty \rightarrow \infty$.

The (green) contour around the branch cut can be “blown up” to a *double keyhole contour* to enclose the **singPts** = $\{r_0, r_\infty\}$ instead. Details for the evaluation of the contour integral are discussed in Section 5, example 1.



■ **Example 9: Meandering contour avoiding half-integer poles $\{\frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \dots\}$**

In order to evaluate the integral representation of the Meijer G -function $G_{1,2}^{2,1}\left(2z \mid \begin{smallmatrix} 1/2 \\ 3, -3 \end{smallmatrix}\right)$ a contour path has to be generated which includes only the integer poles $\{\dots, -2, -1, 0, 1, 2, 3\}$ but avoids the half-integral poles $\{\frac{1}{2}, \frac{3}{2}, \frac{5}{2}\}$

- Generation of the set of data points **pts9** defining the open contour path \mathcal{L}

A list of data points **circList** for an alternating sequence of circular arcs is created automatically; starting and end points for the polylines are added.

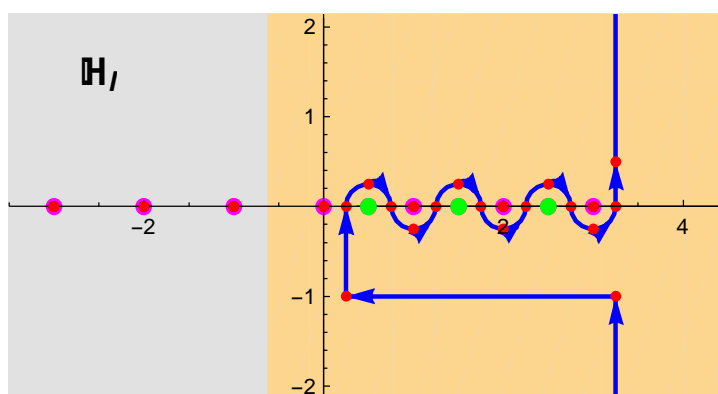
To the last point in **circList**, i.e. $P_e = \left(\frac{13}{4} + 0i\right)$, $i b$ is added : $P_{17} = \left(\frac{13}{4} + b i\right)$.

Similarly, for the prepending points : $P_1 = P_e + a i = \left(\frac{13}{4} + a i\right)$, $P_2 = P_e - i = \left(\frac{13}{4} - i\right)$ and $P_3 = P_a - i = \left(\frac{1}{4} - i\right)$.

In this way the list defining alternating semi-circles is supplemented by two polylines at P_a and P_e . For **a, b** some arbitrary values (e.g. **R = 5**) are assumed which could go to $\mp\infty$ and it is assumed that the contour is closed on the left by an semi-circle with radius $R \rightarrow \infty$.

The integer poles $\{..., -4, -3, -2, -1, 0, 1, 2, 3\}$ and half-integral poles $\{\frac{1}{2}, \frac{3}{2}, \frac{5}{2}\}$ are given as **singPts11** and **singPts22** :

The contour \mathcal{L} goes from $(\frac{13}{4} - i\infty)$ to $(\frac{13}{4} + i\infty)$ and must be closed on the *left side*, i.e. in \mathbb{H}_l because only there the integrand will vanish for $z \rightarrow \infty$. Hence, with this counter-clockwise orientation of the contour \mathcal{L} all singularities with *integer* value (*red* points) from $-n, ..., 0, 1, 2, 3$, are included, *excludes* are, however, the singularities for *half-integral* values $\frac{1}{2}, \frac{3}{2}, \frac{5}{2}, ...$ (green points) :



- Close contour path $\mathcal{L}_{\text{closed}}$ in \mathbb{H}_l

In order to close the contour in the left half-plane \mathbb{H}_l an ancillary point P_{19} on the negative real axis is added to **pts10**. Closing the contour \mathcal{L} by adding a circular arc (with radius $R \rightarrow \infty$) which is defined by **{18, 19, 1}** one obtains for $\mathcal{L}_{\text{close}}$ the following gray-shaded semi-circular region which for radius $R \rightarrow \infty$ finally turns out to be the left half-plane \mathbb{H}_l . All red dots, i.e. the integer poles $\{..., -2, -1, 0, 1, 2, 3\}$, will contribute to the evaluation of the contour integral of the Meijer G -function $G_{1,2}^{2,1}\left(2z \mid \begin{smallmatrix} 1/2 \\ 3, -3 \end{smallmatrix}\right)$.

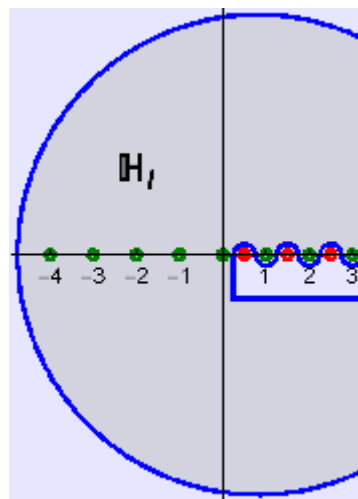
To display the domain inside a closed contour path $\mathcal{L}_{\text{closed}}$ as a gray-shaded area the procedure **generateClosedContourPath** is used.

? **generateClosedContourPath**

ContourIntegration_P2.nb

`generateClosedContourPath[pts,singPts,contour_:All,opts]` generates a closed contour path and returns a pointlist defining the closed contour.

Finally, the closed contour $\mathcal{L}_{\text{closed}}$ is displayed by the procedure **showPointsInContour**; the interior is shown as a gray-shaded area, singular points inside/outside the contour are colored in green/red as already before.



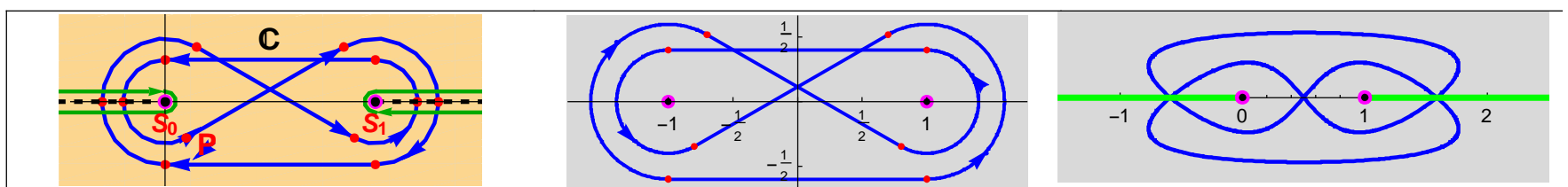
■ **Example 10: Pochhammer double-loop contour γ_{Ph} for Euler's Integral**

For the *Euler integral* $\oint_{\gamma_{\text{Ph}}} t^{a-1}(1-t)^{b-1} dt = -4 e^{i\pi(a+b)} \sin(\pi a) \sin(\pi b) B(a, b)$ with $a, b \in \mathbb{C}$ the *Pochhammer double-loop contour* γ_{Ph} starts from an arbitrary point ($P_1 = (.1 - .173 i)$) in the interval $\{S_0, S_1\} = \{0, +1\}$, encircles S_1 and then S_0 *clockwise*, afterwards encircles S_1 and then S_0 *anti-clockwise*, and returns to P_1 . A short-hand notation for the Pochhammer contour path is $(S_1 +, S_0 +, S_1 -, S_0 -)$ (see e.g. [6]). $B(p, q)$ is the Beta function.

(i) With the procedure `contourPathGraphic` the Pochhammer double-loop contour path can be easily created.

(ii) Another more conventional approach (using arcs, lines, arrows and dots) of the Pochhammer contour path is given by Forst & Hoffmann [7]; the original code was transcribed from Maple to *Mathematica*. Note, the contour path is symmetric with respect to $\{-1, 1\}$ instead of $\{0, 1\}$.

(iii) An elegant representation of the Pochhammer contour path is given by M. Trott [8] in terms of a parametrized path.



■ (2) Points inside/outside a closed Contour

For the evaluation of a contour integral it is important to know which singularities are enclosed by the contour γ . However, in *Mathematica* there is no built-in procedure to decide which of the given singularities are inside and which are outside the closed contour. Therefore this section will deal with the so-called *point-in-polygon problem* which is not only of interest for the evaluation of contour integrals (where the polygon is a closed path γ in the complex plane \mathbb{C}) but was already investigated in computational geometry and in computer graphics. Hence, the problem can be considered as a geometrical one for which two commonly used algorithms are available : one is the *ray crossing algorithm*, the other is the *winding number algorithm* which will be discussed subsequently [9].

■ Ray crossing algorithm

The idea of the *ray crossing algorithm* is to draw a horizontal line starting at a point P in question and extending to infinity. To find out whether point P is located inside or outside a polygon (which must not necessarily be convex) one tests how many times a ray, starting from point P and going in any fixed direction, intersects the edges of the closed polygon. If P is not placed just on the boundary of the polygon, the number of crossings is *even* if P is *outside*, and *odd* if *inside*. Thus this algorithm is therefore also known as *even-odd-rule algorithm* [10,11]. The case of non-convex polygons is a bit more intricate but can be handled using this algorithm too. Moreover, one can omit considering horizontal edges as a fictitious horizontal ray will never cross them. Furthermore, one can also delete those edges which are entirely above or below the y -coordinate of the point $P = (x,y)$.

A very effective implementation of the ray crossing algorithm is the procedure given by P. Wellin [12] (which was slightly improved by the author to make sure that all points inside the closed polygon are colored green whereas points outside are colored red), e.g. see the procedure **PointInPolygonQ**.

```
PointInPolygonQ[poly : {_, _} ..], pt : {x_, y_}, onoff_ : "Off"] :=
  Module[{tri, e1, e2, e3, e4, x1, y1, x2, y2},
    (* ---- Ref : Paul Wellin "Programming with Mathematica, An Introduction" (2013), Chapt. 10.4 pp. 419ff ----- *)
    TriangleArea[tri : {v1_, v2_, v3_}] := Det[Map[PadRight[#, 3, 1] &, tri]]/2;
    Clear[e1, e2, e3, e4];

    e1 = Partition[poly, 2, 1, 1];
    e2 = DeleteCases[e1, {{x1_, y1_}, {x2_, y2_}} /; y1 == y2]; (* eliminate horizontal edges *)
```

```

e3 = DeleteCases[e2, {{x1_, y1_}, {x2_, y2_}} /; (Min[ y1, y2 ] ≥ y || Max[ y1, y2 ] < y)];
(* delete edges entirely above/below y-coord of pt *)

e4 = Map[Reverse @ SortBy[#, Last] &, e3 ];
boole = OddQ[Count[TriangleArea[Join[#, {pt}]]] & /@ e4, _?Positive]];
print[onoff, " pt(x,y)= ", {x, y}, " ⇒ (T|F)= ", boole ];
Return[boole];

];

```

Here is a test for a rectangular contour γ and several test points

? PointInPolygonQ

PointInPolygonQ[poly:{{_,_}..},pt:{x_,y_},onoff_:"Off"] tests whether points 'pt' are located inside or outside the closed polygonal contour 'poly'.
A list with boolean variables {True|False} is returned if the points lie inside (True) or outside respectively on the contour (False).

```

γ = {{1, 1/2}, {-1, 1/2}, {-1, -1/2}, {1, -1/2}, {1, 1/2}}; (* contour *)
PointInPolygonQ[γ, {0, .2}, "Off" ]
PointInPolygonQ[γ, #] & /@ {{0, .2}, {0, .6}, {- .5, -.3}, {.5, .5 + 10-16}}

True

{True, False, True, True}

```

■ Winding number algorithm

The underlying idea of the *winding number algorithm* is to compute for point P its winding number w.r.t. the polygon [13]. If the winding number is non-zero, then P lies inside the polygon otherwise P is located outside or on the boundary of the closed polygon. One way computing the winding number is to sum up the angles extended by each side of the polygon. Denoting by V_i the set of N vertices defining the polygon one calculates the following sum $S = \sum_{i=1}^{N-1} (\angle(V_{i+1} - P, V_i - P) + \angle(V_N - P, V_1 - P))$. If $S \neq 0$ then P is *inside* the polygon. If $S == 0$ then P is *outside* with angle $\angle(a,b)$ returning a value in the interval $(-\pi, \pi]$.

An efficient implementation of this approach is given by M. Trott [14] with the procedure **PointInPolygonQ2**.

? PointInPolygonQ2

```
PointInPolygonQ2[poly_, pt_, onoff_: "Off"] := insidePolygonQCF[poly][pt];

insidePolygonQCF[slist_] := insidePolygonQCF[slist] =
Module[{segmentList, segmentListLength},
(* ----- © Michael Trott (Oct. 2013) ----- *)
segmentList = Partition[If[Last[slist] === First[slist], slist, Append[slist, slist[[1]]], 2, 1];
segmentListLength = Length[segmentList];
Function[{segments, λ},
Compile[{{viewPoint, _Real, 1}},
Module[{sum = 0., px = viewPoint[[1]], py = viewPoint[[2]], q1x, q1y, q2x, q2y},
Do[{q1x, q1y}, {q2x, q2y}] = segments[[k]];
sum = sum + ArcTan[(q1x - px) (q2x - px) + (q1y - py) (q2y - py),
(q1y - py) (q2x - px) - (q1x - px) (q2y - py)], {k, λ}];
boole = Round[sum] != 0]
]
][segmentList, segmentListLength]
];
```

Again, here is a test for a rectangular contour with a list of points $\{\{0, .2\}, \{0, .6\}, \{-.5, -.3\}, \{.5, .5\}\}$; the second point lies outside, the forth point is slightly above the boundary.

```

γ = {{1, 1/2}, {-1, 1/2}, {-1, -1/2}, {1, -1/2}, {1, 1/2}};      (* contour *)
PointInPolygonQ2[γ, {0, .2}, "Off"]
PointInPolygonQ2[γ, #] & /@ {{0, .2}, {0, .6}, {- .5, - .3}, {.5, .5 + 10-16}}

True

{True, False, True, False}

```

See also the implementation of the winding number approach by David Park [15] using the procedure **WindingNumber** which applies **SegmentCrossingIncrement**.

Another variant of the winding number algorithm is to compute the winding number by integrating $\mathbf{1}/z$ (centered at point $P : z_p = x_p + i y_p$) along the polygonal contour γ in the complex plane. Applying Cauchy's Residue Theorem [16] may not be very efficient in comparison to other approaches given above but still this method is in the context of contour integration $n(z_p, \gamma) = \frac{1}{2\pi i} \oint_{\gamma} \frac{1}{z - z_p} dz$ because complex (numerical) integration is feasible in *Mathematica*.

Another implementation with **PointInPolygonQ3** was suggested by T. Heidecke [17] and similarly by R.Brambilla [18] following an idea of Muskhelishvili.

? **PointInPolygonQ3**

```

PointInPolygonQ3[poly_, pt_, onoff_: "Off"] := InsidePolygonQ[poly, pt, onoff]
InsidePolygonQ[polygon_, point_, onoff_: "Off"] :=
Module[{wn, boole},
  wn = WindingNumber[polygon, point, onoff];
  If[wn === 1, boole = True, boole = False];
  Return[boole];
];
WindingNumber[contour_, point_, onoff_: "Off"] :=

```

ContourIntegration_P2.nb

```
Module[{wn, zp},
(* ----- © Thies Heidecke, Mathematica Stack Exchange (Aug 2012) ----- *)
  zp = Complex @@ point;
  Off[NIntegrate::ncvb, NIntegrate::slwcon];
  ζRange = Evaluate @ {ζ, Sequence @@ (Complex @@@ Append[#, #[[1]] ] & [contour])};
  print[onoff, " point zp= ", zp, "; zRange: ", ζRange ];
  wn = Round @ Re @ Chop[  $\frac{1}{2\pi i}$  NIntegrate[  $\frac{1}{\zeta - z_p}$ , ζRange ] ];
  On[NIntegrate::ncvb, NIntegrate::slwcon];
  Return[wn]
];
```

```
contour = {{1,  $\frac{1}{2}$ }, {-1,  $\frac{1}{2}$ }, {-1,  $-\frac{1}{2}$ }, {1,  $-\frac{1}{2}$ }, {1,  $\frac{1}{2}$ }}; (* contour *)
PointInPolygonQ3[contour, {0, .2}]
PointInPolygonQ3[contour, #, "Off"] & /@ {{0, .2}, {0, .6}, {-.5, -.3}, {.5, .5 + 10-11}}

True

{True, False, True, False}
```

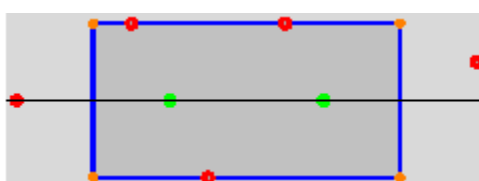
Incorporated in the package **ContourIntegration** is the procedure

? PointInPolygonQ

PointInPolygonQ[poly:{{_,_}..},pt:{x_,y_},onoff_:"Off"] tests whether points 'pt' are located inside or outside the closed polygonal contour 'poly'.
A list with boolean variables {True|False} is returned if the points lie inside (True) or outside respectively on the contour (False).

■ 2.1 Rectangular contour

The simplest test is a *rectangular polygon* for which points inside/outside are shown in green/red.



The graphics showing the points colored in green or red according to their location either inside or outside the closed contour is achieved by the procedure **showPointsInContour**

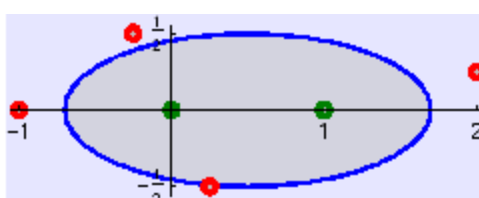
? **showPointsInContour**

`showPointsInContour[poly,ptsInOut,onoff_:"Off",opts___]` draws for a given closed contour 'poly' (which is a polyline) and a number of points 'ptsInOut' a graphics which shows the contour (in blue). Those points located inside the contour are given in green; those points outside or on the contour are shown in red. The decision whether points are inside or outside the closed contour is made using `PointInPolygonQ`. The global variable `$Pts = {inPts,outPts}` contains two distinct sets of points being in/outside the contour.

■ 2.2 Elliptical contour

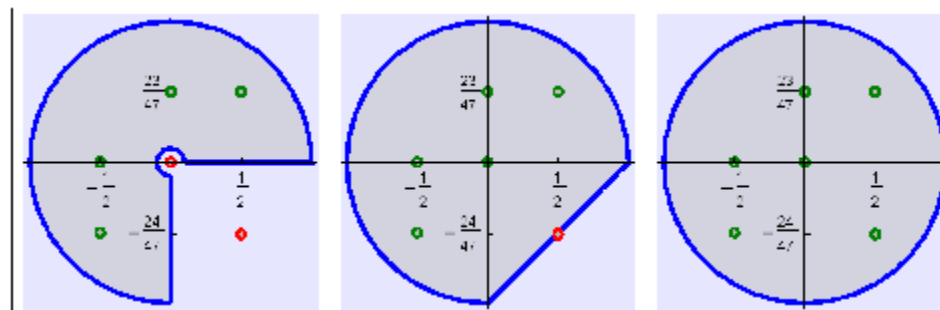
The next example is an *elliptical region* (which is represented as a polyline).

The necessary steps of the previous example can be combined into a single procedure **showPointsInContour**



■ 2.3 Circular arc contour

Points in several circular arc contours



■ 2.4 Closed contour for Meijer G-function

Closed contour for the Meijer G-function $G_{1,2}^{2,1}\left(2z \mid \begin{smallmatrix} 1/2 \\ 3, -3 \end{smallmatrix}\right)$

Here, the task is to generate a contour which meanders around integer poles at $\{-4, -3, -2, -1, 0, 1, 2, 3\}$ and avoids half-integral poles at $\{\frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \dots\}$ and is closed in the left half-plane \mathbb{H}_l . In order to close the contour in \mathbb{H}_l an ancillary point P_{16} on the negative real axis must be added to the point list given. Thus the contour \mathcal{L} is supplemented by a circular arc (with radius $R \rightarrow \infty$) which is defined by $\{15, 16, 1\}$. Finally, the resulting closed contour $\mathcal{L}_{\text{close}}$ will become the following gray-shaded semi-circular region which as $R \rightarrow \infty$ turns out to be the left half-plane \mathbb{H}_l . In this way all green dots, i.e. the integer poles $\{\dots, -4, -3, -2, -1, 0, 1, 2, 3\}$, will contribute to the contour integral determining the Meijer G-function $G_{1,2}^{2,1}\left(2z \mid \begin{smallmatrix} 1/2 \\ 3, -3 \end{smallmatrix}\right)$. The corresponding procedure for closing a given contour is

? **generateClosedContourPath**

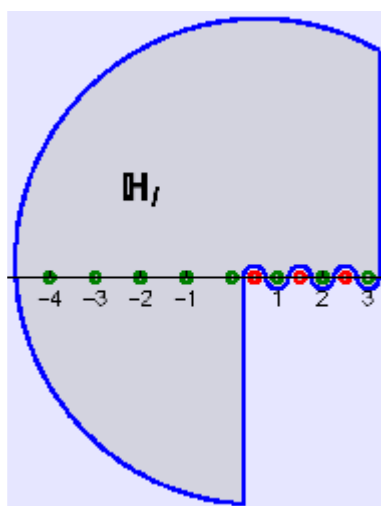
`generateClosedContourPath[pts,singPts,contour_:All,opts]` generates a closed contour path and returns a pointlist defining the closed contour.

The procedure **generateClosedContourPath** generates a closed contour in the form of a polyline.

The variable **closedPolylinePts** to be returned contains a pointlist which allows further processing of the closed contour such as *gray-shading* of

the interior domain etc. .

Finally, the closed contour $\mathcal{L}_{\text{closed}}$ is displayed by the procedure **showPointsInContour**; the interior is shown as a gray-shaded area, singular points *inside/outside* the contour are colored in *green/red* as before.



■ (3) Integral representations of Meijer G-function : $G_{p,q}^{m,n} \left(z \left| \begin{matrix} a_1, \dots, a_p \\ b_1, \dots, b_q \end{matrix} \right. \right)$

As already mentioned above many special functions are given in terms of integral representations which involve contour integrals. Below, as example for integral representations of special functions the Meijer G-function will be studied in detail.

As given e.g. in [4] *The Wolfram Functions Site* with URL <http://functions.wolfram.com/> the Meijer G-function can be defined by means of the following contour integral

$$G_{p,q}^{m,n} \left(z \left| \begin{matrix} a_1, \dots, a_n, a_{n+1}, \dots, a_p \\ b_1, \dots, b_m, b_{m+1}, \dots, b_q \end{matrix} \right. \right) = \frac{1}{2\pi i} \oint_{\mathcal{L}} \frac{\prod_{j=1}^m \Gamma(b_j + s) \prod_{j=1}^n \Gamma(1 - a_j - s)}{\prod_{j=n+1}^p \Gamma(a_j + s) \prod_{j=m+1}^q \Gamma(1 - b_j - s)} z^{-s} ds \quad ; \quad \{m, n, p, q\} \in \mathbb{N} \wedge m \leq q \wedge n \leq p$$

The Meijer G-function is given through the Mellin-Barnes integral representation where the Γ 's denote the usual Gamma functions. The contour \mathcal{L} separates the poles of the products of Γ -functions in the numerator, such as $\Gamma(b_j + s)$ with $(j = 1, \dots, m)$ from those of $\Gamma(1 - a_j - s)$ with $(j = 1, \dots, n)$ in a sophisticated way. There are three possibilities for choosing the contour \mathcal{L} (for details see **Notations.nb** of the Wolfram Functions Site [19]). The *Mathematica* implementation follows the definition suggested by Prudnikov et al. (1990) [20].

? MeijerG

MeijerG[{{a₁, ..., a_n}, {a_{n+1}, ..., a_p}}, {{b₁, ..., b_m}, {b_{m+1}, ..., b_q}}, z] is the Meijer G function $G_{p,q}^{m,n} \left(z \left| \begin{matrix} a_1, \dots, a_p \\ b_1, \dots, b_q \end{matrix} \right. \right)$. >

From the input form **MeijerG**[{{a₁, ..., a_n}, {a_{n+1}, ..., a_p}}, {{b₁, ..., b_m}, {b_{m+1}, ..., b_q}}, z] the traditional form which is found in literature $G_{p,q}^{m,n} \left(z \left| \begin{matrix} a_1, \dots, a_p \\ b_1, \dots, b_q \end{matrix} \right. \right)$ is obtained e.g.

MeijerG[{{a, b}, {c}}, {{d}, {e}}, z] // hF // tF

$$G_{3,2}^{1,2}\left(z \left| \begin{matrix} a, b, c \\ d, e \end{matrix} \right. \right)$$

Here, **hF** and **tF** are *shortcuts* for the procedures **HoldForm** and **TraditionalForm**.

□ 3 Examples of special functions' representation in terms of MeijerG functions

In many cases Meijer G-functions reduce to simpler special functions. A list where Meijer G-functions are reduced to simpler special functions is given in Appendix 2 : Representation of special functions in terms of Meijer G-functions [19] . Here three cases will be discussed in some detail. ,

■ **Example 3.1** $G_{1,2}^{2,1}\left(2z \left| \begin{matrix} 1/2 \\ 3, -3 \end{matrix} \right. \right) \Leftrightarrow \text{Bessel function } K_3(z) \Rightarrow f_{2112}(z, s) = 2^{-s} z^{-s} \Gamma\left(\frac{1}{2} - s\right) \Gamma(s-3) \Gamma(s+3)$

In this example the calculation of the contour integral is done step by step in order to show the approach. Note that for given $G_{1,2}^{2,1}$ the numerator simplifies to 1 due to the empty products $\prod_{j=1}^{p=1}$ sowie $\prod_{j=2}^{q=2}$ (m=2, n=1) and the MeijerG function simplifies to the Bessel function $K_3(z)$.

$$\text{MeijerG}\left[\left\{\left\{\frac{1}{2}\right\}, \{\}\right\}, \{\{3, -3\}, \{\}\}, 2z\right] // \text{tF}$$

$$\sqrt{\pi} (-e^z) K_3(z)$$

Step 1 : Find and visualize the singularities for the integrand f_{2112}

The *integrand* is extracted by application of **SlaterForm** to the MeijerG function given.

$$(f_{2112}[z_, s_] = \text{System`MeijerGDump`SlaterForm}[\text{MeijerG}[\{\{1/2\}, \{\}\}, \{\{3, -3\}, \{\}\}, 2z], s]) // \text{tF}$$

$$2^{-s} z^{-s} \Gamma\left(\frac{1}{2} - s\right) \Gamma(s-3) \Gamma(s+3)$$

With the help of procedure **findSingularities4SpecFunc** the integer-valued singularities of f_{2112} are determined in the range $s = \{-10, 4\}$.

? findSingularities4SpecFct

findSingularities4SpecFct[specFct,s,polesRange,onoff_:"On"] determines the singularities for special functions f(s)='specFct' (such as MeijerG functions, which do not have an obvious denominator for poles). The range of the singularities is confined by 'polesRange'; the singularities found are returned in the global variable '\$poles' in order to be displayed in contourPathGraphics. The global variable '\$polesType' with values 'Integer|HalfInteger|All' selects the poles correspondingly.

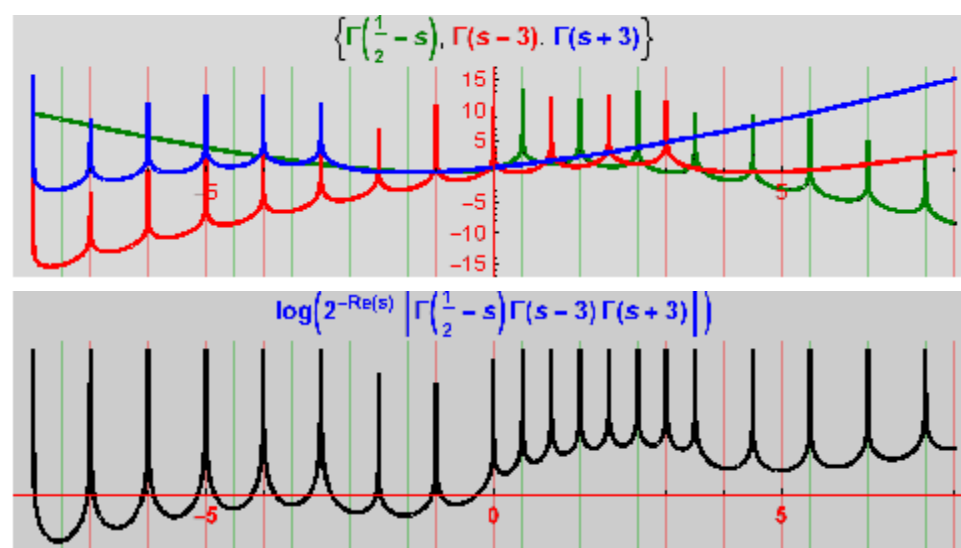
```
$polesType = Integer; (* only integer poles are considered *)
$onoff = True;
findSingularities4SpecFct[f2112[z, s], s, {-10, 4}, "On"];
$poles; (* poles determined by findSingularities *)
```

```
$polesType => Integer poles : {-10, -10, -9, -9, -8, -8, -7, -7, -6, -6, -5, -5, -4, -4, -3, -3, -2, -1, 0, 1, 2, 3}
```

```
Γ-functions product F(s) =  $\Gamma\left(\frac{1}{2} - s\right) \Gamma(s - 3) \Gamma(s + 3)$ 
      has poles : {-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3}
with multiplicity : {2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1}
```

In the first plot the singularities are visualized for each Γ -function of the integrand separately; the second plot shows the superposition of all singularities in the range $s = \{-8, 8\}$. Discarding the half-integer singularities (which originate from $\Gamma\left(\frac{1}{2} - s\right)$) for the calculation of residues as regards to convergence in the left-hand complex plane $s \in \mathbb{C}$ it should be pointed out that for $s \leq -3$ double poles occur whereas the poles $s = \{-2, -1, 0, 1, 2, 3\}$ are simple poles.

```
z0 = 1;
visualizeSingularities[f2112[z, s], z0, {-8, 8}]
```



? VisualizeSingularities

VisualizeSingularities[specFctInt, z0, polesRange] displays the singularities of the integrand $f(z,s) = \text{'specFctInt'}$

for a given special function which is defined as a contour integral $\frac{1}{2\pi i} \oint_{\mathcal{L}} f_{\text{mnpq}}(z, s) ds$. The first graph shows (with different colors for each function component) the singularities given as $\text{Log[Abs[specFctList]]}$ in the range 'polesRange'. The second graph shows the superposition of all singularities of the special function investigated at given 'z0'.

Step 2 : Calculate residues for given singularities

The residues for given integer singularities are evaluated taking into account multiplicity of poles. The result in terms of a truncated series is returned in the variable J_{2112}

? calculateResidues4SpecFct

calculateResidues4SpecFct[f, s, selectPoles_:None, polesRange_:{ }, onoff_:"Off"] evaluates the residues symbolically. 'f' denotes the special function (e.g. MeijerG-function) f(s) with 's' as complex variable $s \in \mathbb{C}$. The singularities of f(s) are determined by calling findSingularities within the range given by 'polesRange'={a,b}; the list of lists {poles, μ ,solK} is returned where 'poles' contains the list of singularities, ' μ ' is a list containing the multiplicity of every pole and solK = 0 is set. For special functions the parameter 'selectPoles'= None must be chosen. The global variable \$polesType = Integer|HalfInteger|All selects the type of poles to be considered. The sum of residues for the singularities selected is calculated and returned. With onoff= "No" printout is suppressed.

```
J2112 = calculateResidues4SpecFct[f2112[z, s], s, None, {-10, 4}, "Off"];
```

Γ -functions product $F(s) = \Gamma\left(\frac{1}{2} - s\right) \Gamma(s - 3) \Gamma(s + 3)$
has poles : { -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3 }
with multiplicity : { 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1 }

processing residue $\text{Res}\left[2^{-s} z^{-s} \Gamma\left(\frac{1}{2} - s\right) \Gamma(s - 3) \Gamma(s + 3), \{s, -10\}\right]$ for 2-fold pole(1)= -10

⋮

processing residue $\text{Res}\left[2^{-s} z^{-s} \Gamma\left(\frac{1}{2} - s\right) \Gamma(s - 3) \Gamma(s + 3), \{s, -2\}\right]$ for 1-fold pole(9)= -2

⋮

processing residue $\text{Res}\left[2^{-s} z^{-s} \Gamma\left(\frac{1}{2} - s\right) \Gamma(s - 3) \Gamma(s + 3), \{s, 3\}\right]$ for 1-fold pole(14)= 3

$$\begin{aligned}
& \text{Residues : } \sum_j \text{Res} \left[2^{-s} z^{-s} \Gamma \left(\frac{1}{2} - s \right) \Gamma(s-3) \Gamma(s+3) \right] |_{s=s_j} = \\
& - \frac{323 \sqrt{\pi} \left(72072 \log(z) + 72072 \psi^{(0)} \left(\frac{21}{2} \right) + 72072 \log(2) + 144144 \gamma - 416071 \right) z^{10}}{1115882127360} - \frac{221 \sqrt{\pi} \left(5544 \log(z) + 5544 \psi^{(0)} \left(\frac{19}{2} \right) + 5544 \log(2) + 11088 \gamma - 30787 \right) z^9}{12262440960} - \\
& - \frac{13 \sqrt{\pi} \left(5544 \log(z) + 5544 \psi^{(0)} \left(\frac{17}{2} \right) + 5544 \log(2) + 11088 \gamma - 29401 \right) z^8}{170311680} - \frac{143 \sqrt{\pi} \left(2520 \log(z) + 2520 \psi^{(0)} \left(\frac{15}{2} \right) + 2520 \log(2) + 5040 \gamma - 12631 \right) z^7}{232243200} - \\
& - \frac{11 \sqrt{\pi} \left(2520 \log(z) + 2520 \psi^{(0)} \left(\frac{13}{2} \right) + 2520 \log(2) + 5040 \gamma - 11749 \right) z^6}{5806080} - \frac{3 \sqrt{\pi} \left(280 \log(z) + 280 \psi^{(0)} \left(\frac{11}{2} \right) + 280 \log(2) + 560 \gamma - 1181 \right) z^5}{71680} - \\
& - \frac{\sqrt{\pi} \left(140 \log(z) + 140 \psi^{(0)} \left(\frac{9}{2} \right) + 140 \log(2) + 280 \gamma - 503 \right) z^4}{6720} - \frac{1}{960} \sqrt{\pi} \left(20 \log(z) + 20 \psi^{(0)} \left(\frac{7}{2} \right) + 20 \log(2) + 40 \gamma - 49 \right) z^3 - \\
& - \frac{\sqrt{\pi} z^2}{40} + \frac{\sqrt{\pi} z}{24} - \frac{\sqrt{\pi}}{3} - \frac{3 \sqrt{\pi}}{z} - \frac{8 \sqrt{\pi}}{z^2} - \frac{8 \sqrt{\pi}}{z^3} \text{ for poles } s = \{-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3\}
\end{aligned}$$

Step 3 : Numerical comparison of the results

Here, the *approximate* result obtained from the finite sum of residues will be compared with the *exact* result of the Meijer G-function.

ContourIntegration_P2.nb

serJ is the finite series of powers $\{z^3, z^4, \dots, z^{10}\}$ (originating from the poles $s = -3, \dots, -10$)

(serJ = (Level[(J2112 // tF // polyForm), 3, List] // Drop[#, 6] &) // Drop[#, - 5] &) // tF

$$\left\{ -\frac{1}{960} \sqrt{\pi} z^3 \left(20 \log(z) + 40 \gamma - 49 + 20 \log(2) + 20 \psi^{(0)}\left(\frac{7}{2}\right) \right), \right. \\ -\frac{\sqrt{\pi} z^4 \left(140 \log(z) + 280 \gamma - 503 + 140 \log(2) + 140 \psi^{(0)}\left(\frac{9}{2}\right) \right)}{6720}, -\frac{3 \sqrt{\pi} z^5 \left(280 \log(z) + 560 \gamma - 1181 + 280 \log(2) + 280 \psi^{(0)}\left(\frac{11}{2}\right) \right)}{71680}, \\ -\frac{11 \sqrt{\pi} z^6 \left(2520 \log(z) + 5040 \gamma - 11749 + 2520 \log(2) + 2520 \psi^{(0)}\left(\frac{13}{2}\right) \right)}{5806080}, -\frac{143 \sqrt{\pi} z^7 \left(2520 \log(z) + 5040 \gamma - 12631 + 2520 \log(2) + 2520 \psi^{(0)}\left(\frac{15}{2}\right) \right)}{232243200}, \\ -\frac{13 \sqrt{\pi} z^8 \left(5544 \log(z) + 11088 \gamma - 29401 + 5544 \log(2) + 5544 \psi^{(0)}\left(\frac{17}{2}\right) \right)}{170311680}, -\frac{221 \sqrt{\pi} z^9 \left(5544 \log(z) + 11088 \gamma - 30787 + 5544 \log(2) + 5544 \psi^{(0)}\left(\frac{19}{2}\right) \right)}{12262440960}, \\ \left. -\frac{323 \sqrt{\pi} z^{10} \left(72072 \log(z) + 144144 \gamma - 416071 + 72072 \log(2) + 72072 \psi^{(0)}\left(\frac{21}{2}\right) \right)}{1115882127360} \right\}$$

Similarly, the corresponding terms from the exact series expansion of Meijer G-function are

serM1 = ((Level[Series[MeijerG[{{{\frac{1}{2}}}, {}]}, {{3, -3}, {}]}, 2 z], {z, 0, 10}] // fs // Normal, 1, List] // Drop[#, 1] &) // Drop[#, 5] &) // Reverse; (* z^3, ..., z^10 *)

c8 = serM1[[5]; c7 = serM1[[6]; (* interchange terms 5 and 6 *)

(serM = serM1 /. {serM1[[5] → c7, serM1[[6] → c8}) // tF

$$\left\{ -\frac{\sqrt{\pi} z^3 (60 \log(z) + 60 \gamma + 37 - 60 \log(2))}{2880}, \frac{\sqrt{\pi} z^4 (-420 \log(z) - 420 \gamma + 101 + 420 \log(2))}{20160}, \right. \\ \frac{\sqrt{\pi} z^5 (-2520 \log(z) - 2520 \gamma + 1621 + 2520 \log(2))}{215040}, \frac{\sqrt{\pi} z^6 (-27720 \log(z) - 27720 \gamma + 25111 + 27720 \log(2))}{5806080}, \\ \frac{\sqrt{\pi} z^7 (-360360 \log(z) - 360360 \gamma + 397129 + 360360 \log(2))}{232243200}, \frac{\sqrt{\pi} z^8 (-360360 \log(z) - 360360 \gamma + 453913 + 360360 \log(2))}{851558400}, \\ \left. \frac{\sqrt{\pi} z^9 (-6126120 \log(z) - 6126120 \gamma + 8527331 + 6126120 \log(2))}{61312204800}, \frac{\sqrt{\pi} z^{10} (-16628040 \log(z) - 16628040 \gamma + 25049807 + 16628040 \log(2))}{797058662400} \right\}$$

Comparison of individual terms (as regards to powers $z^3 \dots z^{10}$) from the series expansion of $\sqrt{\pi} (-e^z) K_3(z)$ with the corresponding terms of the sum of residues \mathbf{J}_{2112} shows numerical agreement e.g. for $\mathbf{z} = \mathbf{2.}$:

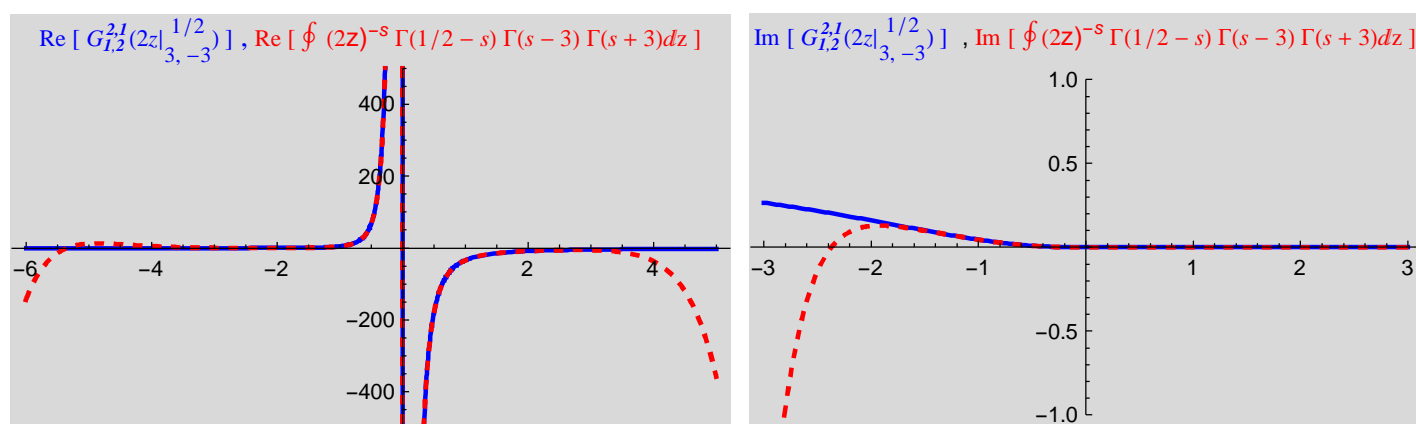
Table[{serJ[[i]], serM[[i]]} /. {z -> 2.}, {i, 1, 8}] // cF

```
{-0.352684, -0.352684}
{-0.198952, -0.198952}
{0.0438937, 0.0438937}
{0.177999, 0.177999}
{0.184751, 0.184751}
{0.13103, 0.13103}
{0.0738765, 0.0738765}
{0.0351856, 0.0351856}
```

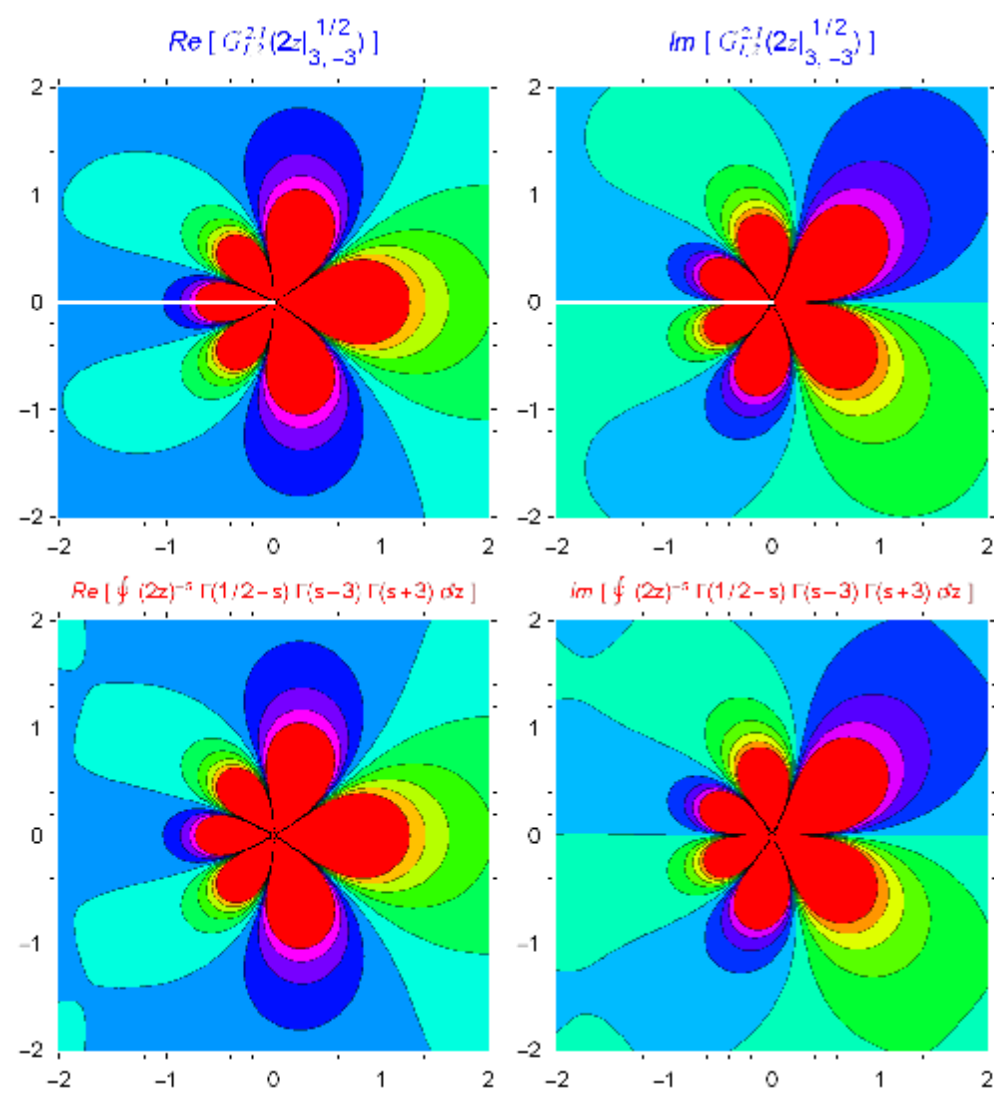
Step 4 : Plotting the results

These are the plots of the real and imaginary part of the *exact* Meijer G-function $G_{I,2}^{2,I} \left(2z \left| \begin{smallmatrix} 1/2 \\ 3, -3 \end{smallmatrix} \right. \right)$ and the *approximate* evaluation of the contour integral $\oint (2z)^{-s} \Gamma(1/2-s) \Gamma(s-3) \Gamma(s+3) dz$ for poles from $\mathbf{s} = \{-10, \dots, 3\}$ considered. Obviously, there is good agreement in the range from $\mathbf{z} = \{-2, 3\}$; taking into account more (integer) poles $s < -10$ will increase the accuracy of the approximation.

ContourIntegration_P2.nb



The contour plots of the real and imaginary part of the *exact* Meijer G-function $G_{I,2}^{2,I}\left(2z \left| \begin{matrix} 1/2 \\ 3, -3 \end{matrix} \right. \right)$ and the *approximate* evaluation of the contour integral $\oint (2z)^{-s} \Gamma(1/2-s) \Gamma(s-3) \Gamma(s+3) dz$ for poles from $s = \{-10, \dots, 3\}$ show qualitative agreement in the range $-2 < x, y < 2$. Note, that the *branch cut* on the negative real axis for the Meijer G-function is missing for the approximate function.



■ **Example 3.2** $G_{2,2}^{1,2}\left(z \left| \begin{matrix} 1, 1 \\ 1, 1 \end{matrix} \right. \right) \Leftrightarrow$ Rational function $\frac{z}{z+1} \Rightarrow f_{1222}(z, s) = z^{-s} \Gamma(-s) \Gamma(s+1)$

ContourIntegration_P2.nb

The second example demonstrates that this particular Meijer G-function reduces to a simple rational function :

`MeijerG[{{1, 1}, {}}, {{1}, {1}}, z]`

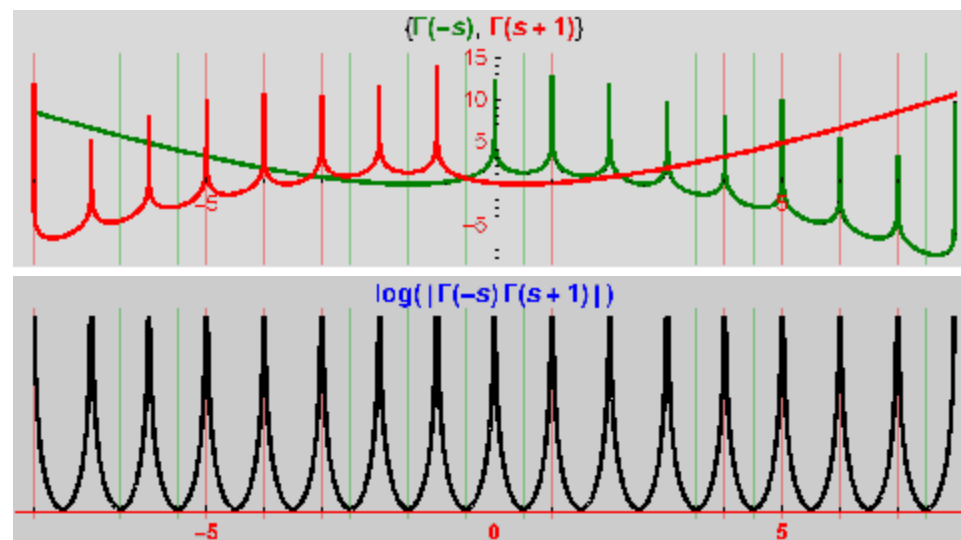
$$\frac{z}{1+z}$$

The corresponding integrand is given as f_{1222}

`(f1222[z_, s_] = System`MeijerGDump`SlaterForm[MeijerG[{{1, 1}, {}}, {{1}, {1}}, z], s]) // tF`

$$z^{-s} \Gamma(-s) \Gamma(s+1)$$

Because of the Γ -functions $\Gamma(-s)\Gamma(s+1)$ there occur integer poles only in the range $s = \{-8, 8\}$ with multiplicity $\mu=1$.



Instead of doing a stepwise approach the procedure `ContourIntegral` (with the variable `selectPoles= SpecFct`) will calculate the contour integral directly.

Note, that in **ContourIntegral** the sum of residues is multiplied by the factor $(2\pi i)$; hence, in order to compare the results obtained with the those from **calculateResidues4SpecFct** division by $(2\pi i)^{-1}$ is required.

```
(J1222 = ContourIntegral[f1222[z, s], s, SpecFct, {-9, 9}, "No"] / (2 π i));
```

$$\text{Contour integral } \mathcal{J} = \oint z^{-s} \Gamma(-s) \Gamma(s+1) ds = 2 i \pi \left(z^9 - z^8 + z^7 - z^6 + z^5 - z^4 + z^3 - z^2 + z - 1 + \frac{1}{z} - \frac{1}{z^2} + \frac{1}{z^3} - \frac{1}{z^4} + \frac{1}{z^5} - \frac{1}{z^6} + \frac{1}{z^7} - \frac{1}{z^8} + \frac{1}{z^9} \right)$$

for 'Integer' poles in the range $s = \{-9, 9\}$

Reconstruction of the infinite series from the truncated series expansion.

The evaluation of the contour integration (where only a finite number of poles is taken into account) leads to a *truncated alternating series* which agrees with the series expansion of $\frac{z}{1+z}$ for $z_0 = 0$.

```
Series[ $\frac{z}{1+z}$ , {z, z0 = 0, 9}] // Normal
```

$$z - z^2 + z^3 - z^4 + z^5 - z^6 + z^7 - z^8 + z^9$$

By means of the *Mathematica* procedure **FindSequenceFunction** the infinite series can formally be reconstructed :

```
(J1222 // polyForm);
```

```
res = Table[-(-z)n, {n, -9, 9}]
```

```
c1222 = FindSequenceFunction[res /. {z → 1}, k]
```

$$\left\{ \frac{1}{z^9}, -\frac{1}{z^8}, \frac{1}{z^7}, -\frac{1}{z^6}, \frac{1}{z^5}, -\frac{1}{z^4}, \frac{1}{z^3}, -\frac{1}{z^2}, \frac{1}{z}, -1, z, -z^2, z^3, -z^4, z^5, -z^6, z^7, -z^8, z^9 \right\}$$

$$(-1)^{1+k}$$

Finally, deduction of the *infinite* series $\mathcal{P}_\infty(z)$ is in agreement with the result following from **MeijerG**[$\{\{1,1\},\{\}\},\{\{1\},\{1\}\},z]$.

$$\mathcal{P}[n_, z_] := \sum_{k=1}^n c_{1222} z^k; \quad \mathcal{P}[\infty, z]$$

ContourIntegration_P2.nb

$$\frac{z}{1+z}$$

■ **Example 3.3** $G_{0,1}^{1,0}(z | 1-a) \iff \text{Exponential \& power function } e^{-z} z^{1-a} \Rightarrow f_{1001}(z, s) = z^{-s} \Gamma(s-1) \text{ with } a=2$

Regarding the third example, the Meijer G-function reduces to $e^{-z} \cdot z^{1-a}$:

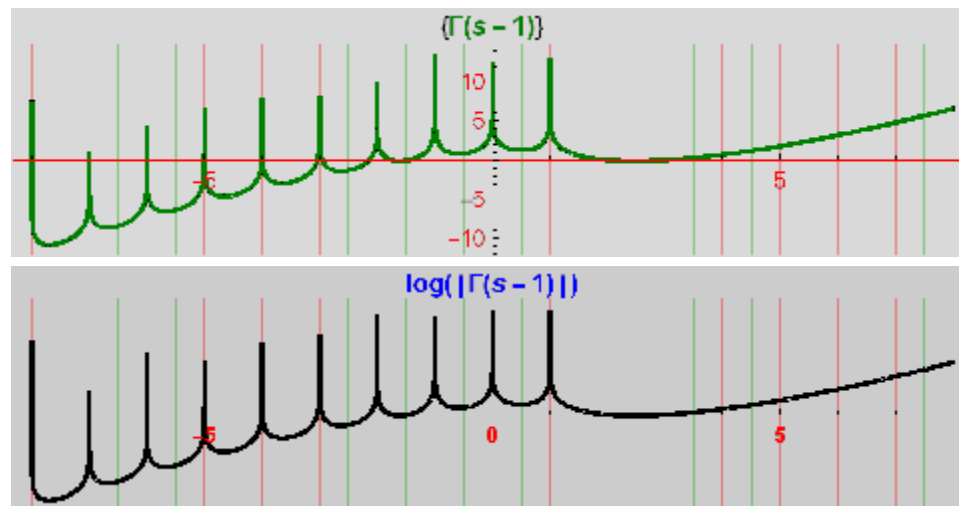
```
MeijerG[{{}, {}}, {{1-a}, {}}, z] // tF
```

$$e^{-z} z^{1-a}$$

The integrand f_{1001} consists of one Γ -function only; for the parameter $a=2$ is chosen. For $s \leq 1$ there occur integer poles with multiplicity $\mu=1$.

```
(f1001[z_, s_] = System`MeijerGDump`SlaterForm[ MeijerG[{{}, {}}, {{-1}, {}}, z, s]) // tF
```

$$z^{-s} \Gamma(s-1)$$



```
(J1001 = ContourIntegral[f1001[z, s], s, SpecFct, {-9, 2}, "No"] / (2 π i));
```

$$\text{Contour integral } \mathcal{I} = \oint z^{-s} \Gamma(s-1) ds = 2i\pi \left(\frac{z^9}{3\,628\,800} - \frac{z^8}{362\,880} + \frac{z^7}{40\,320} - \frac{z^6}{5040} + \frac{z^5}{720} - \frac{z^4}{120} + \frac{z^3}{24} - \frac{z^2}{6} + \frac{z}{2} - 1 + \frac{1}{z} \right) \text{ for 'Integer' poles in the range } s = \{-9, 2\}$$

Reconstruction of the infinite series from the *truncated* series expansion

The evaluation of the contour integration leads again to a truncated alternating series which agrees with the series expansion of $e^{-z} z^{-1}$

Series $[e^{-z} z^{-1}, \{z, 0, 9\}]$ // **Normal** // **polyForm**

$$\frac{z^9}{3\,628\,800} - \frac{z^8}{362\,880} + \frac{z^7}{40\,320} - \frac{z^6}{5040} + \frac{z^5}{720} - \frac{z^4}{120} + \frac{z^3}{24} - \frac{z^2}{6} + \frac{z}{2} - 1 + \frac{1}{z}$$

From the 11 terms of the *truncated* series the associated infinite series can be reconstructed :

series = **(z * J₁₀₀₁ // Expand) /. {Plus → List} /. {z → 1}**
(c₁₀₀₁ = FindSequenceFunction[series, k]) // tF

$$\left\{ 1, -1, \frac{1}{2}, -\frac{1}{6}, \frac{1}{24}, -\frac{1}{120}, \frac{1}{720}, -\frac{1}{5040}, \frac{1}{40\,320}, -\frac{1}{362\,880}, \frac{1}{3\,628\,800} \right\}$$

$$\frac{(-1)^{k+1}}{(1)_{k-1}}$$

The prefactor z is introduced in order to obtain a modified series which enables **FindSequenceFunction** to find the series coefficient c_{1001} in closed form. Hence, the resulting infinite series $\mathcal{P}_{1001}(z)$ is in agreement with the direct result from **MeijerG** $[\{\{\}, \{\}\}, \{\{1-a\}, \{\}\}, z]$.

$$\mathcal{P}_{1001}[n_, z_] := \frac{1}{z} \sum_{k=1}^n c_{1001} z^{k-1}; \quad \mathcal{P}_{1001}[\infty, z]$$

$$\frac{e^{-z}}{z}$$

■ (4) Improper integrals transformed into contour integrals through *change of variables*

There are further types of contour integrals which do not fit into the categories of definite integrals given in Part 1. However, extending the original integration path along the real axis \mathbb{R} into the complex plane \mathbb{C} and closing the contour γ the integral can be evaluated in terms of a contour integral.

For examples *improper integrals* (with integration originally along the real axis \mathbb{R}) can be casted into contour integrals and evaluated in \mathbb{C} by means of some tricky transformation which involves *change of (integration) variable*.

In MathGroup several approaches to this problem are found. See for example [21], the sequel of discussions in MathGroup by Alex Krasnov, Alexander Elkins and others. In addition, there exist some Mathematica packages [22], one provided by David Park as part of his Presentations package, i.e. **Presentations`StudentsIntegral**, and another one, **SymbolicComputing** which supports much more than change of integration variables only.

■ Change of variables using **changeVariable4Integral**

Due to private communication with M. Trott [23] the procedure **changeVariable4Integral** is implemented in the **ContourIntegration** package.

? **changeVariable4Integral**

changeVariable4Integral[$\int_{z_a}^{z_b} f(z) dz$, { ζ , $z2\zeta$, $\zeta2z$ }] calculates the definite integral $\int_{z_a}^{z_b} f(z) dz$ with a change of variables for the integral $\int_{\zeta_a}^{\zeta_b} g(\zeta) d\zeta$. Here 'fz' is the integrand $f(z)$ and 'dz' the differential. 'z' is the old and ' ζ ' the new variable. ' $z2\zeta$ ' is the transform from $z \Rightarrow \zeta$ (e.g. $\zeta = r$ and $z2\zeta = r \cdot e^{i\phi}$), ' $\zeta2z$ ' is the inverse function (e.g. $\zeta2z = z \cdot e^{-i\phi}$, $\phi = \text{const}$). {'za', 'zb'} are the lower/upper limits of the integral which must be transformed too, i.e. $\{z_a, z_b\} \Rightarrow \{\zeta_a, \zeta_b\} = \{\zeta2z/.{z \rightarrow z_a}, \zeta2z/.{z \rightarrow z_b}\}$.

■ Evaluation using contour integration with tricky contours

The following two examples taken from S. Hassani [24] illustrate how *improper integrals* (with integration originally along the real axis \mathbb{R}) can be casted into contour integrals which are evaluated in the complex plane \mathbb{C} by some tricky transformation. The following examples are found in [24] Chapt. 10.3.9 pp. 284 -285.

Example 4.1 : $\int_0^{+\infty} \frac{1}{x^3+1} dx = \frac{2\pi}{3\sqrt{3}}$

Consider the integral along the positive real axis :

$$\mathcal{I}_1 = \int_0^{\infty} \frac{1}{x^3+1} dx .$$

Since the integrand $f(x) = \frac{1}{x^3+1}$ is an *odd* function the lower limit of the integral cannot be extended to $-\infty$. But with a trick the contour can be closed as a 120° sector in \mathbb{H}_+ .

The underlying idea is to perform a contour integration and apply a suitable change of the integration variable to evaluate some line integral.

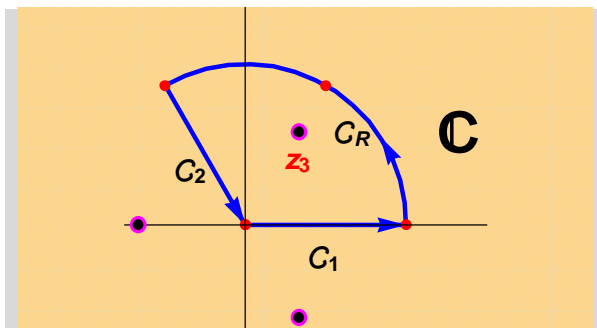
(1) Investigating the singularities of the integrand $f(z)$ which are simply the roots of the denominator : $z^3 + 1 = 0$ with solutions $z(k) = e^{(2k-1)i\pi/3}$ for $(k = 1, 2, 3)$ gives a hint how to close the contour in \mathbb{C} .

$$\text{sorted roots } \{z_1, z_2, z_3\} = \left\{ -1, \frac{1}{2} - \frac{i\sqrt{3}}{2}, \frac{1}{2} + \frac{i\sqrt{3}}{2} \right\}$$

Hence, if the roots are *sorted* with respect to their imaginary parts then the root in the 1st quadrant is given as $z_3 = e^{\frac{i\pi}{3}} = \frac{1+i\sqrt{3}}{2}$

(2) Instead of integration along the real axis from $[0, +\infty)$ a closed contour γ is chosen as a 120° segment including the pole $z_3 = e^{i\pi/3} = \left(\frac{1+i\sqrt{3}}{2} \right)$

located in the first quadrant. The three roots as well as the contour γ that only has z_3 as an interior point are shown below. The closed contour γ is specified by the following points in the complex plane \mathbb{C} : $\{0, R, R \cdot (-\frac{3}{4} + i \frac{3\sqrt{3}}{4}), 0\}$. For consistency the interval $[0, R]$ is denoted by C_1 , the arc segment from $[R, R \cdot (-\frac{3}{4} + i \frac{3\sqrt{3}}{4})]$ is C_R and the interval $[R \cdot (-\frac{3}{4} + i \frac{3\sqrt{3}}{4}), 0]$ is denoted as C_2 where $R \rightarrow \infty$. Hence, the closed contour $\gamma = C_1 + C_R + C_2$ comprises a segment of 120° encircling the pole $z_3 = \frac{1+i\sqrt{3}}{2}$.



(3) The contour integral consists of three parts $\mathcal{J}_\gamma = \mathcal{I}_1 + \mathcal{J}_R + \mathcal{I}_2$ which are evaluated separately.

The contour integral \mathcal{J}_γ can be split into three parts:

$$\mathcal{J}_\gamma = \mathcal{I}_1 + \mathcal{J}_R + \mathcal{I}_2 = \int_0^{+\infty} \frac{1}{x^3+1} dx + \int_{C_R} \frac{1}{z^3+1} dz + \int_{C_2} \frac{1}{z^3+1} dz = 2\pi i \operatorname{Res} f(z) |_{z=z_3}$$

thus the original integral to be calculated is $\mathcal{I}_1 = \mathcal{J}_\gamma - \mathcal{I}_2 - \mathcal{J}_R$. The following steps are performed :

- First, the contour integral \mathcal{J}_γ (taking into account only the pole $z_3 = \frac{1+i\sqrt{3}}{2}$) can be calculated straightforwardly

$$\mathcal{J}_\gamma = \text{ContourIntegral}\left[\frac{1}{z^3+1}, z, \{3\}, \{\}, \text{"No"}\right] \quad // \text{ ce}$$

$$\text{Contour integral } \mathcal{J} = \oint \frac{1}{z^3+1} dz = \frac{1}{3} (\sqrt{3} - i) \pi$$

$$-\frac{i\pi}{3} + \frac{\pi}{\sqrt{3}}$$

Applying the *residue theorem* gives the same result.

- *Second*, the integral \mathcal{J}_R along the circular arc C_R vanishes for $R \rightarrow \infty$, as usual.
- *Third*, calculation of integral \mathcal{I}_2 along C_2

Thus, in order to evaluate the line integral \mathcal{I}_2 along C_2 (which runs from $R \cdot \left(-\frac{3}{4} + i \frac{3\sqrt{3}}{4}\right)$ to 0 with $R \rightarrow \infty$) one *changes the integration variable* from $z \rightarrow \zeta e^{i\phi}$. The integral over ζ turns out simple if $\phi = 2\pi/3$ is chosen : $\mathcal{I}_2 = \int_{C_2} \frac{1}{z^3+1} dz = \int_{\infty}^0 \frac{e^{i2\pi/3}}{(\zeta e^{i2\pi/3})^3+1} d\zeta = -e^{i2\pi/3} \int_0^{\infty} \frac{1}{\zeta^3+1} d\zeta = -e^{i2\pi/3} \mathcal{I}_1$.

The integration can be done *directly* with the help of procedure **changeVariable4Integral** with the transformation of variables are $z \rightarrow \zeta e^{i\frac{2\pi}{3}}$ and its reverse $\zeta \rightarrow z e^{-i\frac{2\pi}{3}}$,

or *explicitly* by changing the integration variable from z to ζ where the integration boundaries must be transformed correspondingly.

In both cases the result is $\frac{2\pi}{3\sqrt{3}}$.

(4) Finally, the original integral $\mathcal{I}_1 = \int_0^{\infty} \frac{1}{x^3+1} dx$ (along the positive real axis or $C_1=[0,\infty)$) is obtained from $\mathcal{J}_\gamma = \mathcal{I}_1 + \mathcal{I}_2 = (1 - e^{i2\pi/3}) \mathcal{I}_1$ where the contour integral is $\mathcal{J}_\gamma = \left(\frac{\pi}{\sqrt{3}} - i\frac{\pi}{3}\right) = \text{Res}_3$. Solving for \mathcal{I}_1 there is: $\mathcal{I}_1 = \frac{2\pi}{3\sqrt{3}}$.

Solve[(1 - e^{i 2 π/3}) I₁ == J_γ, I₁][[1, 1, 2]]

$$\frac{2\pi}{3\sqrt{3}}$$

Example 4.2 : $\int_{-\infty}^{+\infty} e^{i\alpha x - \beta x^2} dx = \sqrt{\pi/\beta} e^{-\alpha^2/4\beta}$.

Consider the Gaussian integral

$$I = \int_{-\infty}^{+\infty} e^{i\alpha x - \beta x^2} dx \quad \text{where } \alpha, \beta \in \mathbb{R}, \beta > 0$$

(Side note : with *Mathematica* V9 at least this integral can be calculated directly.)

With a trick this integral can again be calculated in terms of a contour integral \mathcal{J}_γ .

The underlying idea is to change the variable of integration from $x \rightarrow z + i \frac{\alpha}{2\beta}$, then evaluate the contour integral for the analytic function $e^{-\beta z^2}$ on a closed contour $\gamma = C_1 + C_2 + C_3 + C_4$ with the corresponding line integrals \mathcal{I}_i .

(1) Completing squares in the exponent of $e^{i\alpha x - \beta x^2}$ one obtains a simpler line integral but its path is shifted by $-i \frac{\alpha}{2\beta}$:

(2) The contour $\gamma = C_1 + C_2 + C_3 + C_4$ is *closed* as shown below.

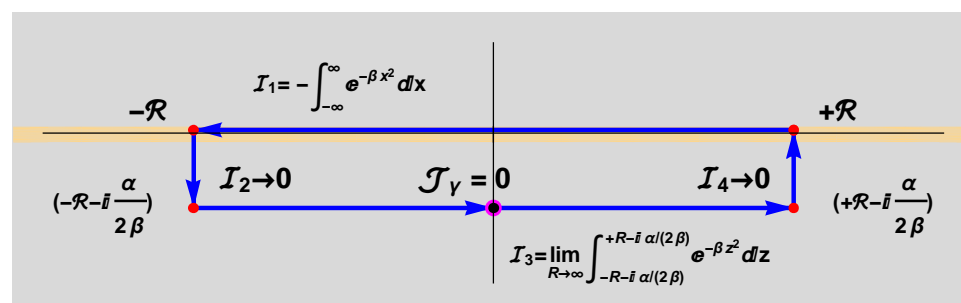
Because $e^{-\beta z^2}$ is an *analytic function* the contour integral $\mathcal{J}_\gamma = \oint_\gamma e^{-\beta z^2} dz$ vanishes. (Cauchy-Goursat theorem).

Contour integral $\mathcal{J} = \oint e^{\beta(-z^2)} dz = 0$

(3) The contour integral $\mathcal{J}_\gamma = 0$ can be split into four line integrals \mathcal{I}_i along C_i ($i=1,\dots,4$):

$$\begin{aligned} \mathcal{J}_\gamma &= \oint_\gamma e^{-\beta z^2} dz = 0 = \sum_{i=1}^4 \mathcal{I}_i = \int_{C_1} e^{-\beta z^2} dz + \int_{C_2} e^{-\beta z^2} dz + \int_{C_3} e^{-\beta z^2} dz + \int_{C_4} e^{-\beta z^2} dz = \\ &= \lim_{R \rightarrow \infty} \int_{-R}^{-R} e^{-\beta x^2} dx + \lim_{R \rightarrow \infty} \int_{-R}^{-R-i\alpha/(2\beta)} e^{-\beta z^2} dz + \lim_{R \rightarrow \infty} \int_{-R-i\alpha/(2\beta)}^{+R-i\alpha/(2\beta)} e^{-\beta z^2} dz + \lim_{R \rightarrow \infty} \int_{+R-i\alpha/(2\beta)}^{+R} e^{-\beta z^2} dz = \end{aligned}$$

$$= - \int_{-\infty}^{+\infty} e^{-\beta x^2} dx + 0 + \lim_{R \rightarrow \infty} \int_{-R-i\alpha/(2\beta)}^{+R-i\alpha/(2\beta)} e^{-\beta z^2} dz + 0 = I_1 + I_3$$



The contour integral \mathcal{J}_γ over the closed contour γ is zero :

In the limit $R \rightarrow \infty$ the integral boundaries go to infinity. Along C_2 (from $-R$ to $-R - i\alpha/(2\beta)$) and C_4 (from $R - i\alpha/(2\beta)$ to R) the following substitutions $z = -(R+i y)$ respectively $z = (R+i y)$ are made which give rise to a vanishing prefactor $\lim_{R \rightarrow \infty} e^{-\beta R^2}$ so that the integrals $I_{2,4}$ tend to zero.

With *Mathematica* the integrals I_i ($i=1,\dots,4$) are calculated as $\left\{-\frac{\sqrt{\pi}}{\sqrt{\beta}}, 0, \frac{\sqrt{\pi}}{\sqrt{\beta}}, 0\right\}$

The only non-zero integrals $I_{1,3}$ are familiar *Gaussian integrals*. Obviously, the sum of all line integrals I_i ($i=1,\dots,4$) vanishes as does the contour integral $\mathcal{J}_\gamma = 0 = I_1 + I_3$.

(4) Finally, the result for the original integral \mathcal{I} is $\int_{-\infty}^{+\infty} e^{i\alpha x - \beta x^2} dx = e^{-\frac{\alpha^2}{4\beta}} \cdot I_3 = e^{-\frac{\alpha^2}{4\beta}} \sqrt{\pi/\beta}$

An alternative treatment is to use straightforwardly the procedure `changeVariable4Integral` with the transformation $x \rightarrow z + i\frac{\alpha}{2\beta}$.

■ (5) Action integrals in the context of Bohr-Sommerfeld theory

Already in the year 1919 Arnold Sommerfeld [25], [26] had presented an interesting approach for the evaluation of certain types of action integrals (occurring in Hamilton-Jacobi theory) using the technique of contour integration and the residue theorem.

■ (1) Action integral of type $\mathcal{J}_r = \oint p_r dr = \oint \sqrt{A + 2 \frac{B}{r} - \frac{C}{r^2}} dr$

- Theory and derivation of the action integral of type $\oint \sqrt{A + 2 \frac{B}{r} - \frac{C}{r^2}} dr$ is discussed in some detail here.
- Branch cut and pole structure of $\oint \sqrt{A + 2 \frac{B}{r} - \frac{C}{r^2}} dr$

However, we are only interested in the evaluation of the following contour integral $\oint \sqrt{A + 2 \frac{B}{r} - \frac{C}{r^2}} dr$ which has a *branch cut* originating from the square root. The *branch points* $\{r_1, r_2\}$ are obtained as solution of the equation $A + 2 \frac{B}{r} - \frac{C}{r^2} = 0$ w.r.t. r :

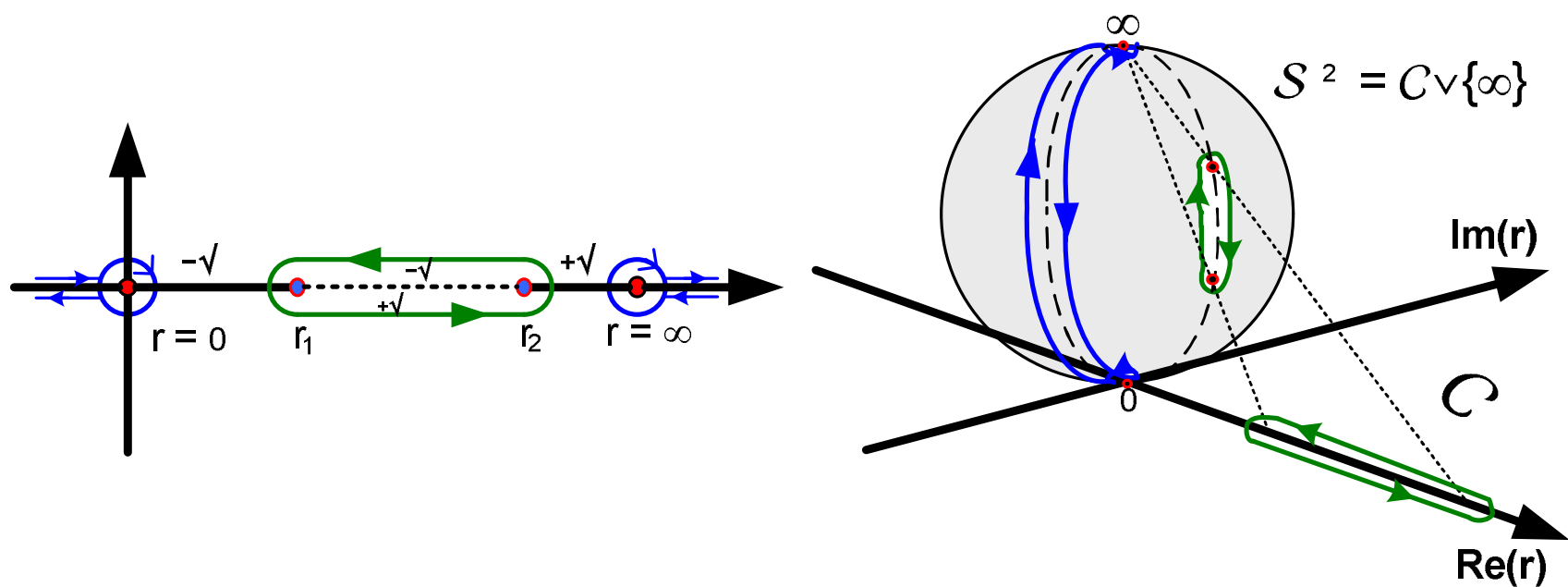
The coefficients of the radicant can be expressed in terms of $\{r_1, r_2\}$:

$$\{r_1, r_2, (r_1 + r_2)\} // \text{fs}$$

$$\left\{ -\frac{C}{A}, -\frac{2B}{A} \right\}$$

The movement (represented by the action integral) is constricted between $r_1 = -\frac{B + \sqrt{B^2 + AC}}{A}$, $r_2 = -\frac{B - \sqrt{B^2 + AC}}{A}$ with $r_1 < r_2$ (and $A = 2mE < 0$). A complete r -cycle runs from r_1 to r_2 and back to r_1 . Along the path $r_1 \Rightarrow r_2$ the momentum $p_r > 0$ so that the *positive sign* of the square root must be taken ($+\sqrt{\quad}$); on the way back, i.e. $r_2 \Rightarrow r_1$, holds $p_r < 0$ and the *negative sign* of the square root has to be selected ($-\sqrt{\quad}$). Thus, the integration is extended over both sheets of the two-valued function $f(r) = \pm \sqrt{A + 2 \frac{B}{r} - \frac{C}{r^2}}$; the *branch points* r_1 and r_2 define a *branch cut* between r_1 and r_2 on the real axis.

For better imagination it is appropriate to map the complex r -plane \mathbb{C} on the Riemann sphere $\mathbb{S}^2 = \mathbb{C} \cup \{\infty\}$; the origin $r_0 = 0$ becomes the “South pole” and $r_\infty \rightarrow \infty$ the “North pole” (here ∞ is called a **ComplexInfinity** in *Mathematica*). Any closed contour path on the Riemann sphere separates two complementary regions (the inner and outer region which depend on the orientation of the contour path). The (blue) contour path is unique and encloses the two singularities at $r_0 = 0$ and $r_\infty = \infty$.



The $\text{Re}(r)$ -axis projected on the Riemann sphere is a meridian through the South pole and the North pole. The positive $\text{Re}(r)$ -axis which extends from 0 to ∞ is the 0° meridian; similarly the negative $\text{Re}(r)$ -axis extending from 0 to $-\infty$ is the 180° meridian. The *branch cut* between r_1 and r_2 is located on the positive $\text{Re}(r)$ -axis and is encircled counter-clockwise there. However, mapping this path on the Riemann sphere will reverse its orientation to clockwise.

Now, the (green) contour path encircling the branch cut between r_1 and r_2 can be stretched (like a rubber band) over the Riemann sphere and is stuck only at the two singularities $r_0 = 0$ ("South pole") and $r_\infty = \infty$ ("North pole"); obviously, on the Riemann sphere both poles are encircled counter-clock-

ContourIntegration_P2.nb

wise. Contributions from the path (blue lines/semi-circles in opposite directions) on the 180° meridian compensate each other so that only the residues for r_0 and r_∞ remain.

• Residues of $f(r) = \sqrt{A + 2 \frac{B}{r} - \frac{C}{r^2}}$

The sign of the square root ($f(r) = \pm \sqrt{\quad}$) is *negative* for $r < r_1$ and *positive* for $r > r_2$ as shown by closer investigation of the behaviour of the function $f(r)$. Thus, the residue \mathcal{R}_0 for a 1st order pole at $z = r_0 = 0$ with $f(z) = -\sqrt{A + 2 \frac{B}{z} - \frac{C}{z^2}}$ becomes

$$f_0[z] := -\sqrt{A + 2 \frac{B}{z} - \frac{C}{z^2}}$$

$$\mathcal{R}_0 = \text{Residue}[f_0[z], \{z, 0\}]$$

$$-\sqrt{-C}$$

Next, as $z \rightarrow \infty$ the residue for r_∞ is calculated. Above r_2 the positive sign of the square root must be chosen $f(z) = +\sqrt{A + 2 \frac{B}{z} - \frac{C}{z^2}}$. Substitution of $w = \frac{1}{z}$ maps the singularity at r_∞ to 0. There occurs a 2nd order pole as is obvious from $f(z) dz = -\frac{1}{w^2} f\left(\frac{1}{w}\right) dw$. The residue \mathcal{R}_∞ becomes

$w = . ;$

$$f_\infty[w] := \left(-f_0[z] * D\left[\frac{1}{w}, w\right] \right) /. \{z \rightarrow \frac{1}{w}\}$$

$f_\infty[w]$

$$\mathcal{R}_\infty = \text{Residue}[f_\infty[w], \{w, 0\}]$$

$$-\frac{\sqrt{A + 2 w B - w^2 C}}{w^2}$$

$$-\frac{B}{\sqrt{A}}$$

Finally, due to change of orientation (the branch cut contour is counter-clockwise whereas the poles r_0 and r_∞ are encircled clockwise) the value of the action integral \mathcal{J}_r is $-2\pi i \cdot (\mathcal{R}_0 + \mathcal{R}_\infty)$

$$\mathcal{J}_r = -2\pi i (\mathcal{R}_0 + \mathcal{R}_\infty) \quad // \quad \text{sf}$$

$$2i\pi \left(\frac{\mathcal{B}}{\sqrt{\mathcal{A}}} + \sqrt{-C} \right)$$

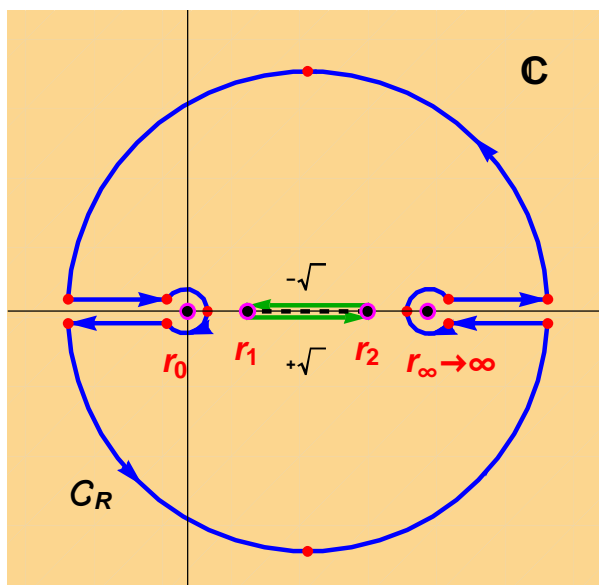
Evaluation with r_1 and r_2 with $F_0(r) = \frac{1}{r} \sqrt{(r-r_1)(r-r_2)}$ and $F_\infty(\rho) = -\frac{1}{\rho} \sqrt{\left(\frac{1}{\rho}-r_1\right)\left(\frac{1}{\rho}-r_2\right)}$ leads to the same result.

Finally, the result for \mathcal{J}_r is

$$\mathcal{J}_r = \oint \sqrt{A + 2\frac{B}{r} + \frac{C}{r^2}} dr = 2\pi i \left(\frac{B}{\sqrt{A}} + \sqrt{-C} \right)$$

- “Double keyhole” contour

The contour around the branch cut can be “blown up” to a “double keyhole” contour shown below which encloses the singular points $\{r_0, r_\infty\}$. As regards to a graphical representation numerical values must be chosen for $\{r_1, r_2\} = \{1/2, 3/2\}$ and for the singular points $\{r_0, r_\infty\} = \{0, 2\}$.



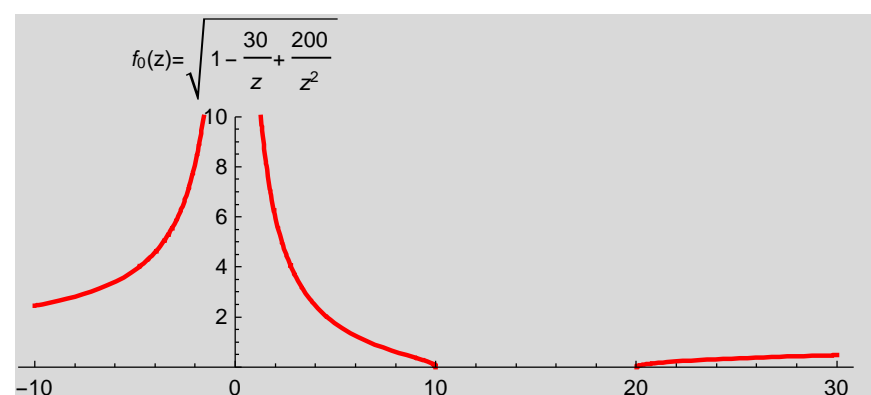
Because the (green) contour path encircles the branch cut *counter-clockwise* the method of residues cannot be applied directly. However, this contour path can also be interpreted such as encircling the outer region of the complex r -plane (blue contour). In this outside region the integrand is unique so that there is no restriction to apply the residue theorem. In the limit $r \rightarrow \infty$ only two singularities $\{r_0, r_\infty\}$ are essential and the (blue) contour can be contracted such that only the singularities are encircled *clockwise*.

- Plots of $\sqrt{\mathcal{A} + 2 \frac{\mathcal{B}}{r} - \frac{\mathcal{C}}{r^2}}$ with parameters arbitrarily set to $\{\mathcal{A}, \mathcal{B}, \mathcal{C}\} = \{1, -15, 200\}$ are displayed below.

In order to visualize the function $f_0(r) = \sqrt{\mathcal{A} + 2 \frac{\mathcal{B}}{r} - \frac{\mathcal{C}}{r^2}} = \frac{1}{r^2} \sqrt{(r - r_1)(r - r_2)}$ in the complex r -plane suitable values for $\{r_1, r_2, r_\infty\}$ should be taken which are large enough to emphasize the branch cut between (r_1, r_2) . Contour and 3d plots give an impression about the behavior of $f_0(r)$.

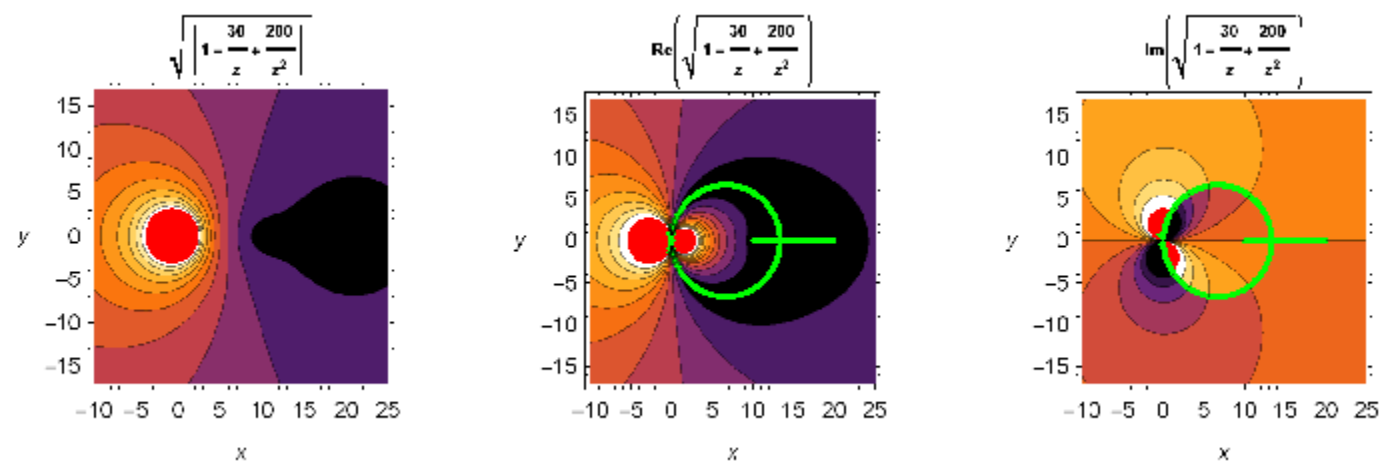
Assuming for $\{r_1, r_2, r_\infty\} = \{10, 20, 50\}$, thus the corresponding values for $\{\mathcal{A}, \mathcal{B}, \mathcal{C}\} = \{1, -15, -200\}$ are found.

- Here is the plot of $f_0(z)$ along the real axis which clearly shows the gap, i.e. the *branch cut*, between **10** and **20**.



- The procedure `complexFctContourPlot` will display the function $f_0(z)$ as a contour plot where possible branch cuts are shown as green lines.

? `complexFctContourPlot`



Obviously, in addition to the *linear branch cut* extending from $r_1 = 10$ to $r_2 = 20$ there is a circular branch cut through the origin with radius $R = \frac{20}{3}$.

ContourIntegration_P2.nb

```
(ComplexAnalysis`BranchCuts[fz, z] /. {Re[z] -> x, Im[z] -> y}) // sf
```

$$(y = 0 \ \&\& \ 10 < x < 20) \ || \ \left(0 < x < \frac{40}{3} \ \&\& \ \left(y = -\frac{\sqrt{-x(-40+3x)}}{\sqrt{3}} \ || \ y = \frac{\sqrt{-x(-40+3x)}}{\sqrt{3}} \right) \right)$$

Here are the corresponding 3d plots using **complexFctPlot3D** for the function $f_0(z)$ in terms of its absolute, real and imaginary values.

• In order to take further *correction terms* into account the reader is referred to the original work of A. Sommerfeld [25] pp. 656 - 659. Modified contour integrals are investigated below.

(i) From physical point of view the introduction of an additional correction term $D_0 r$ is relevant for the investigation of the theory of the *Stark effect*.

$$(i) \quad \mathcal{J}_1 = \oint \sqrt{A + 2 \frac{B}{r} - \frac{C}{r^2} + D_0 r} \, d\mathbf{r} = \mathcal{J}_r + \frac{1}{2} D_0 \mathcal{K}_1 - \frac{1}{8} D_0^2 \mathcal{K}_2$$

The integral can be solved by a series expansion of the square root in terms of powers of D_0 . For the calculation only the pole r_∞ together with the substitution $r = \frac{1}{z}$ has to be taken into account. For $r_0 = 0$ the integrand behaves regular. The type of integrals involved are :

$$\mathcal{K}_1 = -\frac{1}{\sqrt{A}} \oint \frac{1}{z^3} \frac{1}{\sqrt{1 + 2 \frac{B}{A} z - \frac{C}{A} z^2}} dz \quad \text{and} \quad \mathcal{K}_2 = -\frac{1}{A^{3/2}} \oint \frac{1}{z^4} \frac{1}{\left(1 + 2 \frac{B}{A} z - \frac{C}{A} z^2\right)^{3/2}} dz$$

so that $\mathcal{J}_1 = 2\pi i \left[\left(\frac{B}{\sqrt{A}} + \sqrt{-C} \right) + \frac{D_0}{4A^{3/2}} \left(\frac{3B^2}{A} + C \right) + \frac{5}{16} \frac{BD_0^2}{A^{7/2}} \left(\frac{7B^2}{A} + 3C \right) \right]$

(ii) As regards to the theory of *hydrogen-unlike spectra* an additional term D_1/r^3 is introduced.

$$(ii) \quad \mathcal{J}_2 = \oint \sqrt{A + 2 \frac{B}{r} - \frac{C}{r^2} + \frac{D_1}{r^3}} \, d\mathbf{r} = \mathcal{J}_r + \frac{1}{2} D_1 \mathcal{K}_3$$

Again, the integrand is expanded in a power series with respect to D_1 so that :

$$\mathcal{K}_3 = - \oint \frac{z}{\sqrt{A + 2Bz - Cz^2}} dz$$

so that $\mathcal{J}_2 = 2\pi i \left[\left(\frac{B}{\sqrt{A}} + \sqrt{-C} \right) + \frac{B}{2(-C)^{3/2}} D_1 \right]$

(iii) Adding a second correction term D_2/r^4 (where D_2 is of the order of D_1^2) the contour integral to be investigated turns out to be

$$(iii) \mathcal{J}_3 = \oint \sqrt{A + 2\frac{B}{r} - \frac{C}{r^2} + \frac{D_1}{r^3} + \frac{D_2}{r^4}} dr = \mathcal{J}_r + \frac{1}{2} D_1 \mathcal{K}_3 + \frac{1}{2} D_2 \mathcal{K}_4 - \frac{1}{8} D_1^2 \mathcal{K}_5$$

Again, the square root is expanded with respect to the correction terms D_2 and D_3 where terms with powers higher than D_2^n ($n > 2$) and D_3^m ($m > 1$) will be neglected.

$$\mathcal{K}_4 = \oint \frac{1}{r^4} \frac{1}{\sqrt{A + 2\frac{B}{r} - \frac{C}{r^2}}} dr \quad \text{and} \quad \mathcal{K}_5 = \oint \frac{1}{r^6} \frac{1}{\left(A + 2\frac{B}{r} - \frac{C}{r^2}\right)^{3/2}} dr$$

so that $\mathcal{J}_3 = 2\pi i \left[\left(\frac{B}{\sqrt{A}} + \sqrt{-C} \right) + \frac{1}{2} \frac{B}{C^{3/2}} \left(D_1 + \frac{3}{2} \frac{B}{C} D_2 + \frac{15}{8} \frac{B}{C^2} D_1^2 \right) + \frac{1}{4} \frac{A}{(-C)^{3/2}} \left(D_2 + \frac{3}{4} \frac{1}{C} D_1^2 \right) \right]$

■ **(2) Action integral of type** $\mathcal{J}_\theta = \oint p_\theta d\theta = \oint \sqrt{C - \frac{D^2}{\sin^2(\theta)}} d\theta$

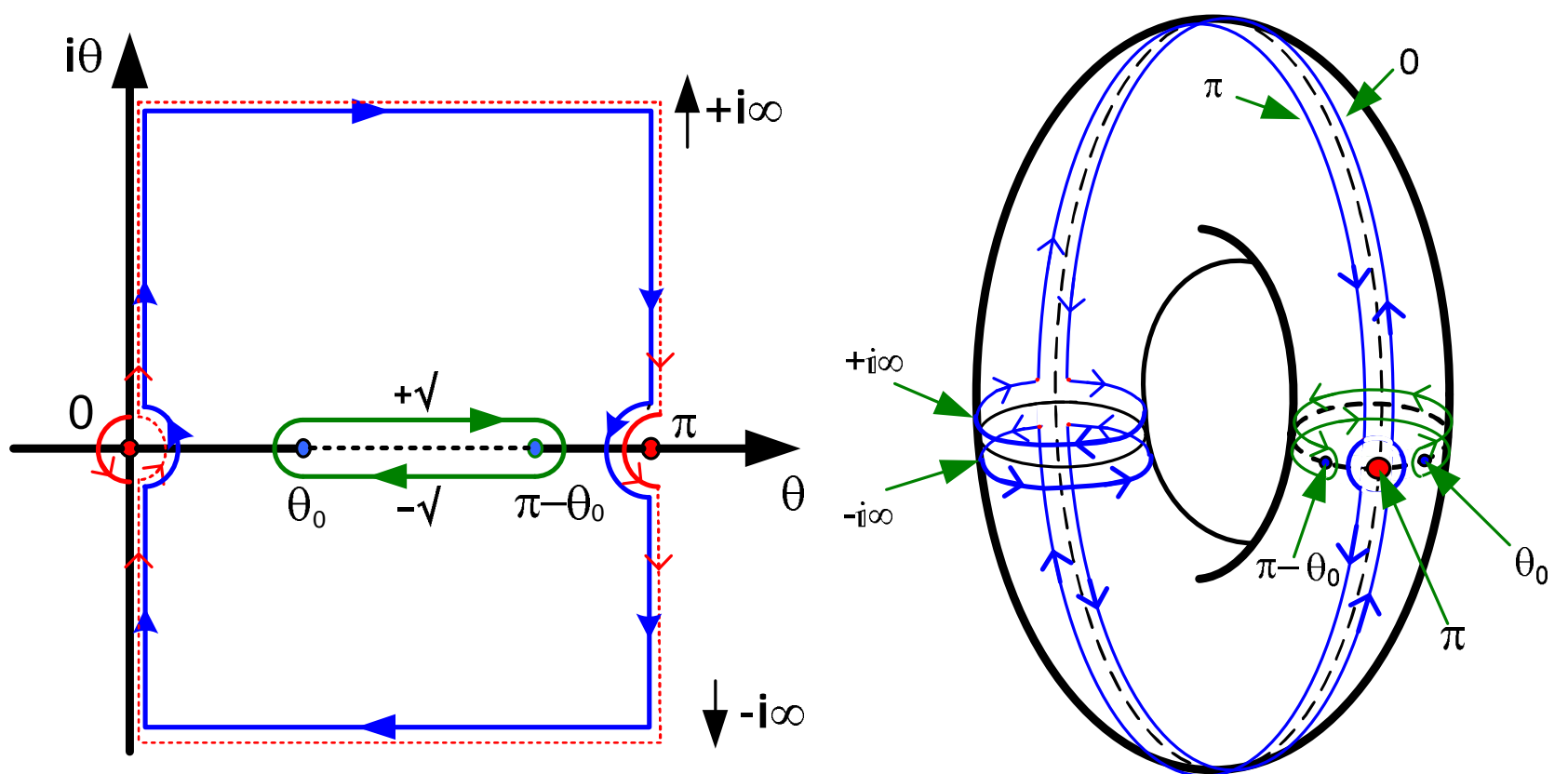
- Theory and derivation of the contour integral type $\oint \sqrt{C - \frac{D^2}{\sin^2(\theta)}} d\theta$ is discussed in some detail here.

Assuming for the coefficient $C = \frac{D^2}{\sin^2(\theta_0)}$ the integrand is simplified to

$$g[\theta_-] := |\mathcal{D}| \sqrt{\sin[\theta_0]^{-2} - \sin[\theta]^{-2}}$$

The integration path in the complex θ -plane encircles the *branch cut* between the angles $\{\theta_0, \pi - \theta_0\}$ and gives rise to the geometry shown in the figure be-

low on the lhs. Due to the periodicity of the integrand $\mathbf{g}(\theta)$ the contour path can be continuously deformed to the border line for the (periodic) strip between 0 and π with the singularities 0 and π excluded.



As regards to periodicity the strip in the complex θ -plane between $[0, \pi)$ can be folded into an infinite tube by stitching together left and right border. If, in addition, both ends of the tube at $+i\infty$ and $-i\infty$ are joined there results a *torus* as shown in the figure on the rhs. Again, the integration path can be deformed such that only the singularity at $\theta = 0$ is encircled. This corresponds to the geometric operation of folding the semi-circle at $\theta = \pi$ with the one

at $\theta = 0$. In addition, only the integration path of 2π at infinity remains since the circumference of the torus is just π (which is the period of the integrand).

The residue (for the single pole) at $\theta = 0$ is

$$\mathcal{R}_0 = \text{Residue}[\mathbf{g}[\theta], \{\theta, 0\}]$$

$$\mathbf{i} \, |\mathcal{D}|$$

Similarly, the asymptotic contribution of the integral for $\theta \rightarrow \infty$ becomes $\oint p_\theta d\theta \simeq \lim_{\theta \rightarrow \infty} \oint g(\theta) d\theta = 2\pi \frac{|\mathcal{D}|}{\sin(\theta_0)}$ where $\frac{|\mathcal{D}|}{\sin^2(\theta_0)} = \sqrt{C}$.

$$\mathcal{J}_\infty = 2\pi \text{Limit}[\mathbf{g}[\mathbf{i} \, \theta], \theta \rightarrow \infty] /. \left\{ \sqrt{(\mathbf{v}_-)^2} \rightarrow \mathbf{v} \right\} /. \left\{ |\mathcal{D}| \text{Csc}[\theta_0] \rightarrow \sqrt{C} \right\}$$

$$2\pi \sqrt{C}$$

Hence, the final result of the contour integral turns out to be

$$\mathcal{J}_\theta = (2\pi \mathbf{i} \, \mathcal{R}_0 + \mathcal{J}_\infty) // \text{Expand} // \text{sf} // \text{polyForm}$$

$$2\pi \left(\sqrt{C} - |\mathcal{D}| \right)$$

$$\mathcal{J}_\theta = \oint \sqrt{C - \frac{D^2}{\sin^2(\theta)}} d\theta = 2\pi \left(\sqrt{C} - |\mathcal{D}| \right)$$

Note, that $\sqrt{C} = L$ and $D = L_z$ are the angular momentum and the \mathbf{z} -component. (In the framework of the Bohr-Sommerfeld theory the evaluation of action integrals leads to quasi-classical quantization.)

■ Conclusions

In conclusion the author is convinced that the package **ContourIntegration.m** will be a useful extension of the built-in procedure **Integrate** in *Mathematica*. It is the author's expectation that *Mathematica* users will find the main procedure **ContourIntegral** and additional routines for finding poles, evaluating the corresponding residues, designing intricate contour paths and visualizing various views of complex functions useful.

■ Appendix

■ (1) Representation of special functions in terms of Meijer G-functions

In several cases Meijer G-functions reduce to simpler special functions; a selection given by E.W. Weisstein and M. Trott [19] is listed below :

$$\bullet G_{1,2}^{2,1} \left(2z \left| \begin{matrix} 1/2 \\ 3, -3 \end{matrix} \right. \right) \Longleftrightarrow \text{Bessel function } K_3(z) \Rightarrow f_{2112}(2z, s) = 2^{-s} z^{-s} \Gamma\left(\frac{1}{2} - s\right) \Gamma(s-3) \Gamma(s+3) \quad (\text{Example 1})$$

Note that for given $G_{1,2}^{2,1}$ the numerator simplifies to 1 due to the empty products $\prod_{j=1}^p$ sowie $\prod_{j=2}^q$ (m=2, n=1)

$$\left\{ \text{MeijerG} \left[\left\{ \left\{ \frac{1}{2} \right\}, \{\} \right\}, \{\{3, -3\}, \{\}\}, 2z \right] // \text{hF} // \text{tF} \quad \left(\star G_{1,2}^{2,1} \left(2z \left| \begin{matrix} 1/2 \\ 3, -3 \end{matrix} \right. \right) \star \right), \right. \\ \left. \text{res} = \text{MeijerG} \left[\left\{ \left\{ \frac{1}{2} \right\}, \{\} \right\}, \{\{3, -3\}, \{\}\}, 2z \right], \text{res} // \text{tF} \right] // \text{cF}$$

$$G_{1,2}^{2,1} \left(2z \left| \begin{matrix} \frac{1}{2} \\ 3, -3 \end{matrix} \right. \right) \\ = e^z \sqrt{\pi} \text{BesselK}[3, z] \\ \sqrt{\pi} (-e^z) K_3(z)$$

$$\bullet G_{0,2}^{1,0}(z | u, -u) \Longleftrightarrow \text{Bessel function } J_2(2\sqrt{z}) \Rightarrow f_{1002}(z, s) = \frac{z^{-s} \Gamma(s+u)}{\Gamma(-s+u+1)}$$

$$\bullet G_{2,2}^{1,2} \left(z \left| \begin{matrix} 1/2, 1/2 \\ 0, 0 \end{matrix} \right. \right) \Longleftrightarrow \text{Elliptic function } 2K(-z) \Rightarrow f_{1222}(z, s) = \frac{z^{-s} \Gamma\left(\frac{1}{2}-s\right)^2 \Gamma(s)}{\Gamma(1-s)}$$

- $G_{\substack{2,1 \\ \substack{1,2} \\ \substack{c,b}}}^{\substack{2,1 \\ \substack{1,2} \\ \substack{c,b}}}\left(2z \mid \begin{matrix} a \\ c, b \end{matrix}\right) \iff$ Hypergeometric function $U(\alpha, \beta, z) \implies f_{2112}(2z, s) = 2^{-s} z^{-s} \Gamma(-a-s+1) \Gamma(b+s) \Gamma(c+s)$
- $G_{\substack{1,3 \\ \substack{3,3} \\ \substack{0,-3/2,-3/2}}}^{\substack{1,3 \\ \substack{3,3} \\ \substack{0,-3/2,-3/2}}}\left(-z \mid \begin{matrix} 1/2, 1/2, 1/2 \\ 0, -3/2, -3/2 \end{matrix}\right) \iff$ Generalized hypergeometric function ${}_pF_q(\alpha; \beta; z) \implies f_{1333}(-z, s) = \frac{(-z)^{-s} \Gamma\left(\frac{1}{2}-s\right)^3 \Gamma(s)}{\Gamma\left(\frac{3}{2}-s\right)^2}$
- $G_{\substack{1,2 \\ \substack{2,2} \\ \substack{1,0}}}^{\substack{1,2 \\ \substack{2,2} \\ \substack{1,0}}}\left(z \mid \begin{matrix} 1, 1 \\ 1, 0 \end{matrix}\right) \iff$ Logarithmic function $\log(z+1) \implies f_{1222}(z, s) = \frac{z^{-s} \Gamma(-s)^2 \Gamma(s+1)}{\Gamma(1-s)}$
- $G_{\substack{1,2 \\ \substack{2,2} \\ \substack{1,1}}}^{\substack{1,2 \\ \substack{2,2} \\ \substack{1,1}}}\left(z \mid \begin{matrix} 1, 1 \\ 1, 1 \end{matrix}\right) \iff$ Rational function $\frac{z}{z+1} \implies f_{1222}(z, s) = z^{-s} \Gamma(-s) \Gamma(s+1)$ (*Example 2*)
- $G_{\substack{1,0 \\ \substack{0,2} \\ \substack{z/2}}}^{\substack{1,0 \\ \substack{0,2} \\ \substack{z/2}}}\left(\frac{z}{2} \mid \begin{matrix} 0, 1/2 \end{matrix}\right) \iff$ Cosine function $\frac{\cos(\sqrt{2z})}{\sqrt{\pi}} \implies f_{1002}(z, s) = \frac{2^s z^{-s} \Gamma(s)}{\Gamma\left(\frac{1}{2}-s\right)}$
- $G_{\substack{1,0 \\ \substack{0,2} \\ \substack{z/2}}}^{\substack{1,0 \\ \substack{0,2} \\ \substack{z/2}}}\left(\frac{z}{2}, 1/2 \mid \begin{matrix} 0, 1/2 \end{matrix}\right) \iff$ Cosine function $\frac{\cos(z)}{\sqrt{\pi}} \implies f_{1002}(z, s) = \frac{2^{s-1} z^{-s} \Gamma\left(\frac{s}{2}\right)}{\Gamma\left(\frac{1}{2}-\frac{s}{2}\right)}$
- $G_{\substack{1,0 \\ \substack{0,2} \\ \substack{z}}}^{\substack{1,0 \\ \substack{0,2} \\ \substack{z}}}\left(z \mid \begin{matrix} 0 \\ 0, 1/2 \end{matrix}\right) \iff$ Imaginary error function $\operatorname{erfi}(\sqrt{z}) \implies f_{1002}(z, s) = \frac{z^{-s} \Gamma(1-s) \Gamma(s)}{\Gamma\left(\frac{1}{2}-s\right)}$
- $G_{\substack{0,1 \\ \substack{1,0} \\ \substack{z}}}^{\substack{0,1 \\ \substack{1,0} \\ \substack{z}}}\left(z \mid 1-a\right) \iff$ Exponential function $e^{-1/z} z^{-a} \implies f_{0110}(z, s) = \frac{z^{-s} \Gamma(s) \Gamma(a-s)}{\Gamma(1-s)}$
- $G_{\substack{1,0 \\ \substack{0,1} \\ \substack{z}}}^{\substack{1,0 \\ \substack{0,1} \\ \substack{z}}}\left(z \mid 1-a\right) \iff$ Exponential function $e^{-z} z^{1-a} \implies f_{1001}(z, s) = z^{-s} \Gamma(s-1)$ with $a = -2$ (*Example 3*)
- $G_{\substack{1,1 \\ \substack{1,1} \\ \substack{z}}}^{\substack{1,1 \\ \substack{1,1} \\ \substack{z}}}\left(z \mid \begin{matrix} 1-a \\ b \end{matrix}\right) \iff z^{-b} (z+1)^{-a-b} \Gamma(a+b) \implies f_{1111}(z, s) = z^{-s} \Gamma(a-s) \Gamma(b+s)$
- $G_{\substack{0,2 \\ \substack{2,0} \\ \substack{z}}}^{\substack{0,2 \\ \substack{2,0} \\ \substack{z}}}\left(z \mid a, b\right) \iff \pi \left(z^{\frac{a-b}{2}+b-1} \csc(\pi(b-a)) I_{a-b}\left(\frac{2}{\sqrt{z}}\right) + z^{\frac{b-a}{2}+a-1} \csc(\pi(a-b)) I_{b-a}\left(\frac{2}{\sqrt{z}}\right) \right) \implies f_{0220}(z, s) = z^{-s} \Gamma(-a-s+1) \Gamma(-b-s+1)$
- $G_{\substack{1,1 \\ \substack{1,2} \\ \substack{z}}}^{\substack{1,1 \\ \substack{1,2} \\ \substack{z}}}\left(z \mid \begin{matrix} a \\ b, c \end{matrix}\right) \iff z^b \Gamma(-a+b+1) {}_1\tilde{F}_1(-a+b+1; b-c+1; -z) \implies f_{1112}(z, s) = \frac{z^{-s} \Gamma(-a-s+1) \Gamma(b+s)}{\Gamma(-c-s+1)}$

$$\begin{aligned}
\bullet G_{3,2}^{1,1}\left(z \left| \begin{array}{c} \mathbf{a1}, \mathbf{a2}, \mathbf{a3} \\ \mathbf{b1}, \mathbf{b2} \end{array} \right. \right) &\Longleftrightarrow \frac{z^{\mathbf{a1}-1} \Gamma(-\mathbf{a1}+\mathbf{b1}+1) {}_2\tilde{F}_2\left(-\mathbf{a1}+\mathbf{b1}+1, -\mathbf{a1}+\mathbf{b2}+1; -\mathbf{a1}+\mathbf{a2}+1, -\mathbf{a1}+\mathbf{a3}+1; \frac{1}{z}\right)}{\Gamma(\mathbf{a1}-\mathbf{b2})} \Rightarrow f_{1132}(z, s) = \frac{z^{-s} \Gamma(-\mathbf{a1}-s+1) \Gamma(\mathbf{b1}+s)}{\Gamma(\mathbf{a2}+s) \Gamma(\mathbf{a3}+s) \Gamma(-\mathbf{b2}-s+1)} \\
\bullet G_{2,4}^{2,1}\left(z \left| \begin{array}{c} \mathbf{a1}, \mathbf{a2} \\ \mathbf{b1}, \mathbf{b2}, \mathbf{b3}, \mathbf{b4} \end{array} \right. \right) &\Longleftrightarrow \frac{\pi z^{\mathbf{b2}} \Gamma(-\mathbf{a1}+\mathbf{b2}+1) \csc(\pi \mathbf{b1}-\pi \mathbf{b2}) {}_2F_3(-\mathbf{a1}+\mathbf{b2}+1, -\mathbf{a2}+\mathbf{b2}+1; -\mathbf{b1}+\mathbf{b2}+1, \mathbf{b2}-\mathbf{b3}+1, \mathbf{b2}-\mathbf{b4}+1; -z)}{\Gamma(\mathbf{a2}-\mathbf{b2}) \Gamma(-\mathbf{b1}+\mathbf{b2}+1) \Gamma(\mathbf{b2}-\mathbf{b3}+1) \Gamma(\mathbf{b2}-\mathbf{b4}+1)} + \\
&\quad \frac{\pi z^{\mathbf{b1}} \Gamma(-\mathbf{a1}+\mathbf{b1}+1) \csc(\pi (\mathbf{b2}-\mathbf{b1})) {}_2F_3(-\mathbf{a1}+\mathbf{b1}+1, -\mathbf{a2}+\mathbf{b1}+1; \mathbf{b1}-\mathbf{b2}+1, \mathbf{b1}-\mathbf{b3}+1, \mathbf{b1}-\mathbf{b4}+1; -z)}{\Gamma(\mathbf{a2}-\mathbf{b1}) \Gamma(\mathbf{b1}-\mathbf{b2}+1) \Gamma(\mathbf{b1}-\mathbf{b3}+1) \Gamma(\mathbf{b1}-\mathbf{b4}+1)} \\
&\Rightarrow f_{2124}(z, s) = \frac{z^{-s} \Gamma(-\mathbf{a1}-s+1) \Gamma(\mathbf{b1}+s) \Gamma(\mathbf{b2}+s)}{\Gamma(\mathbf{a2}+s) \Gamma(-\mathbf{b3}-s+1) \Gamma(-\mathbf{b4}-s+1)}
\end{aligned}$$

■ (2) Useful functions in the context `System`MeijerGDump``

Due to private correspondence with **O. Marichev** (June 2016) there are additional functions available in the context `System`MeijerGDump`` which are useful for a closer investigation/understanding of the evaluation of Meijer G-functions.

■ `MeijerGInfo`

This function extracts from the Meijer G-function the index list $\{\mathbf{m}, \mathbf{n}, \mathbf{p}, \mathbf{q}\}$:

```
System`MeijerGDump`MeijerGInfo[MeijerG[{{1/2}, {}}, {{3, -3}, {}}, 2 z]]
```

```
{m,n,p,q} = {2, 1, 1, 2}
```

```
MeijerG is entire function defined by sums over its LEFT poles in powers of variable
```

```
Attributes[MeijerGInfo] = {HoldAll}
```

```

HoldPattern[MeijerGInfo[MeijerG[{a_List, b_List}, {c_List, d_List}, x_, r_:1]] :=
Module[{m, n, p, q}, {m, n, p, q} = {Length[c], Length[a], Length[Join[a, b]], Length[Join[c, d]]};
Print["{m,n,p,q} = ", {m, n, p, q}];
If[2 (m+n) ≤ p+q, Print["Integral via L(i*∞) contour in definition of MeijerG does not exist"] ];
If[p < q, Print["MeijerG is entire function defined by sums over its LEFT poles in powers of variable"];];
If[p > q, Print["MeijerG is entire function defined by sums over its RIGHT poles in powers of 1/variable"];];

If[p == q, Print["MeijerG is defined by sum over LEFT poles INSIDE unit circle and by sum over RIGHT poles OUTSIDE of it"]];
If[m+n > p, Print["There exists analyticity cone inside which both definition are equivalent."];
If[NumericQ[x], If[Abs[Arg[x]] < π r (m+n-p), Print[x, " is inside of it"],
Print[x, " is outside of it"]
], Print["Cone is defined by ", Abs[Arg[x]] < π r (m+n-p)];

];

If[m+n == p, Print["There exists NO analyticity cone. MeijerG is piecewise analytic."];];
If[m+n < p, Print["special case. MeijerG is not defined by L(i*∞) contour"] ]

];
];

```

■ SlaterForm

This function constructs from the input form of Meijer G-function the integrand for the contour integral consisting of Γ -functions e.g.

```
System`MeijerGDump`SlaterForm[MeijerG[{{1/2}, {}}, {{3, -3}, {}}, 2 z], s] // tF
```

$$2^{-s} z^{-s} \Gamma\left(\frac{1}{2} - s\right) \Gamma(s-3) \Gamma(s+3)$$

```

Attributes[SlaterForm] = {HoldAll} ;

SlaterForm[MeijerG[{a_List, b_List}, {c_List, d_List}, z_, r_: 1], s_] :=
Module[{coeff, ap, am, bp, bm},
  {coeff, ap, am, bp, bm} = Simplify[System`MeijerGDump`sortG[c, 1 - a, b, 1 - d]];
  (r coeff z-s Times @@ Gamma[ap + r s] Times @@ Gamma[am - r s]) /
    (Times @@ Gamma[bp + r s] Times @@ Gamma[bm - r s])
] /; NumericQ[r] && (Positive[r] || Negative[r])

```

The auxiliary function **sortG** extracts from the input form of Meijer G-function the index lists $\{\text{coeff}, a_p, a_m, b_p, b_m\}$ used for evaluation of products of Γ -functions.

```

coeffList = {a → {1/2}, b → {}, c → {3, -3}, d → {}};
{coeff, ap, am, bp, bm} = Simplify[System`MeijerGDump`sortG[c, 1 - a, b, 1 - d] /. coeffList]

{1, {-3, 3}, {1/2}, {}, {}}

```

■ MeijerGToSums

The function **MeijerGToSums** shows the summands of the *infinite sums* (of left poles) which are not evaluated due to **Hold**. Applying **ReleaseHold** evaluates the sums.

```

expr1 = System`MeijerGDump`MeijerGToSums[MeijerG[{{1/2}, {}}, {{3, -3}, {}}, 2 z], s]
expr2 = expr1 // ReleaseHold

```

$$\begin{aligned}
& \text{Hold}[\text{Sum}] \left[\frac{(-1)^s 2^{-3+s} z^{-3+s} \text{Gamma}[6-s] \text{Gamma}\left[-\frac{5}{2}+s\right]}{\text{Gamma}[1+s]}, \{s, 0, 5\} \right] + \\
& \text{Hold}[\text{Sum}] \left[\left(2^{3+s} z^{3+s} \text{Gamma}\left[\frac{7}{2}+s\right] \left(-\text{Log}[2z] + \text{PolyGamma}[0, 1+s] - \text{PolyGamma}\left[0, \frac{7}{2}+s\right] + \text{PolyGamma}[0, 7+s] \right) \right) / \right. \\
& \quad \left. (\text{Gamma}[1+s] \text{Gamma}[7+s]), \{s, 0, \infty\} \right] \\
& - \frac{\sqrt{\pi}}{3} - \frac{8\sqrt{\pi}}{z^3} - \frac{8\sqrt{\pi}}{z^2} - \frac{3\sqrt{\pi}}{z} + \frac{\sqrt{\pi}z}{24} - \frac{\sqrt{\pi}z^2}{40} - \\
& \frac{1}{240z^3} \sqrt{\pi} \left(-1920 - 1920z - 720z^2 - 80z^3 + 10z^4 - 6z^5 + 240e^z z^3 \text{BesselI}[3, z] \text{Log}[2z] + \right. \\
& \quad 240e^z z^3 \text{BesselI}[3, z] \text{PolyGamma}\left[0, \frac{7}{2}\right] - 16 \text{Hypergeometric1F1Regularized}^{(0,1,0)}\left[-\frac{5}{2}, -5, 2z\right] + \\
& \quad \left. 3600z^6 \text{Hypergeometric1F1Regularized}^{(0,1,0)}\left[\frac{7}{2}, 7, 2z\right] + 5z^6 \text{Hypergeometric1F1}^{(1,0,0)}\left[\frac{7}{2}, 7, 2z\right] \right)
\end{aligned}$$

With the rule $\{\text{Hold}[\text{Sum}][\mathbf{x}_-, \text{range}_-] \rightarrow \sum_{\text{range}} @ \mathbf{x}\}$ the sums together with their index range are rewritten in a more comprehensive form.

```

Attributes[MeijerGToSums] = {HoldAll}

HoldPattern[MeijerGToSums[MeijerG[{a_List, b_List}, {c_List, d_List}, x_, r_: 1], k_Symbol]] :=
System`MeijerGDump`TideUpSummand[
Module[{res1, res2, Sum1, var},
Catch[HypergeometricLogDump`MeijerGAMain ;
Block[{HypergeometricLogDump`$HypPFQRegularized = PFQRegularizedFormalSum[k, Sum1]},
var = Hold[Sum];
res1 = Block[{HypergeometricLogDump`$HeadToReplaceSumAWith = Sum1,
$SumLeftPoles = True},
SumLeftPoles[{a, b}, {c, d}, x, r] ];
If[Length[a] + Length[b] == Length[c] + Length[d],
res2 = Block[{HypergeometricLogDump`$HeadToReplaceSumAWith = Sum1,
$SumLeftPoles = False},
SumLeftPoles[{1 - c, 1 - d}, {1 - a, 1 - b},  $\frac{1}{x}$ , r] /.
{Log[ $\frac{1}{x}$ ] → -Log[x], ( $\frac{1}{x}$ )w → x-w} ];
Refine[Piecewise[{{res1, Abs[x] ≤ 1}, {res2, Abs[x] > 1}}] /. {Sum1 → var} ],
Throw[res1 /. {Sum1 → var} ]
]]]] /. {Hold[Sum][body_, {it_, par___}] => Hold[Sum][body /. {it → k}, {k, par}]}], k
] /; ! InexactNumberQ[x] && NumericQ[r] && Positive[r]

```

Auxiliary functions needed in **MeijerGToSums** are **TideUpSummand** and **SumLeftPoles**; from the context **HypergeometricLogDump`** there are **MeijerGAMain**, **\$HypPFQRegularized** and **\$HeadToReplaceSumAWith**

■ (3) Change of integration variables methods

(1) Change of variables using `changeVariable4Integral`

The *change of variables* $z \rightarrow \zeta$ ($= z2\zeta$) (and its inverse $\zeta \rightarrow z$ ($= \zeta2z$)) can be applied to the following type of integral $\int_{z_a}^{z_b} f(z) dz$; of course one must transform the *integrand* $f(z)$ taking into account the Jacobian corresponding to the change of variables, the *differential* dz and the lower and upper limits of the integral $\{z_a, z_b\}$ e.g. $\{0, \infty\}$. This is achieved with the procedure `changeVariable4Integral`

```
Clear[fz, z, za, zb, z2ζ, ζ2z, ζ];
SetAttributes[changeVariable4Integral, HoldFirst];

changeVariable4Integral[ $\int_{z_a}^{z_b} fz\_ dz\_ , \{\zeta\_ , z2\zeta\_ , \zeta2z\_ \}$ ] :=
    Integrate[(fz /. {z -> z2ζ}) * D[z2ζ, ζ], {ζ, ζ2z /. {z -> za}, ζ2z /. {z -> zb}}];
```

implemented in the package `Contour Integration`. [23].

? `changeVariable4Integral`

For example, with the replacement rules $z2\zeta: z \rightarrow (\zeta + i \frac{\alpha}{2\beta})$ and $\zeta2z: \zeta \rightarrow (z - i \frac{\alpha}{2\beta})$ the integral becomes :

```
Clear[z, z, α, β];
$Assumptions = {α ∈ Reals, β ∈ Reals, β > 0};
changeVariable4Integral[ $\int_{-\infty}^{\infty} e^{i\alpha z - \beta z^2} dz , \{\zeta, (\zeta + i \frac{\alpha}{2\beta}), (z - i \frac{\alpha}{2\beta})\}$ ] // fs
```

$$\frac{e^{-\frac{\alpha^2}{4\beta}} \sqrt{\pi}}{\sqrt{\beta}}$$

similarly, with $z\zeta: x \rightarrow r e^{i \frac{2\pi}{3}}$ and its reverse $\zeta z: r \rightarrow x e^{-i \frac{2\pi}{3}}$ the integral is :

$$\text{changeVariable4Integral}\left[\int_0^\infty \frac{1}{1+x^3} dx, \left\{r, r e^{i \frac{2\pi}{3}}, x e^{-i \frac{2\pi}{3}}\right\}\right]$$

$$\frac{2\pi}{3\sqrt{3}}$$

(2) Change of variables using SC package **SymbolicComputing**

An alternative approach makes use of the package **SymbolicComputing.mx** from Youngjoo Chung [22] which is available for *Mathematica* V9.0.1 and higher. It can be downloaded from the following URL

```
(* ToExpression[URLFetch["http://symbcomp.gist.ac.kr/downloads/InstallSymbCompPersonal.m"]] *)
(* installs package SymbolicComputing.mx in ... \AppData\Roaming\Mathematica\Applications *)
```

This powerful package (with more than 800 routines) can do much more than performing change of variables only. In order to avoid some infinite loop the SC package should be loaded first, thus remove the package **ContourIntegration`** then load package **SymbolicComputing`** :

```
removePackage["ContourIntegration`"]; (* remove ContourIntegration package temporarily *)
removePackage["Notation`"];
Unprotect["Global`*"];
Remove["Global`*"];
$ContextPath = $ContextPath /. {"Notation`" -> Sequence[], "ContourIntegration`" -> Sequence[] }
```

Then the SC package can be loaded without any problem.

In order to avoid conflicts with code contained in the main package **ContourIntegration`** the package **SymbolicComputing`** must be loaded first if procedures from the SC packages will be executed.


```
DefaultOperators:={OverHat[_],OverHat[_][__],Subscript[OverHat[_],__],Subscript[OverHat[_],__][__]};
Get["SymbolicComputing`"]//Timing      (* ~ 2 sec : loading successful *)
$SCVersion                             (* "Beta 3.1 (January 7, 2016)" *)
```

```
{2.69882, Null}
```

```
Beta 3.1 (January 7, 2016)
```

If the packages are loaded in reverse order then loading of the SC package will need up to 6 min. and additional warnings will occur.

```
Names["SC*"] // Length
$ContextPath;
```

```
814
```

The essential procedure **SCMAF**[*integral_*,*transInt_*, *replVar_List*, *evalInt*, *opts___*:{ }] together with two more **SC**-routines **SCTransInt** and **SCEvalInt** performs the *transformation of variables* and does a *stepwise* calculation of the integral. (Private communications with Youngjoo Chung 9/2013). For further details use the Help Browser for AddOns : see SymbolicComputing > Function Analysis > SCMAF

```
? SCMAF SCTransInt SCEvalInt
```

SCMAF[*expr*, *f*₁, {*x*₁₁, *x*₁₂, ...}, *f*₂, {*x*₂₁, *x*₂₂, ...}, ...] makes multiple replacements $x_{11} \rightarrow f_1[x_{11}, x_{12}, \dots]$, $x_{21} \rightarrow f_2[x_{21}, x_{22}, \dots]$, etc. See the help browser for more details.

SCTransInt[*expr*] transforms integrals.

SCTransInt[*expr*, *rules*] transforms integrals by applying *rules*.

SCEvalInt[*expr*] evaluates integrals expressed with SCIntegrate.

SCEvalInt[*expr*, *var*₁, *var*₂, ...] evaluates integrals expressed with SCIntegrate for the variables *var*_{*i*}.

```
Clear[fz, z, za, zb, ξ, φ];
SCMAF[ $\int_{za}^{zb} fz\_ dz\_ , SCTransInt, \{All, TransVar \rightarrow \{z, \xi, z == f[\xi]\}\}, SCEvalInt, All, RA,$ 
      Comment  $\rightarrow$  "Upper/lower limits will be changed" ]
```

With the help of the procedure **SCMAF** the evaluation of the integrals requiring change of integration variable can be performed :

The setting of the parameter ϕ can be explicitly given in terms of a replacement rule with $RA \rightarrow \phi == \frac{2\pi}{3}$

```
$Assumptions = { φ ∈ Reals && φ > 0};
```

```
SCMAF[ $\int_0^\infty \frac{1}{1+z^3} dz , SCTransInt, \{All, TransVar \rightarrow \{z, r, z == r e^{i\phi}\}\}, SCEvalInt, All, RA \rightarrow \phi == \frac{2\pi}{3} ]$ 
```

$$\int_0^{\infty} \frac{e^{i\phi}}{1 + e^{3i\phi} r^3} dr$$

$$\frac{2\pi}{3\sqrt{3}}$$

and

```
$Assumptions = { {α, β} ∈ Reals && β > 0};
```

```
SCMAF[ $\int_{-\infty}^\infty e^{i\alpha z - \beta z^2} dz , SCTransInt, \{All, TransVar \rightarrow \{z, \xi, z == \left(\xi + i \frac{\alpha}{2\beta}\right)\}\}, SCEvalInt, All ]$ 
```

$$\int_{-\infty}^{\infty} e^{i \alpha \left(\frac{i \alpha}{2 \beta} + \zeta \right) - \beta \left(\frac{i \alpha}{2 \beta} + \zeta \right)^2} d\zeta$$

$$\frac{e^{-\frac{\alpha^2}{4 \beta}} \sqrt{\pi}}{\sqrt{\beta}}$$

The **SC** package's *own* interpreter for 2d forms of derivatives, integrals etc.

$$\text{SCMAF}\left[\frac{d(\mathbf{x} + \mathbf{x}_0)^m}{d\mathbf{x}}, \text{SCEvalDeriv}, \text{All}, \text{Hold} \rightarrow \mathbf{x}_0\right]$$

$$m (\mathbf{x} + \mathbf{x}_0)^{-1+m}$$

$$\text{SCMAF}\left[\int_0^{\infty} \sin[z^3] dz, \text{SCTransInt}, \{\text{All}\}, \text{SCEvalInt}, \text{All}\right]$$

$$\int_0^{\infty} \sin[z^3] dz$$

$$\frac{1}{6} \text{Gamma}\left[\frac{1}{3}\right]$$

but the usual 2d form of an integral will not be ignored (in the context of the SC package).

$$\int_0^{\infty} \sin[z^3] dz$$

$$\int_0^{\infty} \sin[z^3] dz$$

In order to avoid any conflict with subscripted variables used in the packages **SymbolicComputing`** and **ContourIntegration`** the procedure **SCEnableNotation[False]** will *remove* all symbols associated with the **SC** package and the 2d interpreter of the SC package is *disabled* so that

the usual 2d notation of the *Mathematica* FrontEnd is supported and the calculation with the *Mathematica* kernel done again.

? SCEnableNotation

SCEnableNotation[enable] sets whether to enable the package's own interpretation of the 2–D form of integrals, products, sums, partial derivatives ($\partial_x f$), intersections (\cap) and unions (\cup).

SCEnableNotation[False];

$$\int_0^\infty \sin[z^3] \, dz$$

$$\frac{1}{6} \operatorname{Gamma}\left[\frac{1}{3}\right]$$

Thus, if one wants to make use of the package **ContourIntegration`** again the SC package must be removed in order not to interfere with the package **ContourIntegration`** :

```
removePackage["SymbolicComputing`"]
$ContextPath = $ContextPath /. "SymbolicComputing`" -> Sequence[]
```

SymbolicComputing` * ; SymbolicComputing`Private` *

package SymbolicComputing` was removed

```
{ContourIntegration`, Notation`, Units`, CCompilerDriver`, ResourceLocator`,
 StreamingLoader`, IconizeLoader`, CloudObjectLoader`, PacletManager`, System`, Global`}
```

Reload the package ContourIntegration` (after removal of the SC package) :

```
Clear["Global`*"];
SetDirectory[NotebookDirectory[]];
Get["ContourIntegration`"]
```

■ Acknowledgement

The author wants to thank *Oleg Marichev*/WRI and *Albert Tamazyan* for the help for better understanding of subtle problems related to Meijer G-functions and to allude to some new functions in the context `System`MeijerGDump``. Concerning the problem of a consistent treatment of change of variables in integration the patient support of *Youngjoo Chung*/GIST [22] is gratefully acknowledged to provide the author with the latest version of his package `SymbolicComputing.mx` and give additional tips how to make effective usage of the relevant routine therein. In the same context thanks to various contributors to the *MathGroup* thread [mg131569] “Change of Variables for Integrals” [21]. It should also be noticed that *David Park* had drawn attention to his `Presentations`StudentsIntegral.m` package which provides another approach to the same problem of change of variables as regards to integration. And, finally, the author is particularly grateful to *Michael Trott*/WRI for his tireless competent support answering - over the period where this project became mature - numerous questions and gave profound advice to subtle technical problems in *Mathematica*.

■ References

- [1] **Robert Kragler** "Contour Integration; a Procedure still Missing in Mathematica" Lecture at XVII. Mathematica-Tag, (18. Febr. 2016) WIAS/Berlin
- [2] **Robert Kragler** "Contour Integration - or what is still missing in Mathematica, Part1 : Residues and Contour Integration" (2016)
- [3] **Robert Kragler** "Contour Integration - a Procedure still Missing in Mathematica" ACA 2016, (4 Aug. 2016) Kassel/Germany
- [4] **Michael Trott & Oleg Marichev** from The Wolfram Functions Site : <http://functions.wolfram.com>
- [5] **Eric W. Weisstein** "CRC ENCYCLOPEDIA Of MATHEMATICS" 3rd Ed. Chapman & Hall/CRC Press (2009) Vol. 1 p. 655
- [6] **Frank W.J. Olver, Daniel W. Lozier, Ronald F. Boisvert & Charles W. Clark** "NIST Handbook of Mathematical Functions", Cambridge Univ. Press (2010) ISBN 978-0-521-14063-8, Chapt. 5.12, p. 142

- Wilhelm Forst & Dieter Hoffmann** "Funktionentheorie erkunden mit Maple", Springer-Verlag (2002), ISBN 3-540-42543-8, Chapt. 4.1, pp. 116-117
- [8] **Michael Trott** "The Mathematica Guidebook for Symbolics", Springer-Verlag (2006), ISBN 0-387-95020-6, Solutions 7a, pp. 1245-1248
- [9] **Wikipedia** "Point in polygon", URL http://en.wikipedia.org/wiki/Point_in_polygon;
Daniel Lichtblau in MathGroup Archive 2009 : [mg96475] "Testing if a point is inside a polygon" (Feb. 2009);
 see also the discussion by **Adriano Pascoletti**, **Rafael Torres Carot**, **David Park** and **Paul Wellin**,
 URL : <http://www.mathematica-users.org/web/Mathematica/wiki/wiki.jsp?pageName=Notebook;PointInsidePolygon.nb>
- [10] **M. Shimrat** "Algorithm 112: Position of point relative to polygon" (1962) , Commun. ACM Vol. 5, issue 8 (Aug. 1962)
- [11] **Ivan Sutherland et al.** "A Characterization of Ten Hidden-Surface Algorithms" (1974), ACM Computing Surveys Vol. 6, no. 1
- [12] **Paul Wellin** "Programming with Mathematica; An Introduction" (2013) Cambridge Univ. Press, ISBN 978-1-107-00946-2 , pp. 419-427
- [13] **Wikipedia** "Winding number", URL : https://en.wikipedia.org/wiki/Winding_number
- [14] **Michael Trott** / WRI (private communication Oct. 2013)
- [15] **David Park** in MathGroup Archive 2000 : [mg25350] "Point inside a polygon" (Sept. 2000)
- [16] **Todd Rowland & Eric Weisstein** from MathWorld, URL : <http://mathworld.wolfram.com/Inside-OutsideTheorem.html>
- [17] **Thies Heidecke** in Mathematica Stack Exchange "How to check if a 2D point is in a polygon?" (Aug. 2012)
- [18] **Roberto Luigi Brambilla** (private communication Oct. 2013) and N. I. Muskhelishvili "*Singular Integral Equation*" Dover (1992), pp. 127
- [19] **Eric W. Weisstein, Michael Trott** from The Wolfram Functions Site : <http://functions.wolfram.com/notebooks/SpecialFunctions/MeijerG-Function.nb>
- [20] **A. P. Prudnikov, Yu. A. Brychkov & O. I. Marichev** "*Evaluation of Integrals and the Mellin Transform*" Itogi Nauki i Tekhniki, Seriya Matemat. Analiz (1998) Vol. 27, pp. 3-146;
A. P. Prudnikov, O. I. Marichev & Yu. A. Brychkov "*Integrals and Series. Vol. 3: More Special Functions.*" Gordon and Breach (1990), Chapt. 8.2
 "The Meijer G-Function $G_{p,q}^{m,n} \left(z \left| \begin{smallmatrix} (a_p) \\ (b_q) \end{smallmatrix} \right. \right)$ ", pp. 617-626
Lucy L Slater "*Generalized hypergeometric functions*" Cambridge Univ. Press (1966), ISBN 978-0-521-09061-2, Chapt. 4

Discussions in *MathGroup* : **Alex Krasnov** [mg131613] and **Alexander Elkins** [mg131572]

[22] **Youngjoo Chung** *MathGroup* [mg131582] with **SymbolicComputing`** package presented at WTC2011 (see <https://symbcomp.gist.ac.kr>).
David Park *MathGroup* [mg131569] with **Presentations`StudentsIntegral** package

[23] Private communications: **Michael Trott** / WRI (9/2013)

[24] **Sadri Hassani** "Mathematical Physics; A Modern Introduction to Its Foundations", Springer-Verlag (1999), ISBN 978-3-642-87431-4, Chapt. 10.3.9, pp. 283-285

[25] **Arnold Sommerfeld** "*Atombau und Spektrallinien*", Verlag Harri Deutsch (1978), ISBN 3-87144-484-7
p.655 - 659 'Mathematische Zusätze und Ergänzungen', (4.) Ausführung einiger Integrale auf komplexem Wege.
Ludwig Waldmann "Quantentheorie" Vorlesung, WS 1965/66 University of Erlangen

[26] **Herbert Goldstein, Charles P. Poole Jr. & John L. Safko** "*Classical Mechanics*" 3rd Ed. , Pearson Education, Inc. (2002), Chapt. 10.8,
'The Kepler Problem in Action and Angle Variables', p, 505-516

About the Author

Robert Kragler holds a doctoral degree in theoretical physics from University of Frankfurt/M. (1974) and a Dr. habil. from University of Constance (1981). He is professor from Weingarten University of Applied Sciences where he retired in 2008. He uses *Mathematica* since 1989 for his research and applied computer algebra methods particularly in his lectures courses on calculus and quantum physics .

Robert Kragler

Weingarten University of Applied Sciences

D-88241 Weingarten, Germany

kragler@hs-weingarten.de

<http://portal.hs-weingarten.de/web/kragler/mathematica>

Download : <http://portal.hs-weingarten.de/web/kragler/Mathematica/> > Contour_Integration

Mma files (V10) : **ContourIntegration_P1.nb, ContourIntegration_P2.nb, ContourIntegration.m**
MmaDay_WIAS2016_ContourIntegration.nb, ACA2016_ContourIntegration_P2.nb