On the computation of confluent hypergeometric functions for large imaginary part of parameters b and z

Guillermo Navas-Palencia [1,2] Argimiro Arratia [1]

[1]Dept. Computer Science, Universitat Politècnica de Catalunya, Spain

[2]Numerical Algorithms Group Ltd, United Kingdom

The 5th International Congress on Mathematical Software
Berlin 2016

Overview

1. Introduction
   - Confluent hypergeometric functions
   - Applications
2. Algorithm
   - The steepest descent method
   - $U(a, b, z), \Im(z) \to \infty$
   - $U(a, b, z), \Im(b) \to \infty$
   - ${}_1F_1(a; b; z), \Im(z) \to \infty$
   - ${}_1F_1(a; b; z), \Im(b) \to \infty$
   - Numerical quadrature methods
3. Numerical examples
   - Numerical examples: ${}_1F_1(a; b; z)$
   - Numerical examples: $U(a, b, z)$
4. Concluding remarks

Confluent hypergeometric functions

The standard solutions of Kummer's equation are the confluent hypergeometric functions of the first and second kind: ${}_1F_1(a;b;z)$ and $U(a,b,z)$

$$z\frac{d^2f(z)}{dz^2}+(b-z)\frac{df(z)}{dz}-af(z)=0$$

$U(a,b,z)\sim z^{-a},\ |z|\to\infty.$

Integral representations

$${}_1F_1(a;b;z)=\frac{\Gamma(b)}{\Gamma(a)\Gamma(b-a)}\int_0^1 e^{zt}t^{a-1}(1-t)^{b-a-1}\,dt,\quad \Re(b)>\Re(a)>0$$

$$U(a,b,z)=\frac{1}{\Gamma(a)}\int_0^\infty e^{-zt}t^{a-1}(1+t)^{b-a-1}\,dt,\quad \Re(a)>0,\ |\mathrm{ph}\,z|<\frac{1}{2}\pi$$

Kummer's transformations

$${}_1F_1(a;b;z)=e^z\,{}_1F_1(b-a;b;-z),\quad U(a,b,z)=z^{1-b}U(a-b+1,2-b,z)$$

Applications

Main Goal

- Reduce cancellation issues for complex parameters
- How?
  1. Integral representation + contour deformation
  2. Numerical quadrature methods

Applications - Statistics: Characteristic functions

- Beta distribution

$$\phi_X(t) = {}_1F_1(\alpha; \alpha + \beta; it), \quad \alpha, \beta > 0$$

- Standard Arcsine distribution

$$\phi_X(t) = {}_1F_1\left(\frac{1}{2}; 1; it\right)$$

- F–distribution

$$\phi_X(t) = \frac{\Gamma((p+q)/2)}{\Gamma(q/2)} U\left(\frac{p}{2}, 1 - \frac{q}{2}, -\frac{q}{p} it\right)$$

where $p, q > 0$, are the degrees of freedom

Applications - Finance

Modelling a beta-distributed loss given default in portfolio credit risk models

Computation of **the portfolio Fourier transform** with beta-distributed loss given default at time t

$$\hat{f}(t) = \int_{-\infty}^{\infty} \prod_{n=1}^{N} [1 - p_n(y) + p_n(y) {}_1F_1(\alpha; \alpha+\beta; -itE_n)] f(y)\, dy$$

where

- N: $\#$ assets in portfolio
- $p_n(y)$: default probability
- E_n: exposure at default
- $f(y)$: density function of the credit risk factor

Pricing Asian options

- F. D. Nieuwveldt. *A survey of computational methods for pricing Asian options*, (2009)

The steepest descent method

1-D oscillatory integral - ideal case

$$I := \int_{\alpha}^{\beta} f(x) e^{i\omega g(x)}\, dx$$

where

- $\omega > 0$: frequency parameter
- $f(x)$: amplitude, $g(x)$: oscillator; smooth real functions

Steepest descent method

- Substitution by unions of contours
- Non-oscillatory and exponentially decaying
- Path of steepest descent $h_x(p)$ parametrised by $p \in [0, \infty)$, solve:

$$g(h_x(p)) = g(x) + ip$$

$$\begin{aligned} I[f; h_x] &= e^{i\omega g(x)} \int_0^{\infty} f(h_x(p)) h_x'(p) e^{-\omega p}\, dp \\ &= \frac{e^{i\omega g(x)}}{\omega} \int_0^{\infty} f\left(h_x\left(\frac{q}{\omega}\right)\right) h_x'\left(\frac{q}{\omega}\right) e^{-q}\, dq \end{aligned}$$

The steepest descent method - continued

- Finite integral: $x \in [\alpha, \beta]$

$$I := \int_{\alpha}^{\beta} f(x) e^{i\omega g(x)}\, dx = I[f; h_\alpha] - I[f; h_\beta]$$

- Semi-infinite integral: $x \in [\alpha, \infty)$

$$I := \int_{\alpha}^{\infty} f(x) e^{i\omega g(x)}\, dx = I[f; h_\alpha] - 0$$

Particular case of interest: $g(x) = x \to h_x(p) = x + ip$

$$\int_{\alpha}^{\beta} f(x) e^{i\omega x}\, dx = \frac{ie^{i\omega\alpha}}{\omega} \int_0^{\infty} f\left(\alpha + i\frac{q}{\omega}\right) e^{-q}\, dq - \frac{ie^{i\omega\beta}}{\omega} \int_0^{\infty} f\left(\beta + i\frac{q}{\omega}\right) e^{-q}\, dq$$

$U(a, b, z)$, large imaginary z

Case 1: $U(a, b, z), \Im(z) \to \infty$

- Transform into a highly oscillatory integral

$$U(a,b,z) = \frac{1}{\Gamma(a)} \int_0^\infty e^{-zt} t^{a-1} (1+t)^{b-a-1} \, dt$$
$$= \frac{1}{\Gamma(a)} \int_0^\infty e^{-\Re(z)t} t^{a-1} (1+t)^{b-a-1} e^{-i\Im(z)t} \, dt$$

- where $g(t) = t$, $g'(t) = 1 \neq 0$ and there are no stationary points
- Steepest descent integral with one endpoint

Integral for $U(a, b, z)$, large imaginary z

$$U(a,b,z) = \frac{i}{\omega\Gamma(a)} \int_0^\infty e^{-\Re(z) i \frac{q}{\omega}} \left(i\frac{q}{\omega}\right)^{a-1} \left(1 + i\frac{q}{\omega}\right)^{b-a-1} e^{-q} \, dq \qquad (1)$$

$U(a, b, z)$, large imaginary b

Case 2: $U(a,b,z), \Im(b) \to \infty$

- Avoid singularity at $t = 0$
- Transformation into a highly oscillatory integral

$$U(a,b,z) = \frac{1}{\Gamma(a)} \int_0^\infty e^{-zt} t^{a-1} (1+t)^{b-a-1}\, dt$$
$$= \frac{e^z}{\Gamma(a)} \int_1^\infty e^{-zt} (t-1)^{a-1} t^{\Re(b)-a-1} e^{i\Im(b)\log(t)}\, dt$$

- Solve the path of steepest descent at $t = 1$ with $g(t) = \log(t)$

$$h_1(p) = e^{\log(1)+ip} = e^{ip} \quad \text{and} \quad h_1'(p) = ie^{ip}.$$

- Steepest descent integral with one endpoint and no further contributions

Integral for $U(a, b, z)$, large imaginary b

$$U(a,b,z) = \frac{ie^z}{\omega\Gamma(a)} \int_0^\infty e^{\phi(q,\omega)} (\mu(q,\omega)-1)^{a-1} \mu(q,\omega)^{\Re(b)-a-1} e^{-q}\, dq \qquad (2)$$

where $\mu(q,\omega) = i\frac{q}{\omega}$ and $\phi(q,\omega) = -ze^{\mu(q,\omega)} + \mu(q,\omega)$

${}_1F_1(a;b;z)$, large imaginary z

Case 3: ${}_1F_1(a;b;z), \Im(z) \to \infty$

- Transformation into a highly oscillatory integral

$$
\begin{aligned}
{}_1F_1(a;b;z) &= \frac{\Gamma(b)}{\Gamma(a)\Gamma(b-a)} \int_0^1 e^{zt} t^{a-1}(1-t)^{b-a-1}\, dt \\
&= \frac{\Gamma(b)}{\Gamma(a)\Gamma(b-a)} \int_0^1 e^{\Re(z)t} t^{a-1}(1-t)^{b-a-1} e^{i\Im(z)t}\, dt
\end{aligned}
$$

- where $g(t) = t$, $g'(t) = 1 \neq 0$ and there are no stationary points
- Apply transformation for particular case

Integral for ${}_1F_1(a;b;z)$, large imaginary z

$$
\begin{aligned}
{}_1F_1(a;b;z) = \frac{\Gamma(b)}{\Gamma(a)\Gamma(b-a)} \frac{i}{\omega} \Bigg[\int_0^\infty e^{\Re(z) i \frac{q}{\omega}} \left(i\frac{q}{\omega}\right)^{a-1} \left(1 - i\frac{q}{\omega}\right)^{b-a-1} e^{-q}\, dq \\
- e^{i\omega} \int_0^\infty e^{\Re(z)(1+i\frac{q}{\omega})} \left(1 + i\frac{q}{\omega}\right)^{a-1} \left(-i\frac{q}{\omega}\right)^{b-a-1} e^{-q}\, dq \Bigg] \qquad (3)
\end{aligned}
$$

${}_1F_1(a;b;z)$, large imaginary b

Case 4: ${}_1F_1(a;b;z), \Im(b) \to \infty$

- Option 1: Use connection formula, valid for all $z \neq 0$,

$$\frac{1}{\Gamma(b)}\,{}_1F_1(a;b;z) = \frac{e^{\mp\pi i a}}{\Gamma(b-a)}U(a,b,z) + \frac{e^{\pm\pi i(b-a)}}{\Gamma(a)}e^{z}U(b-a,b,ze^{\pm\pi i}) \qquad (4)$$

- Option 2: ${}_1F_1(a;b;z)$ can be written in the form

$$\begin{aligned} {}_1F_1(a;b;z) &= \frac{\Gamma(b)}{\Gamma(a)\Gamma(b-a)}\int_0^1 e^{zt}t^{a-1}(1-t)^{b-a-1}\,dt \\ &= \frac{\Gamma(b)}{\Gamma(a)\Gamma(b-a)}\int_0^\infty e^{-bt}(1-e^{-t})^{a-1}e^{at+z(1-e^{-t})}\,dt \\ &= \frac{\Gamma(b)}{\Gamma(a)\Gamma(b-a)}\int_0^\infty e^{-\Re(b)t}(1-e^{-t})^{a-1}e^{at+z(1-e^{-t})}e^{-i\Im(b)t}\,dt \end{aligned}$$

Case similar to $U(a,b,z)$ for large imaginary z

Numerical quadrature methods

Adaptive quadrature for oscillatory integrals

- Write integral in terms of its real and imaginary parts
- Two separate integrals with trigonometric weight

$$\int_0^1 f(t)e^{i\omega t}\,dt = \int_0^1 f(t)\cos(\omega t)\,dt + i\int_0^1 \sin(\omega t)\,dt$$

Example: ${}_1F_1(a;b;z)$ when $\Im(z)\to\infty$

$$ {}_1F_1(a;b;z) = \frac{\Gamma(b)}{\Gamma(a)\Gamma(b-a)}\left[\int_0^1 e^{\Re(z)t}t^{a-1}(1-t)^{b-a-1}\cos(\Im(z)t)\,dt \right. $$
$$ \left. +\; i\int_0^1 e^{\Re(z)t}t^{a-1}(1-t)^{b-a-1}\sin(\Im(z)t)\,dt\right] $$

- Not directly applicable to $U(a,b,z)$
- Use connection formula valid for $b \notin \mathbb{Z}$ and $z \neq 0$

$$U(a,b,z) = \frac{\Gamma(1-b)}{\Gamma(a-b+1)}\,{}_1F_1(a;b;z) + \frac{\Gamma(b-1)}{\Gamma(a)}\,{}_1F_1(a-b+1;2-b;z) \quad (5)$$

- Apply recurrence relations on ${}_1F_1(a-b+1;2-b;z)$

Numerical quadrature methods

Gauss-Laguerre quadrature - The numerical steepest descent method (NSD)

- D. Huybrechs and S. Vandewalle. *On the evaluation of highly oscillatory integrals by analytic continuation*, (2006)
- Semi-infinite integrals with exponential decaying
- Applying a Gauss-Laguerre quadrature rule with n points x_k and weights w_k yields a quadrature rule

$$I[f;h_x] \approx Q[f;h_x] := \frac{e^{i\omega g(x)}}{\omega} \sum_{k=1}^{n} w_k f\left(h_x\left(\frac{x_k}{\omega}\right)\right) h'_x\left(\frac{x_k}{\omega}\right)$$

- Approximation error behaves asymptotically as $\mathcal{O}(\omega^{-2n-1})$ as $\omega \to \infty$

Example: $U(a,b,z)$ when $|z| \to \infty$

- Asymptotic expansion for $U(a,b,z)$

$$U(a,b,z) \sim z^{-a} \sum_{n=0}^{\infty} \frac{(a)_n (a-b+1)_n}{n!(-z)^n}, \quad |\mathrm{ph}\, z| < \frac{3}{2}\pi$$

- The error behaves asymptotically as $\mathcal{O}(z^{-n-1})$
- Asymptotic order practically doubled using Gauss-Laguerre

Numerical examples: ${}_1F_1(a; b; z)$

Benchmark

- **CONHYP**: M. Nardin, W. F. Perger, and A. Bhalla. *Algorithm 707. CONHYP : A numerical evaluator of the confluent hypergeometric function for complex arguments of large magnitudes*, (1992).
- **ZJ**: S. Zhang and J. Jin. *Computation of special functions*, (1996)
- Codes in `Fortran 90`. Compiler `gfortran 4.9.3`
- **NSD**: Prototype in `Python` using `Scipy`

${}_1F_1(a,b,z)$	CONHYP	ZJ	NSD	N
$(1,4,50i)$	3.96e−13/4.29e−18i	**1.50e−15/4.28e−18*i***	**1.15e−16/1.11e−16*i***	2
$(3,10,30+100i)$	1.27e−13/1.28e−13i	**6.83e−17/1.07e−14*i***	**2.48e−17/1.24e−14*i***	25
$(15,20,200i)$	9.20e−13/9.20e−13i	E	**8.43e−16/7.93e−16*i***	25
$(400,450,1000i)$	8.32e−12/1.00e−11i	–	1.37e−12/1.02e−13i	50
$(2,20,50-2500i)$	1.35e−11/1.35e−11i	7.30e−11/2.10e−09i	**4.75e−16/6.41e−16*i***	20
$(500,510,100-1000i)$	4.10e−13/3.68e−12i	–	**4.71e−13/3.11e−16*i***	50
$(2,20,-20000i)$	–	5.79e−10/7.99e−07i	**5.92e−16/3.62e−14*i***	10
$(900,930,-10^{10}i)$	–	–	6.78e−13/6.77e−13i	20
$(4000,4200,50000i)^*$	–	–	6.04e−12/5.99e−12i	80

Table: Relative errors for routines computing the confluent hypergeometric function for complex argument. N: number of Gauss-Laguerre quadratures. $(*)$: precision in `mpmath` increased to 30 digits. (E): convergence to incorrect value. $(-)$: overflow.

Numerical examples: $U(a,b,z)$

Benchmark

- CPU MATLAB R2013a (`hypergeom`)

${}_1F_1(a;b;z)$	MATLAB R2013a	NSD	N
$(2, 20, -20000i)$	1.509 (0.068)	0.033	10
$(900, 930, -10^{10}i)$	5.594 (0.739)	0.035	20
$(4000, 4200, 50000i)$	488.384(18.127)	0.043	80

Table: Comparison in terms of CPU time. MATLAB second evaluation in parenthesis. Intel(R) Core(TM) i5-3317U CPU @ 1.70GHz.

- Large test set

Function	Min	Max	Mean
$U(a,b,iz)$	1.97e−18/2.04e−17i	9.97e−13/2.50e−11i	1.34e−14/6.94e−14i
$U(a,ib,z)$	6.57e−18/6.17e−18i	1.49e−11/8.55e−12i	1.38e−13/1.43e−13i

Table: Error statistics for $U(a,b,iz)$ and $U(a,ib,z)$ using $N=100$ quadratures.

- 13-14 digits of precision in real and imaginary part typically achieved

Numerical examples: $U(a, b, z)$

Benchmark

- Large test set : summary
  - Error in $U(a, ib, z)$ exhibits oscillation pattern
  - Similar error expected in ${}_1F_1(a; ib; z)$ with option 1

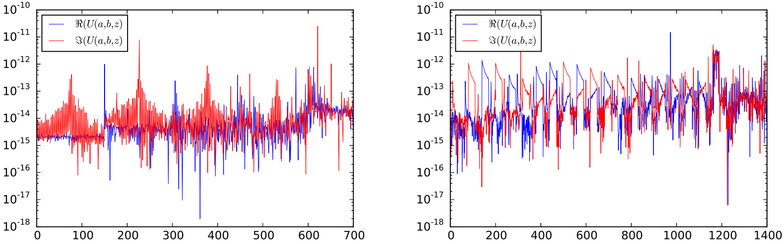

Figure: Relative error in computing $U(a, b, z)$. Error in $U(a, b, iz)$ for $a \in [2, 400]$, $b \in [-500, 500]$, $z \in [10^3, 10^6]$ (left) and $U(a, ib, z)$ for $a \in [10, 100]$, $b \in [10^3, 10^4]$, $z \in [10, 100]$ (right). 700 and 1400 tests respectively.

Concluding remarks

Summary

- Promising results, fast convergence as imaginary part increases
- Alternative to asymptotic expansions
- Outperforms available codes in double precision

Future work

- Tight error bound for Gauss-Laguerre quadrature
- Suitable integral representation for $|\Im(a)| \to \infty$
- Robust implementation

Possible further improvements

- Double exponential quadrature rules
- Use of conformal maps for the acceleration of double exponential integrals
- Talbot quadratures
- Complex $f(x)$ and $g(x)$

Danke schön!