ALTAY CONFERENCE PROCEEDINGS IN MATHEMATICS
2 (2025), No. 1, pp. 1-14
https://altayconfproceedings.com/
ISSN 3108 - 5288

Research Article

Computational aspects of approximating the Horn hypergeometric functions H_3 by branched continued fractions

MARTA DMYTRYSHYN*, SOFIIA HLADUN, MYKHAILO HOLOD, AND VOLODYMYR HLADUN

ABSTRACT. This paper investigates the approximation of Horn hypergeometric function H_3 using branched continued fractions. Based on the formal branched continued fraction expansion for the ratio of hypergeometric functions H_3, a branched continued fraction expansion for a specific function is constructed. Numerical experiments using a custom Python implementation compare the convergence properties of the branched continued fraction approximants with the partial sums of the corresponding double power series. Results, presented in tables and plots, demonstrate that the branched continued fraction approach generally offers better convergence properties, including potentially wider regions of convergence and higher accuracy, particularly in regions where the power series diverges or converges slowly. The convergence behavior is visualized through error plots in different complex planes, suggesting that the branched continued fraction provides a robust tool for approximating this special function. Additionally, algorithms for computing approximants of continued fractions are studied. The results show that the continuant method is unstable and slower than the backward recurrence algorithms. The backward recurrence algorithms are stable, and their parallel implementation is faster than the single-threaded version.

Keywords: Horn hypergeometric function H_3, branched continued fraction, convergence, approximation by rational functions, backward recurrence algorithm.

2020 Mathematics Subject Classification: 33C65, 30B99, 40A99, 41A20, 65D15.

1. INTRODUCTION

Current practices in applying modern technology and science in security and defense demand deep knowledge of applied mathematics, particularly special functions. These functions are used to construct approximate or exact analytical solutions to equations describing complex processes, such as those in physics, chemistry, and engineering, thereby providing a better and more meaningful understanding of the properties of these processes and mechanisms. Due to their importance, many works are dedicated to these functions, and even website was developed (https://functions.wolfram.com).

Branched Continued Fractions (BCFs) are a natural generalization of classical continued fractions [9, 10, 11, 13, 26, 27], inheriting many of their properties, including the most important approximation characteristics: a wide convergence region, good convergence rate [7, 12, 14, 16, 25], and numerical stability, which involves no accumulation or slow accumulation of errors when computing approximants [15, 20, 21, 22, 23]. Dmytro Bodnar and his students have demonstrated the effectiveness of approximating special functions, particularly hypergeometric functions, using BCFs [1, 2, 4, 5] and [18, 19, 28].

Received: 29.07.2025; Accepted: 25.09.2025; Published Online: 06.11.2025
*Corresponding author: Marta Dmytryshyn; m.dmytryshyn@wunu.edu.ua
DOI: 10.64700/altay.6
Presented in *International Workshop on Modern Problems of Analysis, Optimization, Approximation and Their Applications*

Despite significant achievements in approximating special functions, this topic remains one of the most important in the analytical theory of BCFs and still has many open problems, especially concerning hypergeometric functions.

One of the main tasks is investigating the convergence and numerical stability of BCF expansions for hypergeometric functions. Given the small number of theoretical studies regarding their convergence, applied research into these expansions, particularly for the Horn function H_3, is an interesting direction that should be explored to bring new knowledge to the study of special functions.

2. Branched continued fraction expansion

Let us consider Horn's hypergeometric function H_3 [24], defined as a double power series (DPS) of the form

$$H_3(\alpha,\beta;\gamma;\mathbf{z})=\sum_{r,s=0}^{+\infty}\frac{(\alpha)_{2r+s}(\beta)_s}{(\gamma)_{r+s}}\frac{z_1^r z_2^s}{r!s!}, \tag{2.1}$$

where $\alpha,\beta,\gamma\in\mathbb{C}$, $\gamma\notin\{0,-1,-2,\ldots\}$, $\mathbf{z}=(z_1,z_2)\in\mathbb{C}^2$, $(x)_k$ is the Pochhammer symbol, and let I be a set of multiindices

$$I=\{i(k)=(i_1,i_2,\ldots,i_k):\ 1\le i_r\le 2,\ 1\le r\le k,\ k\ge 1\}.$$

In [3], a formal BCF expansion for the ratio of functions (2.1) was obtained:

$$\frac{H_3(\alpha,\beta;\gamma;\mathbf{z})}{H_3(\alpha+1,\beta;\gamma+1;\mathbf{z})}=1+\sum_{i_1=1}^{2}\cfrac{c_{i(1)}(\mathbf{z})}{d_{i(1)}(\mathbf{z})+\sum_{i_2=1}^{2}\cfrac{c_{i(2)}(\mathbf{z})}{d_{i(2)}(\mathbf{z})+\ddots}}, \tag{2.2}$$

where the elements are determined by the formulas

$$c_1(\mathbf{z})=-\frac{(2\gamma-\alpha)(\alpha+1)z_1}{\gamma(\gamma+1)},\quad c_2(\mathbf{z})=-\frac{\beta(\gamma-\alpha)(1-4z_1)z_2}{\gamma(\gamma+1)}, \tag{2.3}$$

$$d_{i(k)}(\mathbf{z})=1-\frac{\alpha-\beta-1+\sum_{r=0}^{k-1}(\delta_{i_r}^1-\delta_{i_r}^2)}{\gamma+k}\delta_{i_k}^2 z_2-2\frac{2\gamma-\alpha+k+\sum_{r=0}^{k-1}\delta_{i_r}^2}{\gamma+k}\delta_{i_k}^2 z_1, \tag{2.4}$$

$$c_{i(k),1}(\mathbf{z})=-\frac{(2\gamma-\alpha+k+\sum_{r=0}^{k}\delta_{i_r}^2-2\delta_{i_k}^2(2\gamma-\alpha-\beta+k)z_2)(\alpha+\sum_{r=0}^{k}\delta_{i_r}^1)z_1}{(\gamma+k)(\gamma+k+1)}, \tag{2.5}$$

$$c_{i(k),2}(\mathbf{z})=-\frac{(\beta+\sum_{r=0}^{k}\delta_{i_r}^2)(\gamma-\alpha+\sum_{r=0}^{k}\delta_{i_r}^2)(1-4z_1)z_2}{(\gamma+k)(\gamma+k+1)} \tag{2.6}$$

for all $i(k)\in I$, where $i_0=1$, δ_i^j is the Kronecker delta. In [6], it was proved that if the parameters of the function (2.1) satisfy the inequalities $\gamma\ge\alpha\ge 0$, $\gamma\ge\beta\ge 0$, then BCF (2.2) converges to a finite value $f(\mathbf{z})$ for every $\mathbf{z}\in G$, where

$$G=\{\mathbf{z}\in\mathbb{C}^2:\ |z_1|\le h_1,\ |z_2|\le h_2\},$$

and h_1,h_2 are positive constants such that

$$8h_1(1+2h_2)(1-4h_1-h_2)+4(1+4h_1)h_2\le(1-4h_1-h_2)^2$$

and, in addition the convergent is uniformly on every compact subset of $\mathrm{Int}(G)$ to a function $f(\mathbf{z})$ holomorphic in $\mathrm{Int}(G)$. It was also proved that this BCF converges uniformly on every compact subset of the set

$$H = \bigcup_{\varphi\in(-\pi/2,\pi/2)} G_\varphi,$$

to a function $f(\mathbf{z})$ holomorphic in H, where

$$G_\varphi = \{\mathbf{z}\in\mathbb{C}^2:\ \mathrm{Re}(z_1e^{-i\varphi})<\lambda_1\cos\varphi,\ |\mathrm{Re}(z_2e^{-i\varphi})|<\lambda_2\cos\varphi,$$
$$|z_k|+\mathrm{Re}(z_ke^{-2i\varphi})<\nu_k\cos^2\varphi,\ k=1,2,\ |z_1z_2|-\mathrm{Re}(z_1z_2e^{-2\varphi})<\nu_3\cos^2\varphi\},$$

where λ_1, λ_2, ν_1, ν_2, ν_3, μ_1, μ_2 are positive constants such that

$$\frac{\nu_2+4\nu_3}{\mu_2}\le\min\left\{2(1-\mu_1)-2\frac{\nu_1}{\mu_1},\ 2(1-4\lambda_1-\lambda_2-\mu_2)-\frac{2\nu_1+4\nu_3}{\mu_1}\right\}.$$

Note that for $\varphi=0$, the set H is the convergence set established in [3, Theorem 2]. Since the sets $\mathrm{Int}(G)$ and H contain a neighborhood of the origin, considering the proof of part (B) of [3, Theorem 2], we conclude that the function $f(\mathbf{z})$ is the analytic continuation of the function on the left side of equality (2.2) into $\mathrm{Int}(G)$ and H. Also, note that in both cases, no estimate for the rate of convergence of the BCF (2.2) was established.

Using expansion (2.2), let's construct the BCF expansion for the function

$$H_3(1,1;3/2;-\mathbf{z}) = \frac{-1}{4\sqrt{z_2-z_1+z_2^2}}\ln\frac{1+2z_2-2\sqrt{z_2-z_1+z_2^2}}{1+2z_2+2\sqrt{z_2-z_1+z_2^2}}. \tag{2.7}$$

It is known [8] that

$$H_3(1,1;3/2;-\mathbf{z}) = \sum_{r,s=0}^{+\infty}(-1)^{r+s}\frac{(1)_{2r+s}(1)_s}{(3/2)_{r+s}}\frac{z_1^rz_2^s}{r!s!}. \tag{2.8}$$

Setting $\alpha=0$, replacing γ with $\gamma-1$, and considering that $H_3(0,\beta,\gamma-1;\mathbf{z})=1$, from (2.2) we obtain

$$H_3(1,\beta;\gamma;\mathbf{z}) = \cfrac{1}{1+\sum_{i_1=1}^{2}\cfrac{c_{i(1)}(\mathbf{z})}{d_{i(1)}(\mathbf{z})+\sum_{i_2=1}^{2}\cfrac{c_{i(2)}(\mathbf{z})}{d_{i(2)}(\mathbf{z})+\ddots}}}, \tag{2.9}$$

where the elements of the BCF are defined by (2.3)–(2.6), in which $\alpha=0$, and γ is replaced by $\gamma-1$. Next, setting $\beta=1,\gamma=3/2$ in (2.9), we obtain the BCF expansion for function (2.7):

$$H_3(1,1;3/2;-\mathbf{z}) = \cfrac{1}{1+\sum_{i_1=1}^{2}\cfrac{c_{i(1)}(\mathbf{z})}{d_{i(1)}(\mathbf{z})+\sum_{i_2=1}^{2}\cfrac{c_{i(2)}(\mathbf{z})}{d_{i(2)}(\mathbf{z})+\ddots}}}, \tag{2.10}$$

where

$$c_1(\mathbf{z}) = \frac{4z_1}{3}, \quad c_2(\mathbf{z}) = \frac{2(1+4z_1)z_2}{3}, \tag{2.11}$$

$$d_{i(k)}(\mathbf{z}) = 1 + 2\frac{-2+\sum_{r=0}^{k-1}(\delta_{i_r}^1 - \delta_{i_r}^2)}{2k+1}\delta_{i_k}^2 z_2 + 4\frac{k+1+\sum_{r=0}^{k-1}\delta_{i_r}^2}{2k+1}\delta_{i_k}^2 z_1, \tag{2.12}$$

$$c_{i(k),1}(\mathbf{z}) = \frac{4(1+k+\sum_{r=0}^{k}\delta_{i_r}^2 + 2\delta_{i_k}^2 k z_2)\sum_{r=0}^{k}\delta_{i_r}^1 z_1}{(2k+1)(2k+3)}, \tag{2.13}$$

$$c_{i(k),2}(\mathbf{z}) = \frac{2(1+\sum_{r=0}^{k}\delta_{i_r}^2)(1+2\sum_{r=0}^{k}\delta_{i_r}^2)(1+4z_1)z_2}{(2k+1)(2k+3)} \tag{2.14}$$

for all $i(k) \in I$.

Since the parameters of the hypergeometric function in (2.8) satisfy the conditions $\gamma \geq \alpha \geq 0, \gamma \geq \beta \geq 0$, from the above we conclude that the BCF (2.10) converges to a finite value $f(\mathbf{z})$ for every $\mathbf{z} \in G$, and, moreover, converges uniformly on every compact subset of $\mathrm{Int}(G)$ to the function $f(\mathbf{z})$ holomorphic in $\mathrm{Int}(G)$, and also that (2.10) converges uniformly on every compact subset of H. Furthermore, the function $f(\mathbf{z})$ is the analytic continuation of function (2.7) into $\mathrm{Int}(G)$ and H.

Note that the problem of choosing optimal values for h_1, h_2 for G and $\lambda_1, \lambda_2, \nu_1, \nu_2, \nu_3, \mu_1, \mu_2$ for H remains open.

3. Approximation of the special function (2.7)

From a computational perspective, approximating H_3 using BCFs involves several key aspects: developing efficient algorithms that can compute approximants to H_3 with minimal computational resources while maintaining high accuracy; studying the convergence behavior of the BCF approximation, including the region of convergence and the rate of convergence; ensuring numerical stability in the recursive computation of approximants, so that rounding errors are not accumulated — which is especially important for high-order approximations.

Let $n \in \mathbb{N}$ be a fixed natural number and $f_n(\mathbf{z})$ be an nth approximant of the expansion (2.9).

For a comparative analysis of the accuracy and convergence rate of the function approximation, software was created, the main purpose of which is to calculate the approximate values of functions of two variables using two methods:

- Branched continued fraction: Calculation of the nth approximants according to the BCF expansion;
- Double power series: Calculation of the nth partial sums of the DPS.

Technology stack:

- Programming language: Python;
- Development environment: PyCharm.

Main components and functionality:

(A) BCF Computation
- Classes implemented for calculating the partial numerators and denominators of the BCF.
- A recursive approach using the backward recurrence algorithm is used to calculate the value of the nth approximant of the BCF, which involves calculating the tails of

the nth approximant according to the formulas:

$$G_{i(k)}^{(n)}(\mathbf{z}) = d_{i(k)}(\mathbf{z}) + \sum_{i_{k+1}=1}^{2} \frac{c_{i(k+1)}(\mathbf{z})}{G_{i(k+1)}^{(n)}(\mathbf{z})}, \quad i(k) \in I, \ 0 \le k \le n-1, \quad G_{i(n)}^{(n)} = d_{i(n)}, \ i(n) \in I.$$

(B) Series Sum Computation
- Classes implemented for calculating the nth order partial sums of the DPS.
- Auxiliary methods implemented for calculating Pochhammer symbols, Kronecker deltas, and other related mathematical constructs.

(C) Analysis and Comparison

The software allows analysis and comparison of relative errors (Table 1) of the calculated function approximations at specified points $\mathbf{z}$ in the complex plane $\mathbb{C}^2$.

(D) Visualization

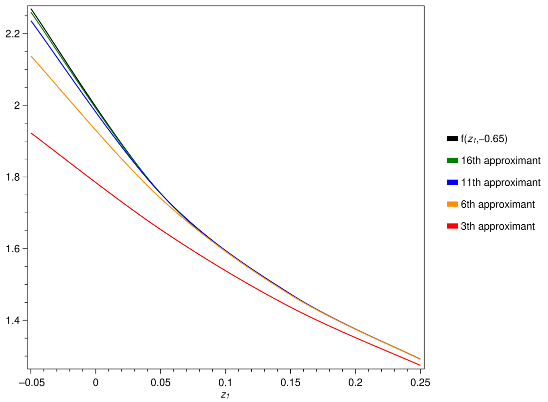
The software provides the ability to visualize results, compare approximation plots with the exact function values, and plot regions, where the BCF approximant and the power series partial sum achieve a specified approximation accuracy.

Let's analyze the approximation errors of function (2.7) using the BCF (2.10) and the DPS (2.8). Table 1 shows the relative errors of approximation at some points in $\mathbb{C}^2$ for function (2.7) by the 20th approximant $f_{20}(\mathbf{z})$ and the 20th partial sum $S_{20}(\mathbf{z})$. Analyzing the data in this table, we see that the BCF provides a better approximation than the DPS. Furthermore, there are points where the BCF converges, but the DPS diverges.

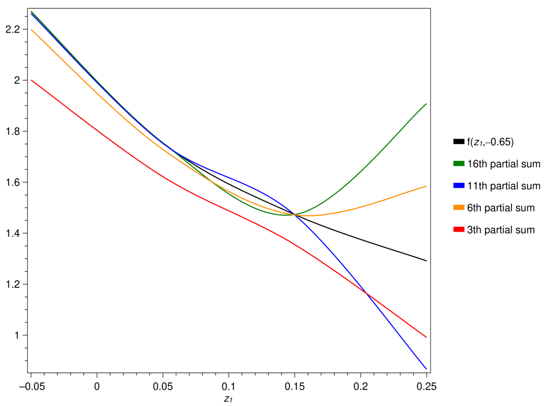
$\mathbf{z}$	**(2.10)**	**(2.8)**
(0.1250+0.0000i, 0.2500+0.0000i)	1.55×10^{-16}	5.07×10^{-09}
(0.1250+0.0000i, 0.3750+0.0000i)	6.88×10^{-14}	4.37×10^{-09}
(0.5000+0.0000i, 0.2500+0.0000i)	1.07×10^{-12}	$1.13\times10^{+04}$
(0.1083+0.0625i, 0.2165+0.1250i)	2.12×10^{-16}	5.39×10^{-09}
(0.3248+0.1875i, 0.4330+0.2500i)	1.37×10^{-13}	$2.60\times10^{+01}$
(0.4330+0.2500i, 0.6495+0.3750i)	5.61×10^{-12}	$6.77\times10^{+05}$
(0.0625+0.1083i, 0.1250+0.2165i)	3.14×10^{-16}	6.50×10^{-09}
(0.1875+0.3248i, 0.1250+0.2165i)	3.62×10^{-13}	$4.19\times10^{+01}$
(0.0000+0.1250i, 0.0000+0.2500i)	1.01×10^{-14}	9.05×10^{-09}
(0.0000+0.3750i, 0.0000+0.2500i)	9.20×10^{-12}	$6.01\times10^{+01}$
(-0.0625+0.1083i, -0.1250+0.2165i)	1.82×10^{-12}	1.48×10^{-08}
(-0.1875+0.3248i, -0.2500+0.4330i)	8.26×10^{-10}	$9.59\times10^{+01}$
(-0.0625-0.1083i, -0.1250-0.2165i)	1.82×10^{-12}	1.48×10^{-08}
(-0.1875-0.3248i, -0.2500-0.4330i)	8.26×10^{-10}	$9.59\times10^{+01}$
(0.0000-0.1250i, 0.0000-0.2500i)	1.01×10^{-14}	9.05×10^{-09}
(0.0000-0.3750i, 0.0000-0.2500i)	9.20×10^{-12}	$6.01\times10^{+01}$
(0.0625-0.1083i, 0.1250-0.2165i)	3.14×10^{-16}	6.50×10^{-09}
(0.1875-0.3248i, 0.1250-0.2165i)	3.62×10^{-13}	$4.19\times10^{+01}$
(1.0000+0.0000i, 2.0000+0.0000i)	9.73×10^{-09}	$3.71\times10^{+20}$

TABLE 1. Relative Errors of 20th Approximant and the 20th Partial Sum for (2.7)

Figure 1(A) and 1(B) show plots of the values of the BCF approximants (2.10), the partial sums of the DPS (2.8), and function (2.7) for $z_1 \in (-0.05, 0.25)$ and $z_2 = -0.65$. These figures also illustrate the advantages of approximating function (2.7) with the BCF compared to the DPS.



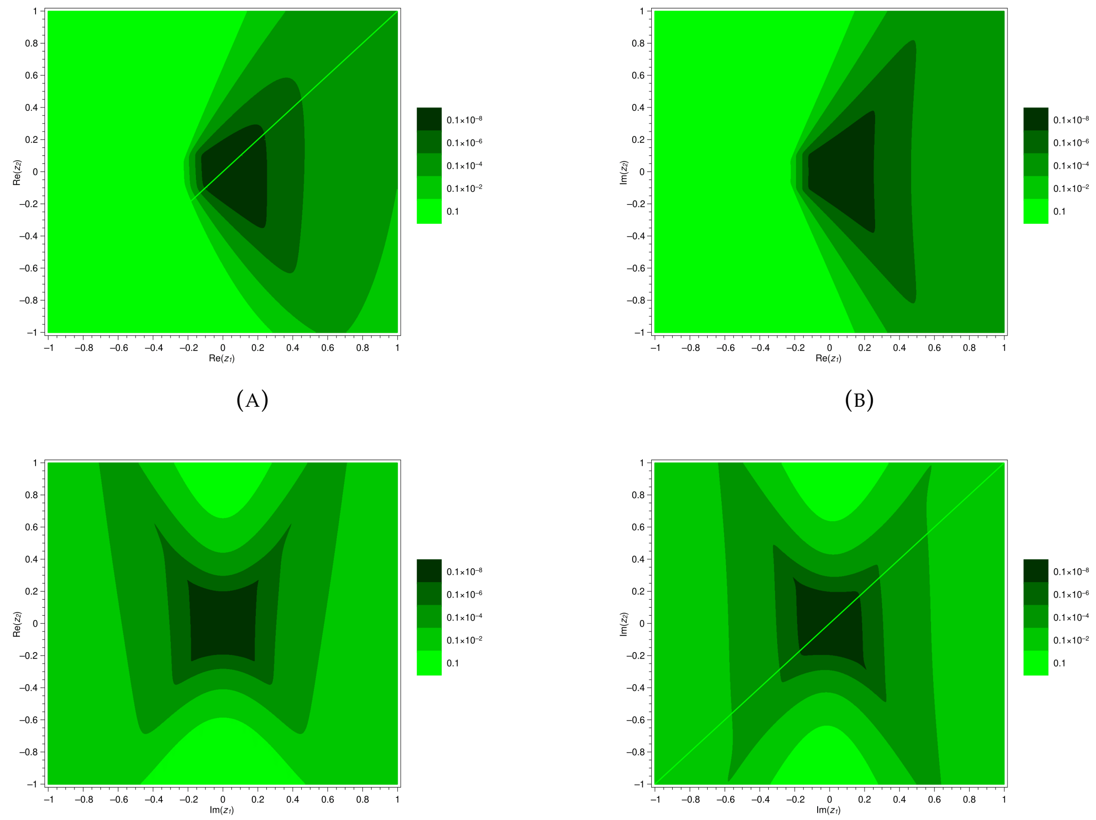
(A)



(B)

FIGURE 1. The plots of values of nth approximants of the BCF (A) and nth partial sums of the DPS (B) for function (2.7) with $z_2 = -0.65$.

Figure 2(A)–(D) show regions in different planes where the 11th approximant of the BCF (2.10) guarantees specified absolute error bounds for the approximation of function (2.7). Similarly, Figure 3(A)–(D) would depict regions where the 11th partial sum of the double power series (2.8) guarantees specified absolute error bounds for the approximation of function (2.7). As can be seen, the BCF (2.10) converges fastest in regions close to the origin, and the rate of convergence decreases as the distance from the origin increases. Considering the results of the applied research and the analytical expressions for the regions $\mathrm{Int}(G)$ and H, we conclude that there are wider convergence regions for the BCF (2.10), and consequently, wider regions of analytic continuation for the special function (2.7). Furthermore, the configurations of the regions in Figure 2(A)–(D) indicate the possibility of an analytical description of the convergence regions in different planes.



(C) (D)

FIGURE 2. Regions where the 11th approximant of BCF (2.10) guarantees specified absolute error bounds for the approximation of function (2.7) for (A) $\mathrm{Im}(z_1) = \mathrm{Im}(z_2) = 0$, (B) $\mathrm{Im}(z_1) = \mathrm{Re}(z_2) = 0$, (C) $\mathrm{Re}(z_1) = \mathrm{Im}(z_2) = 0$, (D) $\mathrm{Re}(z_1) = \mathrm{Re}(z_2) = 0$.

4. COMPARATIVE ANALYSIS OF THE EFFICIENCY OF ALGORITHMS FOR COMPUTING APPROXIMANTS OF BRANCHED CONTINUED FRACTIONS

This section examines and compares four approaches to computing the nth approximant of the branched continued fraction (2.10): the classical backward recurrence algorithm (BR algorithm), its parallel implementations on CPU and GPU, as well as the continuant method. Particular attention is devoted to the execution speed of the algorithms and their robustness against the accumulation of rounding errors.

We consider the approximant of the branched continued fraction as a tree-like structure of depth n, whose nodes are the coefficients $c_{i(k)}$ and $d_{i(k)}$. Each node of the tree is characterized by a multiindex `index` = $(i_1, i_2, \ldots, i_k)$, where $k-1$ denotes the depth level (the layer of the branched continued fraction) for coefficients $c_{i(k)}$ and k denotes the depth level for coefficients $d_{i(k)}$.

The classical BR algorithm computes the nth approximant of the branched continued fraction by performing a reverse traversal (bottom-up), that is, starting from the leaf coefficients

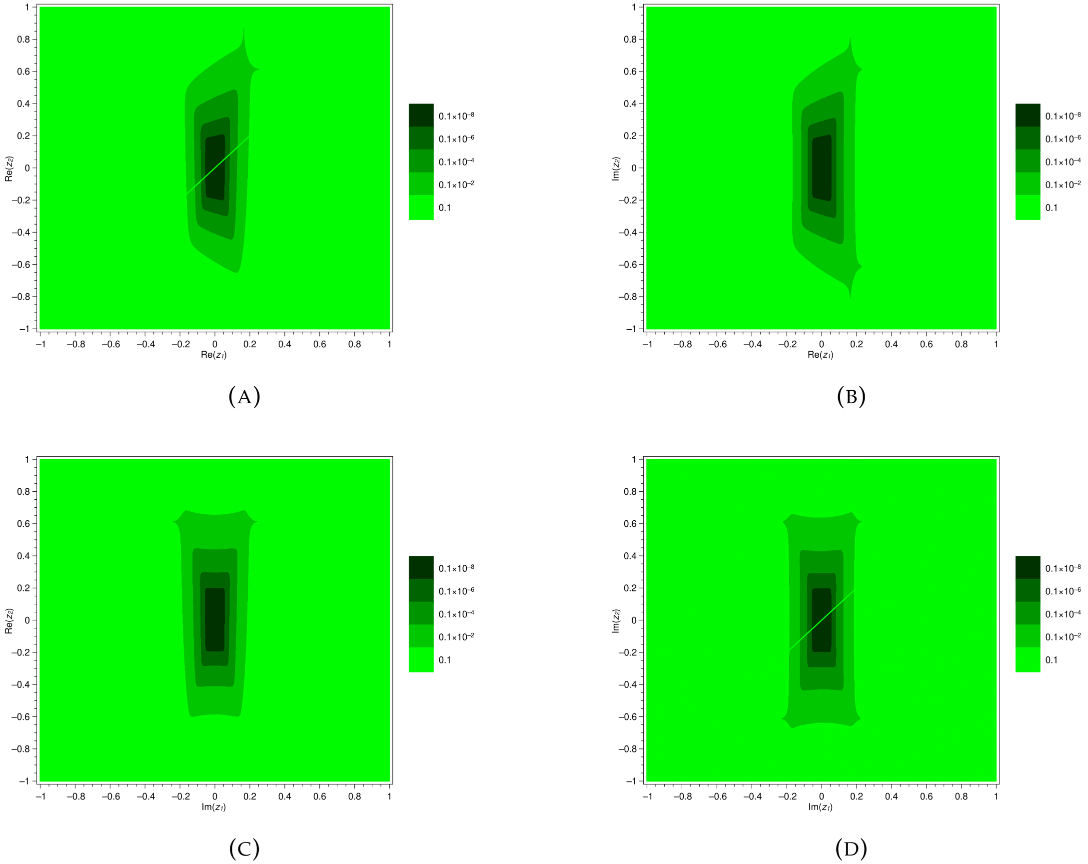


FIGURE 3. Regions where the 11th partial sum of power series (2.8) guarantees specified absolute error bounds for the approximation of function (2.7) for (A) $\mathrm{Im}(z_1) = \mathrm{Im}(z_2) = 0$, (B) $\mathrm{Im}(z_1) = \mathrm{Re}(z_2) = 0$, (C) $\mathrm{Re}(z_1) = \mathrm{Im}(z_2) = 0$, (D) $\mathrm{Re}(z_1) = \mathrm{Re}(z_2) = 0$.

and proceeding towards the root of the fraction. The algorithm is implemented as a recursive function that takes three arguments:

1. `depth` — the current recursion depth (with value 0 corresponding to the coefficients d_0, c_1 and c_2).
2. `index` — an array of length n representing a multi-index that uniquely identifies the path to a coefficient within the fraction structure.
3. `n` — the order of the approximant, corresponding to the maximum depth of computation.

To access the coefficients of the fraction, auxiliary functions `get_d(index, depth)` and `get_c(index, depth)` are employed, which return the values $d_{i(k)}$ and $c_{i(k)}$, respectively.

The pseudocode of the BR algorithm has the following form:

Algorithm 1 BR-algorithm

```
function BR_RECURSIVE(index, depth, n)
    v ← get_d(index, depth)                    ▷ Retrieve coefficient d at current depth
    if depth == n then
        return v                               ▷ Maximum depth reached
    end if
    for i = 1 to 2 do                          ▷ Sum contributions of child branches
        index[depth] ← i
        v ← v + get_c(index, depth)/br_recursive(index, depth + 1, n)
    end for
    return v
end function
```

This implementation allows one to compute the value of the branched continued fraction by invoking the function with arguments `depth = 0` and an initial array `initial_index = (0, 0, ..., 0)` of length n. Moreover, it makes it possible to compute any remainder of the approximant (a subfraction) by specifying the required multi-index and the depth of the first coefficient $d_{i(k)}$ of this subfraction.

Here we consider a parallel version of the BR algorithm in which multiple processes compute subfractions simultaneously on the CPU. At each depth level, these subfractions can be calculated independently, as they do not depend on one another until their results are summed. This allows the workload to be distributed across several processes, speeding up the execution of the algorithm.

The parallel BR algorithm can be formally described by the following pseudocode:

A parallel version of the BR algorithm has also been implemented using the Numba library, which enables the creation of CUDA [17] kernels for computations on the GPU. This implementation is similar to the CPU version; however, GPU threads are used instead of processes. Since Numba does not support recursive calls within CUDA kernels, the recursive function was replaced with an iterative implementation using a loop and a stack to simulate recursion.

The continuant method computes the value of the nth approximant of the branched continued fraction as the ratio of the determinants of two matrices, C_0 and C_1. The matrix C_0 is constructed such that its main diagonal contains the coefficients $d_{i(k)}$, the entries above the main diagonal are the coefficients $c_{i(k)}$, and symmetrically to them (below the main diagonal) are the values -1; all other elements of the matrix are zero. The matrix C_1 is obtained from C_0 by removing the first row and the first column.

The form of the matrix C_1 is presented below:

$$\begin{pmatrix} d_0 & c_1 & c_2 & 0 & 0 & 0 & 0 & \cdots \\ -1 & d_1 & 0 & c_{11} & c_{12} & 0 & 0 & \cdots \\ -1 & 0 & d_2 & 0 & 0 & c_{21} & c_{22} & \cdots \\ 0 & -1 & 0 & d_{11} & 0 & 0 & 0 & \cdots \\ 0 & -1 & 0 & 0 & d_{12} & 0 & 0 & \cdots \\ 0 & 0 & -1 & 0 & 0 & d_{21} & 0 & \cdots \\ 0 & 0 & -1 & 0 & 0 & 0 & d_{22} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

The continuant method can be formally described by the following pseudocode:

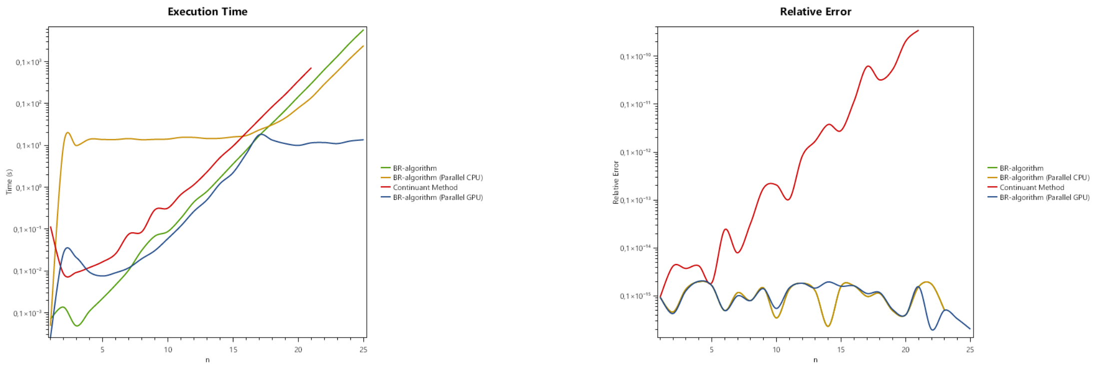
Algorithm 2 Parallel BR-algorithm

function RECURSIVE_PARALLEL(n, num_processes)
  if $n == 1$ **then**
    return BR_RECURSIVE(initial_index(n), 0, n)
  end if
  ▷ Determine the optimal depth for parallelization
  $parallel\_depth \leftarrow \lfloor \log_2(num\_processes) \rfloor$
  if $parallel\_depth \geq n$ **then**
    $parallel\_depth \leftarrow n - 1$
  end if
  ▷ Generate the set of all multi-indices of length $parallel\_depth$ in lexicographic order
  $indices\_set \leftarrow$ generate_indices(length=parallel_depth)
  ▷ 1. PARALLEL PHASE: compute independent subfractions
  $results \leftarrow \{\}$ ▷ Storage for results
  for all $idx \in indices\_set$ **in parallel do**
    $results[idx] \leftarrow$ BR_RECURSIVE$(idx, parallel_depth, n)$
  end for
  ▷ 2. SEQUENTIAL PHASE: compute the upper part of the fraction
  ▷ Modify `get_d` to return precomputed results at $parallel\_depth$
  function GET_D_MEMOIZED(index, depth)
    if $depth == parallel\_depth$ **then**
      return $results[index]$
    else
      return GET_D_ORIGINAL(index, depth)
    end if
  end function
  return BR_RECURSIVE(initial_index(n), 0, parallel_depth)
end function

Algorithm 3 Continuant Method

function CONTINUANT_METHOD(n)
  ▷ Construct the matrix C_0 according to the fraction structure
  $C_0 \leftarrow$ construct_matrix(n)
  ▷ Obtain the matrix C_1
  $C_1 \leftarrow C_0$ without the first row and column
  ▷ Perform LU decomposition for both matrices
  $L_0, U_0 \leftarrow$ LU_decomposition(C_0)
  $L_1, U_1 \leftarrow$ LU_decomposition(C_1)
  ▷ Extract diagonal elements of U
  $d_0 \leftarrow$ diagonal elements of U_0
  $d_1 \leftarrow$ diagonal elements of U_1
  ▷ Note: computing the determinants directly may cause variable overflow, so we use element-wise division of the diagonal elements instead
  ▷ Calculate the result as the product of diagonal elements ratios
  return $d_0[0] \cdot \prod_{i=1}^{\text{length}(d_0)-1} \frac{d_0[i]}{d_1[i-1]}$
end function

Figure 4 shows that the relative error of computing approximants using the continuant method increases with n, whereas the relative error for different implementations of the BR algorithm remains within machine epsilon. Furthermore, the continuant method requires a significant amount of memory, as the sizes of the matrices C_0 and C_1 grow exponentially with n. In this study, approximants were computed only up to $n = 21$, since for $n = 22$ the required memory would exceed 32 GB.



(A) (B)

FIGURE 4. Execution time and relative error of computing the branched continued fraction approximant (2.10).

For small values of n (up to 7), the classical BR algorithm implementation is the fastest. For $n > 7$, the GPU implementation of the BR algorithm becomes faster than the classical one. In all cases, the continuant method is slower than the classical BR algorithm and, for $n > 18$, is outperformed by the parallel BR algorithm on the CPU.

The BR algorithm executed in parallel on the CPU involves process creation, which requires additional time, making it slower than other implementations for $n \leq 18$ (see Fig. 5). However, for $n > 18$, the advantage of parallel execution outweighs the overhead of process creation, and this implementation becomes faster than both the classical BR algorithm and the continuant method.

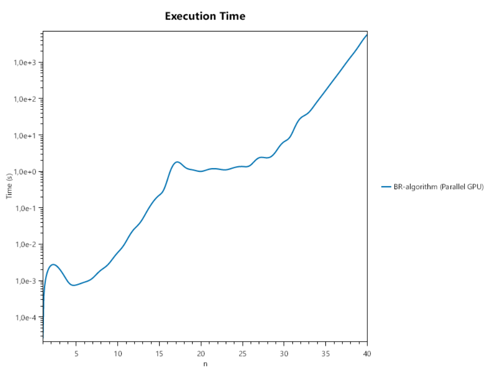


FIGURE 5. Execution time of the parallel GPU implementation of the BR algorithm.

As the value of n increases, the parallel GPU implementation of the BR algorithm engages more threads, but only at the last level of the fraction. This means that the workload per individual GPU thread does not increase; instead, the number of active threads and the load on the CPU grow. For $n > 16$, however, the additional workload falls solely on the GPU, while the CPU workload remains constant. Consequently, in the range $17 \leq n \leq 25$, the growth of execution time slows down: all GPU threads are fully utilized, and each thread handles a relatively small portion of the work. For $n > 25$, the workload per thread increases significantly again, leading to a faster increase in execution time.

5. Conclusion

In this work, using the expansion of the ratio of Horn's hypergeometric functions H_3 [5], an expansion for the special function (2.7) into BCF (2.10) was constructed. To construct and analyze the approximations (using nth approximants and nth partial sums) of this special function, specialized software was developed. The analysis of the applied research showed that the nth approximants converge fastest in regions close to the origin, and the rate of convergence decreases with increasing distance from the origin. Furthermore, the constructed BCF expansion has a wider convergence region and a better rate of convergence compared to the corresponding hypergeometric series.

Numerical experiments confirmed the feasibility and effectiveness of using BCFs as a tool for approximating special functions, particularly hypergeometric functions. Graphical illustrations indicate the potential for developing a new and promising direction in the analytical theory of continued fractions: the study of convergence regions of BCFs in various planes. The practical results obtained can also be used to investigate the convergence of BCF expansions of other special functions.

Additionally, different implementations of the BR algorithm and the continuant method were analyzed in terms of execution time and rounding errors. The BR algorithm was found to be stable, whereas the continuant method exhibited instability. Moreover, the continuant method is slower than the other approaches. The parallel implementation of the BR algorithm is faster than the single-threaded version, and its performance improves as more threads are used.

Author contributions. All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Financial disclosure. This research received no external funding.

Conflict of interest. The authors declare no potential conflict of interests.

References

[1] T. Antonova, C. Cesarano, R. Dmytryshyn and S. Sharyn: *An approximation to Appell's hypergeometric function F_2 by branched continued fraction*, Dolomites Res. Notes Approx., **17** (2024), 22–31.
[2] T. Antonova, R. Dmytryshyn and V. Goran: *On the analytic continuation of Lauricella-Saran hypergeometric function* $F_k(a_1, a_2, b_1, b_2; a_1, b_2, c_3; z)$, Mathematics, **11** (21) (2023), Article ID: 4487.
[3] T. Antonova, R. Dmytryshyn and V. Kravtsiv: *Branched continued fraction expansions of Horn's hypergeometric function H_3 ratios*, Mathematics, **9** (2) (2021), Article ID: 148.
[4] T. Antonova, R. Dmytryshyn, I.-A. Lutsiv and S. Sharyn: *On some branched continued fraction expansions for Horn's hypergeometric function $H_4(a, b; c, d; z_1, z_2)$ ratios*, Axioms, **12** (3) (2023), Article ID: 299.
[5] T. Antonova, R. Dmytryshyn and S. Sharyn: *Branched continued fraction representations of ratios of Horn's confluent function* H_6, Constr. Math. Anal., **6** (1) (2023), 22–37.
[6] T. M. Antonova: *On convergence of branched continued fraction expansions of Horn's hypergeometric function H_3 ratios*, Carpathian Math. Publ., **13** (3) (2021), 642–650.

[7] D. I. Bodnar, O. S. Manzii: *Expansion of the ratio of Appel hypergeometric functions H_3 into a branching continued fraction and its limit behavior*, J. Math. Sci., **107** (3) (2001), 3550–3554.
[8] Yu. A. Brychkov, N. V. Savischenko: *On some formulas for the Horn functions $H_3(a,b;c;w,z)$, $\mathrm{H}_6(a;c;w,z)$ and Humbert function $\Psi_1(a,b;c;w,z)$*, Integral Transforms Spec. Funct., **32** (9) (2021), 661–676.
[9] A. Cuyt, V. B. Petersen, B. Verdonk, H. Waadeland and W. B. Jones: *Handbook of Continued Fractions for Special Functions*. Springer, Dordrecht (2008).
[10] R. Dmytryshyn, V. Goran: *On the analytic extension of Lauricella–Saran's hypergeometric function F_K to symmetric domains*, Symmetry, **16** (2) (2024), Article ID: 220.
[11] R. Dmytryshyn, I.-A. Lutsiv and O. Bodnar: *On the domains of convergence of the branched continued fraction expansion of ratio $H_4(a,d+1;c,d;z)/H_4(a,d+2;c,d+1;z)$*, Res. Math., **31** (2023), 19–26.
[12] R. Dmytryshyn, I.-A. Lutsiv and M. Dmytryshyn: *On the analytic extension of the Horn's hypergeometric function H_4*, Carpathian Math. Publ., **16** (1) (2024), 32–39.
[13] R. Dmytryshyn, V. Oleksyn: *On Analytical Extension of Generalized Hypergeometric Function ${}_3F_2$*, Axioms, **13** (11) (2024). Article ID: 759.
[14] R. Dmytryshyn, T. Antonova and S. Hladun: *On analytical continuation of the Horn's hypergeometric functions H_3 and their ratios*, Axioms, **14** (1) (2025), Article ID: 67.
[15] R. Dmytryshyn, C. Cesarano, I.-A. Lutsiv and M. Dmytryshyn: *Numerical stability of the branched continued fraction expansion of Horn's hypergeometric function H_4*, Matem. Stud., **61** (1) (2024), 51–60.
[16] R. Dmytryshyn, S. Sharyn: *Representation of special functions by multidimensional A- and J-fractions with independent variables*, Fractal Fract., **9** (2) (2025), Article ID: 89.
[17] J. Sanders, E. Kandrot: *CUDA by Example: An Introduction to General-Purpose GPU Programming*, Addison-Wesley, Boston (2010).
[18] V. R. Hladun, N. P. Hoyenko, O. S. Manziy and L. Ventyk: *On convergence of function $F_4(1,2;2,2;z_1,z_2)$ expansion into a branched continued fraction*, Math. Model. Comput., **9** (3) (2022), 767–778.
[19] V. Hladun, R. Rusyn and M. Dmytryshyn: *On the analytic extension of three ratios of Horn's confluent hypergeometric function H_7*, Res. Math.,**32** (1) (2024), 60–70.
[20] V. R. Hladun: *Some sets of relative stability under perturbations of branched continued fractions with complex elements and a variable number of branches*, J. Math. Sci., **215** (1) (2016), 11–25.
[21] V. Hladun, D. Bodnar and R. Rusyn; *Convergence sets and relative stability to perturbations of a branched continued fraction with positive elements*, Carpathian Math. Publ., **16** (1) (2024), 16–31.
[22] V. R. Hladun, M. V. Dmytryshyn, V. V. Kravtsiv and R. S. Rusyn: *Numerical stability of the branched continued fraction expansions of the ratios of Horn's confluent hypergeometric functions* H_6, Math. Model. Comput., **11** (4) (2024), 1152–1166.
[23] V. Hladun, V. Kravtsiv, M. Dmytryshyn and R. Rusyn: *On numerical stability of continued fractions*, Mat. Stud., **62** (2) (2024), 168–183.
[24] J. Horn; *Hypergeometrische Funktionen zweier Veränderlichen*, Math. Ann., **105** (1931), 381–407.
[25] W. B. Jones, W. J. Thron: *Continued Fractions: Analytic Theory and Applications*, Addison-Wesley Pub. Co., Reading (1980).
[26] D. Kaliuzhnyi-Verbovetskyi, V. Pivovarchik: *Recovering the shape of a quantum caterpillar tree by two spectra*, Mech. Math. Methods, **5** (1) (2023), 14–24.
[27] A. A. Kaminsky, M. F. Selivanov: *On the application of branched operator continued fractions for a boundary problem of linear viscoelasticity*, Int. Appl. Mech., **42** (2) (2006), 115–126.
[28] O. Manziy, V. Hladun and L. Ventyk: *The algorithms of constructing the continued fractions for any rations of the hypergeometric Gaussian functions*, Math. Model. Comput., **4** (1) (2017), 48–58.

MARTA DMYTRYSHYN
WEST UKRAINIAN NATIONAL UNIVERSITY
DEPARTMENT OF MANAGEMENT AND ADMINISTRATION
11 LVIVSKA STR., 46009 TERNOPIL, UKRAINE
UNIVERSITY OF WISCONSIN-MADISON
INTERNATIONAL DIVISION
MADISON, WI 53706, UNITED STATES
Email address: m.dmytryshyn@wunu.edu.ua, dmytryshyn@wisc.edu

Sofiia Hladun
Lviv Polytechnic National University
Department of Applied Mathematics
12 Stepan Bandera Str., 79013, Lviv, Ukraine
Email address: sofiia.hladun.pm.2022@lpnu.ua

Mykhailo Holod
Lviv Polytechnic National University
Department of Applied Mathematics
12 Stepan Bandera Str., 79013, Lviv, Ukraine
Email address: mykhailo.holod.pm.2021@lpnu.ua

Volodymyr Hladun
Lviv Polytechnic National University
Department of Applied Mathematics
12 Stepan Bandera Str., 79013, Lviv, Ukraine
Email address: volodymyr.r.hladun@lpnu.ua