Python computations of general Heun functions from their integral series representations

T. Birkandan
Department of Physics, Istanbul Technical University, 34469 Istanbul, Turkey;
e-mail: birkandant@itu.edu.tr

P.-L. Giscard
Univ. Littoral Côte d'Opale, UR 2597, LMPA, Laboratoire de Mathématiques Pures et Appliquées Joseph Liouville, F-62100 Calais, France; e-mail: giscard@univ-littoral.fr

A. Tamar
Independent researcher, India; e-mail: adityatamar@gmail.com

We present a numerical implementation in Python of the recently developed unconditionally convergent representation of general Heun functions as integral series. We produce two codes available for download, one especially aimed at reproducing the output of Mathematica's HeunG function, the other for general-purpose computations. We show that the present code compares favorably with Mathematica's function, point to further improvements, and discuss the issue of singularities.

1 Introduction

Heun-type equations have found a large number of applications in physics, especially since the early 2000s [1, 2, 3]. In a recent publication [4], a novel formulation of all functions of the Heun class was given as unconditionally convergent integral series. While seemingly difficult to implement numerically owing to the large number of iterated integrals the method requires, we expose here significant simplifications allowing a fast and reliable mean to numerically evaluate the required integral series. The approach exposed here as well the underlying mathematical results [4] are valid for all functions of the Heun class: general, confluent, bi-confluent, doubly confluent and tri-confluent. For the sake of conciseness we present only the results pertaining to the general Heun function[1], namely,

$$\frac{d^2H_G(z)}{dz^2}+\left[\frac{\gamma}{z}+\frac{\delta}{z-1}+\frac{\epsilon}{z-t}\right]\frac{dH_G(z)}{dz} + \frac{\alpha\beta z-q}{z(z-1)(z-t)}H_G(z)=0, \tag{1}$$

where t is the location of the regular singular point other than $\{0,\,1,\,\infty\}$ [5, 6].

[1] By following a confluence procedure on its singularities [5] other equations of Heun class are obtained from (1); in particular confluence of the singularities at $z=t$ and $z=\infty$ yields the confluent Heun equation which is central to applications in black hole physics, see §4.

1.1 Mathematical background

The approach developed in [4] relies on the following observation: we can design a 2×2 matrix $\mathsf{M}(z)$ depending on a complex variable z such that the unique solution $\mathsf{U}(z,z_0)$ to the equation

$$\frac{d}{dz}\mathsf{U}(z,z_0)=\mathsf{M}(z)\mathsf{U}(z,z_0), \tag{2}$$

with initial condition $\mathsf{U}(z_0,z_0)=\mathsf{Id}$ the 2×2 identity matrix, comprises a Heun function. More precisely, setting

$$\mathsf{M}(z)=\begin{pmatrix}1 & 1\\ B_1(z)+B_2(z)-1 & B_1(z)-1\end{pmatrix}, \tag{3}$$

with

$$B_1(z):=-\frac{\gamma}{z}-\frac{\delta}{z-1}-\frac{\epsilon}{z-t},$$
$$B_2(z):=-\frac{\alpha\beta z-q}{z(z-1)(z-t)},$$

and

$$\psi(z_0):=\begin{pmatrix}H_0\\ H_0'-H_0\end{pmatrix}, \tag{4}$$

then $\psi(z):=\mathsf{U}(z,z_0).\psi(z_0)$ is the unique general Heun function $H_G(z)$ solution of the Cauchy problem given by equation (1) with boundary conditions $H_G(z_0)=H_0$ and $H_G'(z_0)=H_0'$. In itself this reformulation of the problem gains us nothing because matrix $\mathsf{M}(z)$ does not commute with itself at different z values: $\mathsf{M}(z').\mathsf{M}(z')\neq\mathsf{M}(z')\mathsf{M}(z)$ when $z\neq z'$. Because of this $\mathsf{U}(z,z_0)$, called the evolution operator of system (2), cannot easily be expressed in terms of $\mathsf{M}(z)$. Instead, $\mathsf{U}(z, z_0)$ is the ordered exponential of $\mathsf{M}(z)$, a generalisation of the notion of ordinary matrix exponential.

Evaluating ordered exponentials analytically is an old and difficult problem. Until recently it was only possible to use either one of two approaches: Floquet-based methods which only provide approximate solutions and require $\mathsf{M}(z)$ to be periodic as a function of z; or the Magnus expansion [7], a continuous analog of the Baker-Campbell-Hausdorff formula which involves an infinite series of increasingly intricate expressions and suffers from incurable divergence issues [8]. In 2015, [9] gave a graph-theoretic formulation of ordered exponentials allowing their exact expression in terms of continued fractions of finite depth and breadth taken with respect to a convolution like product, known as the Volterra composition.

While it is not necessary to review the graph theoretic arguments underlying this result, it is most important for our purpose to recall the theory of Volterra composition. To this end, consider the space D of distributions of the form

$$f(z', z) = \tilde{f}(z', z)\Theta(z' - z) + \sum_{i=1}^{\infty} \tilde{f}_i(z', z)\delta^{(i)}(z' - z),$$

where $\delta^{(i)}$ is the ith derivative of the Dirac delta distribution $\delta^{(0)}(z' - z) \equiv \delta(z' - z)$, $\Theta(z' - z)$ is the Heaviside step function with the convention $\Theta(0) = 1$ and all $\tilde{f}_i(z', z)$ are smooth functions in both variables. We define the following product, for $f, l \in D$,

$$(f * l)(z', z) = \int_{-\infty}^{\infty} f(z', \zeta)l(\zeta, z)d\zeta.$$

This product makes D into a non-commutative algebra [10, 11] with unit element $1_* \equiv \delta(z'-z)$. Now consider the ensemble $\mathrm{Sm}_\Theta \subsetneq D$ of distributions of D for which all smooth coefficients $\tilde{f}_i(z', z) = 0$, i.e. only the Heaviside part remains. On Sm_Θ the $*$-product simplifies to

$$\begin{aligned}(f * l)(z', z) &= \int_{-\infty}^{\infty} \tilde{f}(z', \zeta)\tilde{l}(\zeta, z)\Theta(z' - \zeta)\Theta(\zeta, z)\, d\zeta,\\ &= \int_{z}^{z'} \tilde{f}(z', \zeta)\tilde{l}(\zeta, z)d\zeta\, \Theta(z' - z), \end{aligned} \tag{5}$$

which is the convolution-like product introduced by Volterra in his studies of integral equations, now known as the Volterra composition [12]. Not only do distributions of Sm_Θ have $*$-inverses for all $z', z \in \mathbb{C}$ except at a countable number of isolated points [11] but, most importantly for our purposes here, they have $*$-resolvents. Such resolvents, denoted $(1_* - f)^{*-1}$ are given by the Neumann series,

$$\big(1_* - f\big)^{*-1}(z', z) = \sum_{n=0}^{\infty} f^{*n}(z', z),$$

where $f^{*0} = 1_*$ and $f^{*i} = f * f^{*(i-1)}$. Writing the $*$-products explictly yields the integral series

$$\begin{aligned}\big(1_* - f\big)^{*-1}(z', z) &= \delta(z' - z) + \tilde{f}(z', z)\Theta(z' - z)\\ &+ \int_{z}^{z'} \tilde{f}(z', \zeta)\tilde{f}(\zeta, z)d\zeta\Theta(z' - z)\\ &+ \int_{z}^{z'}\int_{\zeta_1}^{z'} f(z', \zeta_2)f(\zeta_2, \zeta_1)f(\zeta_1, z)d\zeta_2 d\zeta_1\, \Theta(z' - z)\\ &+ \dots\end{aligned}$$

For all $f \in \mathrm{Sm}_\Theta$ this series is unconditionally convergent everywhere on $z', z \in I^2$ except at the singularities of f [9]. The $*$-resolvent of f solves

$$f * \big(1_* - f\big)^{*-1} = \big(1_* - f\big)^{*-1} - 1_*,$$

which means that it solves the linear Volterra integral equation of the second kind with kernel $f(z', z) = \tilde{f}(z', z)\Theta(z' - z)$. The method of path-sum gives any entry of any ordered exponential exactly as continued fractions of $*$-resolvents, each of which is representable by its unconditionally convergent Neumann series.

For the matrix M of Eq. (3) and initial vector of (4), the path-sum continued fraction yields

$$\begin{aligned}H_G(z) = H_0 + H_0\int_{z_0}^{z} G_1(\zeta, z_0)d\zeta\ + \\ (H_0' - H_0)\left(e^{z-z_0} - 1 + \int_{z_0}^{z}(e^{z-\zeta} - 1)G_2(\zeta, z_0)d\zeta\right),\end{aligned} \tag{6}$$

where the G_i are defined from $*$-resolvents, $G_i = (1_* - K_i)^{*-1} - 1_* = \sum_{n=1}^{\infty} K_i^{*n}$ with kernels

$$\begin{aligned}K_1(z, z_0) &= 1 + \frac{e^{-z}}{z^\gamma(z-1)^\delta(t-z)^\epsilon}\times\\ &\quad\int_{z_0}^{z} \zeta_1^\gamma(\zeta_1 - 1)^\delta(t - \zeta_1)^\epsilon e^{\zeta_1}X(\zeta_1)d\zeta_1,\\ K_2(z, z_0) &= X(z)e^{z-z_0} - \frac{q - \alpha\beta z}{(z-1)z(z-t)}.\end{aligned}$$

where

$$X(z) := \frac{q - \alpha\beta z}{(z-1)z(z-t)} - \frac{\epsilon}{t-z} - \frac{\gamma}{z} - \frac{\delta}{z-1} - 1.$$

See [4] for the proof of this result.

2 Numerical implementation

2.1 *Integral series on discrete times*

Let $I =]a, b[$ be the interval of interest over which the Heun function is sought, $z_0 \in \bar{I}$ (where $\bar{I}$ designates the closure of I) be the point at which the initial values are known and let $\{z_i \in I\}_{0\leq i\leq N-1}$ be the discrete values at which the Heun function is to be numerically evaluated. For simplicity, suppose that the distance $|z_{i+1} - z_i| = \Delta z$ is the same for all $0 \leq i \leq N-2$. This assumption is not necessary but will alleviate the notation. Now for a smooth function $\tilde{f}(z', z)$ over I^2, we define a matrix F with entries

$$\mathsf{F}_{i,j} := f(z_i, z_j) = \tilde{f}(z_i, z_j)\Theta(z_i - z_j).$$

Note that, by construction, F is lower *triangular* owing to the Heaviside step function. Of major importance is the observation that once I is discretized, the Volterra composition of Eq. (5) turns into an ordinary matrix product

$$(f * l)(z_i, z_j) = \int_{z_j}^{z_i} \tilde{f}(z_i, \zeta)\tilde{l}(\zeta, z_j)d\zeta\ \Theta(z' - z)$$

$$\downarrow$$

$$\sum_{z_j \leq z_k \leq z_i} \tilde{f}(z_i, z_k)\tilde{l}(z_k, z_j)\, dz = (\mathsf{F}.\mathsf{L})_{i,j}\, \Delta z.$$

Here $\mathsf{L}_{k,j} := l(z_k, z_j)\Theta(z_k - z_j)$. Rigorously,

$$\lim_{\Delta z \to 0} (\mathsf{F}.\mathsf{L})_{i,j}\, \Delta z = \int_{z_j}^{z_i} f(z_i, \zeta)l(\zeta, z_j)dz,$$

This matricial presentation is useful for direct integration as well since for example

$$\int_{z_i}^{z_j} f(\zeta, z_i)dz\, \Theta(z_j - z_i) = \lim_{dz \to 0} (\mathsf{H}.\mathsf{F})_{i,j}\, \Delta z, \quad (7)$$

where $\mathsf{H}_{i,j} := 1$ if $i \geq j$ and 0 otherwise. Most importantly, this line of results extends to $*$-resolvents,

$$\big(1_* - f\big)^{*-1}(z_i, z_j) = \lim_{\Delta z \to 0} \frac{1}{\Delta z}\big(\mathsf{Id} - \Delta z\, \mathsf{F}\big)^{-1}_{i,j},$$

so that, in practical numerical computations with $\Delta z \ll 1$, we may use

$$(1_* - f)^{*-1}(z_i, z_j) \simeq \frac{1}{\Delta z}\big(\mathsf{Id} - \Delta z\, \mathsf{F}\big)^{-1}_{i,j},$$

and

$$(1_* - f)^{*-1} - 1_* \simeq \big(\mathsf{Id} - \Delta z\, \mathsf{F}\big)^{-1}_{i,j} - \mathsf{Id}_{i,j}/\Delta z.$$

We improve on the above by noting that these results correspond to using the rectangular rule of integration as revealed by Eq. (7). Using the trapezoidal rule instead leads to much more accurate results. To follow this rule, the usual matrix product $\mathsf{F}.\mathsf{L}\,\Delta z$ representing $f * l$ must be replaced by

$$\frac{\Delta z}{2}(\mathsf{F} - \mathsf{dF}).\mathsf{L} + \frac{\Delta z}{2}\mathsf{F}(\mathsf{L} - \mathsf{dL}),$$

where e.g. dF is the diagonal of F. Now the $*$-resolvent of f becomes

$$(1_* - f)^{*-1}(z_i, z_j) \simeq \frac{1}{\Delta z}\left(\mathsf{Id} - \Delta z\mathsf{F} + \frac{\Delta z}{2}\mathsf{dF}\right)^{-1}_{i,j}.$$

Taken together these results completely bypass the iterated integral series rather than evaluating directly what it converges to. The entire complexity inherent in the integral series representation of the general Heun function is reduced to multiplying and inverting *triangular*, well-conditioned matrices. For the latter assertion, consider the diagonal elements of $\mathsf{Id} - \Delta z\mathsf{F} + (\Delta z/2)\mathsf{dF}$: these can be made to be different from 0 by tuning Δz.

2.2 *Numerical implementation*

Concretely, a code implementing the path-sum formulation of the evolution operator $\mathsf{U}(z, z_0)$ and, from there, the general Heun function, must evaluate kernels K_1 and K_2, construct the corresponding matrices, compute their ordinary matricial resolvents, add and multiply them as required.

Let $\mathsf{K}_{1,2}$ be the triangular matrix representing kernel K_1 or K_2, i.e. $(\mathsf{K}_{1,2})_{i,j} = K_{1,2}(z_i, z_j)\Theta(z_i - z_j)$. Then, construct the matrix

$$\mathsf{R} := H_0\left(\mathsf{H} + \frac{\Delta z}{2}\mathsf{g}_1.(\mathsf{H} - \mathsf{dH}) + \frac{\Delta z}{2}(\mathsf{g}_1 - \mathsf{dg}_1).\mathsf{H}\right) + (H_0' - H_0)\left(\mathsf{E} + \frac{\Delta z}{2}(\mathsf{E} - \mathsf{dE}).\mathsf{G}_2\frac{\Delta z}{2} + \mathsf{E}.(\mathsf{G}_2 - \mathsf{dG}_2)\right),$$

where $\mathsf{g}_1 := \mathsf{Id} + \Delta z\mathsf{G}_1$, $\mathsf{E}_{i,j} := (e^{z_i - z_j} - 1)\Theta(z_i - z_j)$ and $\mathsf{G}_{1,2} := \Delta z^{-1}(\mathsf{Id} - \Delta z\mathsf{K}_{1,2} + (\Delta z/2)\mathsf{dK}_{1,2})^{-1} - \mathsf{Id}/\Delta z$. Comparing the above with Eq. (6) indicates that

$$\lim_{\Delta z \to 0} \mathsf{R}_{i,j} = \mathsf{U}(z_i, z_j).\psi(z_0),$$

so that in particular

$$\lim_{\Delta z \to 0} \mathsf{R}_{i,0} = H_G(z_i).$$

This shows that only the first column of R is useful. This observation is profitably exploited numerically, as it avoids the need to even compute matrix inverses $(\mathsf{Id} - \Delta z \mathsf{K}_{1,2} + (\Delta z/2)\mathsf{dK}_{1,2})^{-1}$, rather asking for the solution of the triangular system $(\mathsf{Id} - \Delta z \mathsf{K}_{1,2} + (\Delta z/2)\mathsf{dK}_{1,2}).\vec{x} = \vec{v}$, where $v = (1, 0, 0...)^{\mathrm{T}}$, which is faster and requires less memory. From there further matrix multiplications can be implemented vectorially on $\vec{x}$.

The last remaining difficulty lies in constructing matrix K_1 as $K_1(z_i, z_j)$ involves an integral

$$\mathfrak{I}(z_i, z_j) := \int_{z_j}^{z_i} \zeta_1^{\gamma}(\zeta_1 - 1)^{\delta}(t - \zeta_1)^{\epsilon} e^{\zeta_1} X(\zeta_1) d\zeta_1.$$

Consequently, computing all entries $(\mathsf{K}_1)_{i,j}$ naively requires a quadratically growing number of integrals $\mathfrak{I}(z_i, z_j)$ to be evaluated. This difficulty disappears thanks to the integral linearity: first we compute only integrals $\mathfrak{I}(z_{j+1}, z_j)$ over infinitesimal intervals $[z_i, z_{i+1}]$, which is done e.g. with the trapezoidal rule. Then the first column of K_1 is

$$(\mathsf{K}_1)_{i,0} = 1 + \frac{e^{-z_i}}{z_i^{\gamma}(z_i - 1)^{\delta}(t - z_i)^{\epsilon}} \mathfrak{I}(z_i, z_0),$$

the required integrals of which we obtain from cumulative sums over the integrals on infinitesimal intervals since $\mathfrak{I}(z_i, z_0) = \sum_{j=0}^{i-1} \mathfrak{I}(z_{j+1}, z_j)$. The integrals $\mathfrak{I}(z_i, z_j)$ with $j > 0$ which are required to construct the subsequent columns of K_1 are similarly given by $\mathfrak{I}(z_i, z_j) = \mathfrak{I}(z_i, z_0) - \mathfrak{I}(z_j, z_0)$, which can be done at once for all $j \geq 0$ with a meshgrid function.

At this point the numerical code relies solely on well-conditioned triangular matrices, need not inverting any of them explicitly as it solves two matrix-vector linear equations instead, computes no integral explicitly but relies on the trapezoidal rule throughout, is guaranteed to be everywhere convergent except at singularities and is exact under the limit $\Delta z \to 0$.

2.3 *Optional improvement*

Seeking more flexibility in the calculations, the code allows one to subdivide the interval of interest I into N_1 smaller intervals. Over each of the subintervals, $H_G(z)$ is evaluated from its path-sum formulation as presented in the preceding sections with N_2 points. The value of $H_G(z)$ at the border of an interval is used as Cauchy initial value for the computations over the next interval. Overall, a total of $N = N_1 N_2$ points are computed. Tuning N_1 and N_2 independently allows the user to trade accuracy for large N values and vice-versa. For example, picking $N_1 \gg 1$ while keeping N_2 moderate allows for much faster evaluation than pure path-sum should the user require a very large number $N \gg 1$ of values, or to evaluate H_G over a very large interval. At the opposite, more accurate results will be obtained by making N_2 large while N_1 can be as low as 1.

3 Performance

3.1 *Reproducing HeunG*

Mathematica's HeunG function is a new feature in version 12.1 of the software. It produces the regular solution of the general Heun equation with boundary conditions $H_G(0) = 1$ and $H'_G(0) = q/(\gamma\, t)$. HeunG is capable of both analytical and numerical computations and we propose here only to compare its numerical performance with that of the integral-series based Python code available on GitHub [13].

We chose the following parameters for the Heun function : $t = 9/2$, called 'a' in Mathematica's HeunG, $q = -1$, $\alpha = 1$, $\beta = -3/2$, $\gamma = -14/100$, $\delta = 432/100$ and $\epsilon = 1.0 + \alpha + \beta - \gamma - \delta$. We want to represent $H_G(z)$ for $z \in I = [-2.2, 0.8]$. Observe how this interval comprises a singularity at $z = 0$. Problems with this singularity are avoided upon specifying ordinary conditions on the left $z < 0$ and right $z > 0$ of it at $|z_0| \ll 1$. Values for $H_G(z_0)$ and $H'_G(z_0)$ are known within machine precision from the first three orders of the general series expansion of $H_G(z)$ near $z = 0$. For the more general issue of crossing over singularities, see the discussion in §3.3 below.

Mathematica was tasked with producing a table of values for $H_G(z_j)$ with $z_j = -2.2 + j \times 3/N$, where N is the number of points, passing the parameters as finite precision reals, which allowed it to proceed faster. The Python code ('num_heunG.py' available for download) was given the same task, fixing $N_2 = 100$ and increasing N_1 so as to obtain the desired number $N = N_1 N_2$ of points. This provides an accuracy of 10^{-6} for all points.

Code	N	Time (sec.)
HeunG	1000	2.22
Python	1000	0.0096
HeunG	10000	22.4
Python	10000	0.090
HeunG	50000	110.8
Python	50000	0.43
HeunG	100000	231
Python	100000	0.90
HeunG	200000	464
Python	200000	1.87

The Python code is written using Python 3.8. We employed the 'time' module in Python and the command 'Timing[...]' in MATHEMATICA to measure the run time. In this analysis, we used a Dell laptop running Ubuntu 18.04, equipped with Intel Core i7-8665U CPU @ 1.90GHz $\times 8$ and 15,5 GiB memory, using a single core and serial computations.

3.2 General Heun functions with arbitrary boundary conditions

The Python implementation of the integral series formulation of Heun function is natively well suited to evaluating the general Cauchy problem of determining the solution $H_G(z)$ of Eq. (1) with arbitrary Cauchy conditions at z_0 on an interval I that does not cross over a singularity. The Python code ('num_heun.py' available for download) accepts the parameters of the Heun function, the interval I and the values of N_1 and N_2 as inputs.

3.3 Crossing singularities and exploring the complex plane

The difficulty encountered by the present code in crossing over singularities is of a different nature from that encountered by series-based methods. Indeed, here the underlying integral representation of the Heun function is valid and guaranteed to converge everywhere in the complex plane $z \in \mathbb{C}$ except at the singularities themselves. In particular there is no need for analytic continuation. Rather, the difficulty lies solely in that ordinary line integrals fail to be defined at the singularities and cannot pass through them. Instead, since the integral representation exists and is the same everywhere else on the complex place, we may integrate along any smooth curve γ starting at z_0, ending at the desired z_1 and avoiding all singularities in between, while choosing principal values or otherwise for any residue that may arise. Since the punctured plane is connected this will recover values for $H_G(z_1)$ and $H_G'(z_1)$ irrespectively of whether or not singularities lie on the line joining z_0 and z_1. From this point onwards, both the mathematics and the code presented here work the same replacing

$$\int_{z_0}^{z_1} d\zeta \mapsto \int_{\substack{\gamma\subset\mathbb{C}\setminus\mathcal{S} \\ \gamma: z_0\to z_1}} d\gamma$$

where $\mathcal{S}$ denotes the set of singularities of H_G. More generally, this strategy yields the desired representation for $H_G(z)$ over the whole complex plane with no further alteration of the code. The concrete Python implementation of this extension, while it does not seem to present fundamental difficulty, has yet to be implemented.

4 Prospective applications to gravitational physics

The representation of Heun functions as uniformly convergent integral series as well as the computability of the latter as demonstrated here should help tackle mathematical bottlenecks arising in numerical solutions of the Teukolsky equation [14]. This equation describes metric perturbations of Kerr spacetime. In order to calculate quantities of physical interest steming from this equation such as quasinormal modes (QNMs) there are currently two alternatives: rely on Leaver's method [15], which uses a Coulomb wave functions series expansion; or utilise the Mano-Suzuki-Takesugi (MST) method [16], which wields two different hypergeometric series valid near the black hole horizon and at spatial infinity, then match both using matched asymptotic expansions and an auxiliary parameter absent from the Teukolsky equation. Despite its ability to calculate further astrophysical quantities [17], the mathematical complexity of the MST approach has been identified as a major hindrance in understanding the underlying physical picture [17].

With a uniformly convergent and numerically efficient method of solving the Heun equations, in particular the confluent Heun equations equivalent to the Teukolsky radial and angular equations, it is possible in principle to solve the coupled system to obtain QNM values [18].

By the *very same construction* and without using any other special functions, we may compute further quantities such as gravitational fluxes due to orbiting bodies [19] and scattering cross sections, both of which require uniformly convergent solutions to the Teukolsky equation. In fact, the lack hitherto of such convergence has proved to be a formidable obstacle for astrophysical applications [20]. We also remark that the uniformly convergent integral series representation of Heun function does not use the Sasaki-Nakamura transform [21]. Thus, we expect that the mathematical challenges arising from the long range nature of the Teukolsky potential recede whenever the integral series formulation is used.

Acknowledgements

P.-L. G. is supported by the Agence Nationale de la Recherche young researcher grant No. ANR-19-CE40-0006.

References

[1] M. Hortaçsu. Heun functions and some of their applications in physics. Advances in High Energy Physics, 2018:8621573, 2018.

[2] M. Hortaçsu and T. Birkandan. Quantum field theory applications of Heun type functions. Rep. Math. Phys., 79(1):81–87, 2017.

[3] A.M. Ishkhanyan. Schrödinger potentials solvable in terms of the general Heun functions. Annals of Physics, 388:456–471, 2018.

[4] P.-L. Giscard and A. Tamar. Elementary integral series for heun functions. with an application to black-hole perturbation theory, 2020.

[5] S.Y. Slavyanov and W. Lay. Special Functions: A Unified Theory Based on Singularities. Oxford University Press, New York, 2000.

[6] A. Ronveaux, F. M. Arscott, S. Y. Slavyanov, Schmidt D., Wolf G., Maroni P., and Duval A. Heun's Differential Equations. Oxford University Press, 1995.

[7] W. Magnus. On the exponential solution of differential equations for a linear operator. Commun. Pure Appl. Math., 7(4):649–673, 1954.

[8] S. Blanes, F. Casas, J.A. Oteo, and J. Ros. The magnus expansion and some of its applications. Phys. Rep., 470(5):151 – 238, 2009.

[9] P.-L. Giscard, K. Lui, S. J. Thwaite, and D. Jaksch. An exact formulation of the time-ordered exponential using path-sums. J. Math. Phys., 56(5):053503, 2015.

[10] P.-L. Giscard and S. Pozza. Lanczos-Like Algorithm for the Time-Ordered Exponential: The $*$-Inverse Problem. Appl. Math., 65:807–827, 2020.

[11] P.-L. Giscard and S. Pozza. Tridiagonalization of systems of coupled linear differential equations with variable coefficients by a Lanczos-like method. Linear Algebra Its Appl., 624:153–173, 2020.

[12] V. Volterra and J. Pérès. Leçons sur la composition et les fonctions permutables. Éditions Jacques Gabay, 1924.

[13] T. Birkandan, P.-L. Giscard, and A. Tamar. `https://github.com/tbirkandan/integralseries_heun`, 2021.

[14] S.A. Teukolsky. Perturbations of a rotating black hole. i. fundamental equations for gravitational, electromagnetic, and neutrino-field perturbations. Astroph. J., 185:635–648, 1973.

[15] E. Leaver. An analytic representation for the quasi-normal modes of Kerr black holes. Proc. R. Soc. Lond. A, 402:285–298, 1985.

[16] S. Mano, H. Suzuki, and E. Takasugi. Analytical solutions of the teukolsky equation and their low frequency expansions. Prog. Theor. Phys., 95:1079–1096, 1996.

[17] M. Sasaki and H. Tagoshi. Analytic black hole perturbation approach to gravitational radiation. Living Rev. Relativ., 6:6, 2003.

[18] Y. Hatsuda. Quasinormal modes of Kerr-de Sitter black holes via the Heun function. Class. Quant. Grav., 38(2):025015, 2020.

[19] R. Fujita and H. Tagoshi. New numerical methods to evaluate homogeneous solutions of the teukolsky equation. Prog. Theor. Phys., 112:1079–1096, 2004.

[20] T. Stratton, L. Leite, S. Dolan, and L. Crispino. Series reduction method for scattering of planar waves by Kerr black holes. Phys. Rev. D, 102(4):044025, 2020.

[21] M. Sasaki and T. Nakamura. Gravitational Radiation From a Kerr Black Hole. 1. Formulation and a Method for Numerical Analysis. Prog. Theor. Phys., 67:1788, 1982.