Fast elliptic solvers in cylindrical coordinates and the Coulomb collision operator

Andras Pataki and Leslie Greengard
Courant Institute, New York University, 251 Mercer Street, NY, NY 10012

Abstract
In this paper, we describe a new class of fast solvers for separable elliptic partial differential equations in cylindrical coordinates \((r, \theta, z)\) with free-space radiation conditions. By combining integral equation methods in the radial variable \(r\) with Fourier methods in \(\theta\) and \(z\), we show that high-order accuracy can be achieved in both the governing potential and its derivatives. A weak singularity arises in the Fourier transform with respect to \(z\) that is handled with special purpose quadratures. We show how these solvers can be applied to the evaluation of the Coulomb collision operator in kinetic models of ionized gases.

Keywords: collision operator, plasma physics, Poisson equation, biharmonic equation, fast solvers

PACS: 51.10.+y, 52.65.Ff, 02.60.Lj, 41.20.Cv

1. Introduction
A variety of problems in computational physics require the solution of the Poisson and biharmonic equations in cylindrical coordinate systems, particularly when the source distribution (the right-hand side) is axisymmetric or involves only a few azimuthal modes. The present paper was motivated by the need to compute the Coulomb collision operator \(C(f_a, f_b)\) in kinetic simulations of the Boltzmann-Fokker-Planck equation \[4, 5, 31, 32, 33\]:

\[
\frac{\partial}{\partial t} f_a + v \cdot \nabla f_a + \frac{\mathbf{e}_a}{m_a} (\mathbf{E} + v \times \mathbf{B}) \cdot \partial_v f_a = \sum_b C(f_a, f_b). \tag{1}
\]

Here, \(f_a(x, v, t)\) denotes the state of an ionized gas for plasma species \(a\) and the index \(b\) runs over all species present. In the Fokker-Planck-Landau formalism \[26\],

\[
C(f_a, f_b) = \gamma_{ab} \partial_v \cdot \int \mathcal{S}(v - v') \left( \frac{\partial_v f_a(v)}{m_a} f_b(v') - f_a(v) \frac{\partial_v f_b(v')}{m_b} \right) dv'. \tag{2}
\]

where

\[
\mathcal{S}(v - v')_{ij} = \delta_{ij} \frac{1}{|v - v'|} - \frac{(v_i - v'_i)(v_j - v'_j)}{|v - v'|^3}. \tag{3}
\]

An alternative representation makes use of the Rosenbluth potentials \[32\]:
\[ C(f_a, f_b) = \frac{\gamma_{ab}}{m_a} \partial_v \left[ \partial_v \cdot (f_a \partial_v G_b) - 2 \left( 1 + \frac{m_a}{m_b} \right) f_a \partial_v H_b \right] \]  \tag{4}

where

\[ H_b(v) = \int \frac{1}{|v - v'|} f_b(v') \, dv' \quad \text{or} \quad \Delta H_b = -4\pi f_b \]  \tag{5}

and

\[ G_b(v) = \int |v - v'| f_b(v') \, dv' \quad \text{or} \quad \Delta^2 G_b = -8\pi f_b \]  \tag{6}

Note that four derivatives of \( G_b \) are required in (4), while \( G_b \) itself satisfies the inhomogeneous biharmonic equation (6). Thus, direct discretization of the partial differential equation, followed by evaluation of the collision operator via (4) would require eight steps of numerical differentiation, with significant loss of accuracy.

It is natural, therefore, to consider alternative methods with the dual goals of achieving high order accuracy and minimizing the condition number of the solution process. Because of the design of magnetic confinement devices for plasmas, it is also important to be able to construct numerical methods in cylindrical coordinate systems, since the distribution functions \( f_b(v) \) are often axisymmetric or involve only a few azimuthal modes.

There is, of course, a substantial literature on computing Coulomb collisions and on solving elliptic partial differential equations in cylindrical coordinates. We refer the reader to \[5, 7, 11, 19, 22, 24, 28, 29, 31, 32, 33\] for some methods in current use in plasma physics. For a discussion of relativistic effects, see [6]. Most closely related to our approach are the methods of \[15, 30\] and \[20, 23, 24, 32\]. The first two are fast and achieve high order (“spectral”) accuracy, but use Fourier methods in Cartesian coordinates and do not address the axisymmetric (or low azimuthal mode) case. The latter rely on separation of variables in spherical coordinates, for which the axisymmetric case leads naturally to a representation involving Legendre polynomials and the general case to a representation involving associated Legendre functions.

In the numerical analysis literature, most solvers based on cylindrical coordinates tend to concern themselves with periodic (in \( z \)) or finite domain boundary conditions rather than free-space boundary conditions (see, for example \[9, 25\]). Here, we develop a method for computing the Rosenbluth potentials using separation of variables and a mix of integral equation and Fourier analysis techniques. We show that free-space (radiation) conditions can be imposed in a straightforward manner and that high order accuracy can be achieved in all derivatives with minimal loss of precision. The solver requires \( O(N \log N) \) work, where \( N \) is the number of grid points used to sample the distribution function.

Finally, we should make a remark about notation. The collision operator and the Rosenbluth potentials in (5), (6) are defined in velocity variables, for which we will use the standard cylindrical coordinates \((r, \theta, z)\) for \( v \). In the context of plasma physics, \( r = |v_{\perp}| \), where \(|v_{\perp}|\) is the magnitude of the component of the velocity perpendicular to the magnetic field, \( \theta \) is the gyrophase angle, and \( z = v_z \) is the component of the velocity field parallel to the magnetic field. The problem is purely axisymmetric when the velocity field is independent of the gyrophase angle.

One disadvantage of our solver is that we can be adaptive in the \( r \) direction, but not in the \( z \) or \( \theta \) directions, since we use spectral discretizations in the latter variables. For fully adaptive three-dimensional calculations, one could employ fast multipole-accelerated integral equation solvers,
as described in [14, 27]. These methods directly compute the convolution of the data \( f_0(v) \) with the free-space Green’s function. In the axisymmetric case, one could use an axisymmetric version of the fast multipole method [34]. The constant, however, is larger for these schemes than for methods based on separation of variables, and we limit our attention to methods that rely on a tensor product mesh in \( r, \theta \) and \( z \), which is adequate for most current simulations of the Boltzmann-Fokker-Planck equation [1].

2. The Poisson equation in cylindrical coordinates

In order to compute the Rosenbluth potential \( H_b \), we must solve the Poisson equation in free space

\[
\Delta u(v) = f(v).
\]

In cylindrical coordinates \( v = (r, \theta, z) \), we have

\[
u_{rr}(r, \theta, z) + \frac{1}{r} u_r(r, \theta, z) + \frac{1}{r^2} u_{\theta\theta}(r, \theta, z) + u_{zz}(r, \theta, z) = f(r, \theta, z),
\]

and we assume that \( f \) is identically zero outside the region

\[
\Omega = \{(r, \theta, z) : 0 \leq r \leq R, -A \leq z \leq A, 0 \leq \theta \leq 2\pi\}.
\]

Since \( u \) and \( f \) are periodic in \( \theta \), we represent them as Fourier series:

\[
u(r, \theta, z) = \sum_{n=-\infty}^{\infty} u^{(n)}(r, z)e^{in\theta}
\]

\[
f(r, \theta, z) = \sum_{n=-\infty}^{\infty} f^{(n)}(r, z)e^{in\theta}
\]

The derivatives in this representation will be written as

\[
u_{rr}(r, \theta, z) = \sum_{n=-\infty}^{\infty} u^{(n)}_{rr}(r, z)e^{in\theta}
\]

\[
u_{\theta\theta}(r, \theta, z) = \sum_{n=-\infty}^{\infty} u^{(n)}_{\theta\theta}(r, z)e^{in\theta}
\]

\[
u_{zz}(r, \theta, z) = \sum_{n=-\infty}^{\infty} u^{(n)}_{zz}(r, z)e^{in\theta}
\]

Substituting into (7) and equating terms corresponding to the \( n \)th azimuthal mode, we obtain:

\[
u^{(n)}_{rr}(r, z) + \frac{1}{r} u^{(n)}_{r}(r, z) + \frac{n^2}{r^2} u^{(n)}(r, z) + u^{(n)}_{zz}(r, z) = f^{(n)}(r, z).
\]

For each mode, we now have a partial differential equation (PDE) in the two variables \( r \) and \( z \) which we need to solve on the rectangular domain

\[
\Omega_{rz} = \{(r, z) : 0 \leq r \leq R, -A \leq z \leq A\}.
\]
Let us now take the Fourier transform of the equation in the z direction. That is we write
\[ u^{(n)}(r, z) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\kappa z} \hat{u}^{(n)}(r, \kappa) d\kappa \] (10)
\[ \hat{u}^{(n)}(r, \kappa) = \int_{-\infty}^{\infty} e^{-i\kappa z} u^{(n)}(r, z) dz \]
\[ f^{(n)}(r, z) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\kappa z} \hat{f}^{(n)}(r, \kappa) d\kappa \]
\[ \hat{f}^{(n)}(r, \kappa) = \int_{-\infty}^{\infty} e^{-i\kappa z} f^{(n)}(r, z) dz \]
\[ \hat{u}^{(n)}_{rr}(r, \kappa) + \frac{1}{r} \hat{u}^{(n)}_r(r, \kappa) = \left( \frac{n^2}{r^2} + \kappa^2 \right) \hat{u}^{(n)}(r, \kappa) = \hat{f}^{(n)}(r, \kappa). \] (12)

In the Fourier transform domain, the PDE becomes an ordinary differential equation (ODE), where \( \kappa \) (as well as \( n \)) is now fixed:

To simplify notation (when the context is clear), we will write \( \hat{u}(r) \) instead of \( \hat{u}^{(n)}(r, \kappa) \) and \( \hat{u}'(r) \) instead of \( \hat{u}^{(n)}_r(r, \kappa) \) to denote the derivative when discussing the solution of the ODE.

The equation (12) is an inhomogeneous modified Bessel equation [2]. In the homogeneous case, the equation has two linearly independent solutions, namely \( J_n(\kappa r) \) and \( K_n(\kappa r) \), the modified Bessel functions of order \( n \). The function \( J_n(\kappa r) \) is regular at the origin, and grows exponentially as \( r \to \infty \), while \( K_n(\kappa r) \) is logarithmically singular at the origin, but decays exponentially fast as \( r \to \infty \).

2.1. Boundary conditions for the modified Bessel equation

In order to have a properly posed ODE, we seek two boundary conditions, one at \( r = 0 \) and one at \( r = R \), beyond which the equation is homogeneous. For the \( n = 0 \) mode, the condition

\[ \hat{u}'(0) = \hat{u}_r^{(0)}(0, \kappa) = 0 \]

ensures regularity at the origin, while for modes \( n \neq 0 \)
\[ \hat{u}(0) = \hat{u}^{(n)}(0, \kappa) = 0 \]
is necessary. This is easily seen from taking the limit of the equation (12) as \( r \to 0 \).

Since we are seeking to solve the Poisson equation in free space, our ODE is actually posed on the half line \([0, \infty)\), with the radiation condition that the solution decay at infinity. This can be accounted for exactly in terms of a suitable boundary condition at \( r = R \). To see this, note that for \( r > R \) the solution must be proportional to \( K_n(\kappa r) \), since \( J_n(\kappa r) \) grows without bound. That is,

\[ \hat{u}(r) = C_{n, \kappa} \cdot J_n(\kappa r) \] for \( r \geq R \),

where \( C_{n, \kappa} \) is an unknown constant. The solution on \([0, R]\) and its derivative must match this solution at \( r = R \), so that

\[ \hat{u}(R) = C_{n, \kappa} \cdot K_n(\kappa R), \]
\[ \hat{u}'(R) = C_{n, \kappa} \kappa \cdot K'_n(\kappa R). \]
Eliminating the constant $C_{n,k}$ we obtain the exact "radiation" boundary condition:

$$\hat{u}(R) = \frac{K_n(|k|R)}{|k| \cdot K_n'(|k|R)} \hat{u}'(R) = 0.$$  \hfill (13)

In summary, the ODE boundary value problem we must solve is (12), subject to the boundary conditions:

$$\hat{u}'(0) = 0 \quad n = 0$$

$$\hat{u}(0) = 0 \quad n \neq 0$$ \hfill (14)

$$\hat{u}(R) = \frac{K_n(|k|R)}{|k| \cdot K_n'(|k|R)} \hat{u}'(R) = 0.$$ \hfill (15)

In broad terms, this completes the description of the Poisson solver, which proceeds in four steps.

---

**Informal description of fast Poisson solver**

1. Expand the right hand size $f(r, \theta, z)$ as a Fourier series in $\theta$, in order to get $\tilde{f}^{(n)}(r, z)$,
2. Compute the Fourier transform of $\tilde{f}^{(n)}(r, z)$ in the $z$ direction to get $\tilde{f}^{(n)}(r, \kappa)$,
3. Solve the ODE (12) for each $\kappa$ and $n$ to obtain $\hat{u}^{(n)}(r, \kappa)$,
4. Compute the inverse Fourier transform of $\hat{u}^{(n)}(r, \kappa)$ to get $u^{(n)}(r, z)$,
5. Sum the Fourier series in $\theta$ to get the final solution $u(r, \theta, z)$.

We will rely on fairly standard methods for all of the above, except Steps 3 and 4. For Step 3, we use an analytic solution based on knowledge of the underlying Green’s function for the ODE and accelerated by a simple “sweeping” algorithm. Step 4 will require some care, since it is straightforward to show that $\hat{u}^{(n)}(r, \kappa)$ is logarithmically singular as $\kappa \to 0$ for $n = 0$ and has a singularity of the order $\kappa^{2m} \log \kappa$ for $n \neq 0$.

### 3. Discretization and solution

We assume $f(r, \theta, z)$ is given on a tensor product grid with $N_\theta$ equispaced points in the $\theta$ direction on $[0, 2\pi]$, $N_z$ equispaced points in the $z$ direction on $[-A, A]$, and $N_r$ points in the $r$ direction on $[0, R]$. We divide $[0, R]$ into $N_r$ intervals with interval endpoints $R_0 = 0, R_1, R_2, \ldots, R_{N_r} = R$. We use a $P$th order (scaled) Chebyshev grid on each, so that $N_r = N_P$. We will denote by $(r_j, j = 1, \ldots, N_r)$ the grid points in increasing order. When the particular interval $m$ ($1 \leq m \leq N_r$) is of interest, the $p$th grid point on that interval ($1 \leq p \leq P$) is $r_j = r_{(m-1)P+p}$.

The discretized data will be denoted by

$$f_h(r_j, \theta_n, z_k) = f(r_j, \theta_n, z_k) \text{ for } 0 \leq j < N_r, \ 0 < n < N_\theta, \ 0 \leq k < N_z.$$

#### 3.1. Step 1: Transformation in \theta

We use the fast Fourier transform (FFT) to compute $f_h^{(n)}(r_j, z_k)$, the discretized version of $f^{(n)}(r, z)$:

$$f_h^{(n)}(r_j, z_k) = \frac{2\pi}{N_\theta} \sum_{l=0}^{N_\theta-1} e^{-2\pi i nl} f_h(r_j, \theta_n, z_k) \approx f^{(n)}(r_j, z_k).$$ \hfill (16)
It should be noted that, if \( f(r, \theta, z) \) is \( n \)-times differentiable, then the series truncated after \( N \) terms has an error of the order
\[
O\left(\frac{1}{N^{n-1}}\right) \quad \text{as} \quad N \to \infty.
\]
(17)

If \( f \) is infinitely differentiable, then the error goes to zero faster than any finite power of \( 1/N \). Schemes with this property are often referred to as having spectral accuracy. Moreover, the trapezoidal rule approximations of the series coefficients in (16) converge at the same rate [16, 35].

3.2. Step 2: Transformation in \( z \)

Since \( f(r, \theta, z) \) and \( f^{(n)}(r, z) \) are compactly supported, we need to compute the finite integral
\[
\hat{f}^{(n)}(r, \kappa) = \int_{-A}^{A} e^{-i\kappa z} f^{(n)}(r, z) dz.
\]

Letting \( h_z = \frac{2A}{N_z} \) and \( z_l = lh_z \), the trapezoidal rule yields:
\[
\hat{f}^{(n)}\left(r_j, \frac{\pi}{A} k\right) = \frac{2A}{N_z} \sum_{l=-N_z/2}^{N_z/2} e^{-i\kappa z_l} f^{(n)}(r_j, z_l) \approx \hat{f}^{(n)}\left(r_j, \frac{\pi}{A} k\right).
\]
(18)

This is computable using the FFT, and yields the values of the Fourier transform at equally spaced points of step size \( \pi/A \) in the \( \kappa \) domain. A few remarks are in order:

- The ratio \( \frac{N_z}{A} \) determines the range of frequencies that are resolved. If \( \frac{N_z}{A} \) increases (\( h_z \) decreases), higher frequency modes of the data are computed.
- We will assume that, to precision \( \epsilon \), \( \hat{f}(r, \kappa) \) is supported on the interval \([-\kappa_{max}, \kappa_{max}]\). For a given \( A, N_z \) must be chosen sufficiently large that \( \pi N_z/(2A) > \kappa_{max} \). (This is simply asking that the grid in \( z \) be fine enough to resolve the data.)
- Increasing \( N_z \) and \( A \) simultaneously so that \( N_z/A \) remains fixed leaves the range of \( \kappa \) unchanged, but increases the number of sample points where \( \hat{f}^{(n)} \) is computed in the range \([-\pi N_z/(2A), \pi N_z/(2A)]\).
- The trapezoidal approximation (13) is spectrally accurate, since the integrand and all its derivatives are assumed to have vanished by the time \( z = \pm A \).

3.3. Step 3: Solving the modified Bessel equation

We turn now to the solution of the modified Bessel equation (12), subject to the boundary conditions (15) for \( \kappa \neq 0 \). (As noted above, the equation has a weakly singular solution at \( \kappa = 0 \). Our quadrature rule for computing the inverse Fourier transform in section 3.4 will avoid the origin when integrating along the \( \kappa \) axis.)

One possible approach to solving the equation is to use a spectral integration-based ODE solver [17] that represents the second derivative as a Chebyshev series:
\[
\hat{u}''(r) = \sum_{k=0}^{N} a_k T_k(r).
\]
Multiplying the equation \((12)\) by \(r^2\) and systematic use of the following two identities for Chebyshev polynomials

\[
\int T_n(r) dr + C = \frac{1}{2(n+1)} T_{n+1}(r) - \frac{1}{2(n-1)} T_{n-1}(r)
\]

\[
r T_n(r) = \frac{T_{n+1}(r) + T_{n-1}(r)}{2}
\]

yields a banded linear system (of bandwidth 7) to which are appended two dense rows that correspond to the imposition of the desired boundary conditions. Such a system can be solved in linear time by careful Gaussian elimination, achieving spectral accuracy. For non-singular ODEs, this linear system can be viewed as the discretization of a second-kind integral equation for the unknown second derivative, and thus as a well-conditioned formulation of the problem. Unfortunately, in our case, the differential operator is singular at the origin. As a result, the integral equation is not of the second kind and the approach becomes ill-conditioned for fine grids, with the attendant loss of precision.

An alternative strategy is to use the fact that our ODE is classical and well studied, with a known Green’s function \(G_n(r, s)\). We can, therefore, write down the exact solution as a convolution:

\[
\hat{u}^{(n)}(r, \kappa) = \int_0^\infty G_n^0(r, s) f(s) ds
\]  

(19)

where

\[
G_n^0(r, s) = \begin{cases} 
I_n(\kappa r) K_n(\kappa s)/W(s) & \text{if } r \leq s \\
K_n(\kappa r) I_n(\kappa s)/W(s) & \text{if } s < r
\end{cases}
\]

where

\[
W(s) = \kappa (I_n' (\kappa s) K_n(\kappa s) - K_n'(\kappa s) I_n(\kappa s)) = -\frac{1}{s}
\]

This choice of Green’s function correctly imposes the regularity condition at the origin and the radiation condition at infinity. In this formulation, there is no need to solve a linear system - one needs only to evaluate the integral in (19). Naive implementation of this formula would require \(O(N^2)\) work. Because of the structure of the Green’s function, however, there is a simple \(O(N_r)\) solver based on the observation that

\[
\hat{u}^{(n)}(r, \kappa) = K_n(\kappa r) \int_0^r I_n(\kappa s) f(s)/W(s) ds + \\
kappa I_n(\kappa r) \int_r^\infty K_n(\kappa s) f(s)/W(s) ds.
\]  

(20)

The only source of error comes from the quadrature approximation of the preceding integrals. Derivatives of the solution are also obtained analytically. For example,

\[
\hat{u}^{(n)}_r(r, \kappa) = \kappa K_n(\kappa r) \int_0^r I_n(\kappa s) f(s)/W(s) ds + \\
k\kappa I_n(\kappa r) \int_r^\infty K_n(\kappa s) f(s)/W(s) ds.
\]  

(21)
There are some implementation issues in using (20), having to do with scaling and quadrature due to the fast growth/decay of Bessel functions for increasing $n$ and $r$. In particular, when $r$ lies in the $m$th interval denoted by $[R_{m-1}, R_m]$, we write

$$
\hat{u}^{(n)}(r, \kappa) = \frac{K_n(\kappa r)}{K_n(\kappa R_m)} \int_0^r I_n(\kappa s) K_n(\kappa R_m) f(s)/W(s)ds + \int_r^\infty \frac{K_n(\kappa s)}{K_n(\kappa R_m)} f(s)/W(s)ds .
$$

(22)

3.4. Step 4: Computing the inverse Fourier transform

We now need to compute the inverse Fourier transform of $\hat{u}^{(n)}(r, \kappa)$ to recover $u^{(n)}(r, z)$, according to (10). Since $\hat{u}^{(n)}(r, \kappa)$ is compactly supported to the desired precision on $[-\kappa_{\text{max}}, \kappa_{\text{max}}]$, we actually need to compute

$$
u^{(n)}(r, z) = \frac{1}{2\pi} \int_{\kappa_{\text{max}}}^{\kappa_{\text{max}}} e^{i\kappa z} \hat{u}^{(n)}(r, \kappa) d\kappa
$$

(23)

where (as discussed in section 3.2) $\pi N_z/(2A) > \kappa_{\text{max}}$. A complication is that $\hat{u}^{(n)}(r, \kappa)$ has a logarithmic singularity at $\kappa = 0$.

Fortunately, in the last decade or so, a variety of quadrature rules have been developed that rely on slight modifications of the trapezoidal rule, yield high-order accuracy, and still permit the use of the FFT. Two such schemes are the end-point corrected trapezoidal rule due to Kapur and Rokhlin [21] and the hybrid Gauss-trapezoidal rule due to Alpert [3]. We will make use of the latter.

**Theorem 1.** (modified from [3]). Let $f(\kappa)$ be a compactly supported function on $[-\kappa_{\text{max}}, \kappa_{\text{max}}]$ which is smooth away from the origin and takes the form

$$f(\kappa) = s_1(\kappa) \log(|\kappa|) + s_2(\kappa)
$$

in a neighborhood of the origin, where $s_1$ and $s_2$ are smooth functions. Let

$$I(f) = \int_{-\kappa_{\text{max}}}^{\kappa_{\text{max}}} f(\kappa) d\kappa
$$

and let $h = \frac{2\kappa_{\text{max}}}{N_z}$. Then, for every integer $m > 0$ and every $N_z > 2m$, there exist weights $w_{j,m}$ and nodes $\kappa_{j,m}$ such that

$$I_h(f) = h \sum_{\kappa = -N_z/2}^{N_z/2} f(kh) + \sum_{j=-m}^{m} w_{j,m} f(\kappa_{j,m})
$$

(24)

satisfies

$$I_h(f) = I(f) + O(h^m).
$$

In other words, the hybrid Gauss-trapezoidal rule achieves $m^\text{th}$ order accuracy by replacing the $2m$ trapezoidal nodes nearest the origin with specially located nodes (and weights). The paper [3] provides tables of these nodes for orders 2-16 (and the corresponding ones for a variety of other singularities as well).
In the present context, therefore, we will compute the integral (23) using the formula (24):

\[ u^{(0)}(r, z) = \frac{h}{2\pi} \sum_{k=-N_z/2}^{N_z/2} \hat{u}^{(0)}(r, \kappa_k) + \sum_{l=-m}^{m} w_{l,m} e^{i\omega_{l,m}z} \hat{u}^{(0)}(r, \kappa_{l,m}) \]  

(25)

with mesh spacing \( h = 2\pi N_z/(2A) \div N_z \), \( \kappa_k = \frac{\pi}{A} k \), and \( w_{l,m}, \kappa_{l,m} \) taken from [3]. Evaluating \( u^{(0)}(r, z) \) on our grid \( z_j = \frac{2\pi j}{N_z} \), we have:

\[ u^{(0)}(r, z_j) = \frac{1}{2A} \sum_{k=-N_z/2}^{N_z/2} e^{\frac{2\pi i k N_z}{2A}} \hat{u}^{(0)}(r, \kappa_k) + \sum_{l=-m}^{m} w_{l,m} e^{i\omega_{l,m}z_j} \hat{u}^{(0)}(r, \kappa_{l,m}) \]

The first term is straightforward to compute with the (inverse) FFT, requiring \( N_z \log N_z \) operations. The second term can be computed directly using the FFT, as discussed in section 3.2. The values at the irregular nodes can be computed directly or using the non-uniform FFT. For \( m \) sufficiently large, the sums can be computed simultaneously using the non-uniform FFT (see [13] and the more recent review [18]).

**Remark 1.** The quadrature rule (25) determines the discrete values of the continuous Fourier transform variable \( \kappa \) where \( \hat{u}^{(0)}(r, \kappa) \) needs to be sampled. The number of such points is \( O(N_z + m) \). This, in turn, tells us where \( \tilde{u}^{(0)}(r, \kappa) \) is needed. The values at the regular nodes are obtained with the FFT, as discussed in section 3.2. The values at the irregular nodes \( \kappa_{l,m} \) can be computed directly or using the non-uniform FFT.

**Remark 2.** (Oversampling). In practice, there is one more issue which needs to be addressed. In the integral (23), \( z \) is bounded by \( A \), so that the most oscillatory integrand is \( e^{iAx} \hat{u}^{(0)}(r, \kappa) \). It is easy to see that there are a maximum of \( N_z/2 \) periods of the function \( e^{iAx} \) over the interval of integration \([-\pi N_z/(2A), \pi N_z/(2A)]\). The maximum for the function \( \hat{u}^{(0)}(r, \kappa) \) is similar. Thus, the trapezoidal rule with \( N_z \) points yields only one point per wavelength for the most oscillatory argument, in violation of the Shannon sampling theorem. We, therefore, oversample the integrand by a factor of \( \eta \), by setting \( N_z' = \eta N_z \). (As discussed in section 5.2, we must simultaneously set \( A' = \eta A \) in computing the forward transform.)

Setting \( \eta = 1 \) yields exponentially small errors near \( z=0 \), but \( O(1) \) errors at \( z = A \). Setting \( \eta \geq 2 \) ensures convergence for \( z \) in the entire range \([-A, A]\), with exponential improvement as \( \eta \) increases. Setting \( \eta = 4 \) is sufficient for double precision accuracy for \( N_z > 16 \), assuming the function is bandlimited to machine precision at \( \kappa_{\max} = \pi N_z/(2A) \).

3.5. Step 5: Sum the Fourier in \( \theta \) to obtain the full solution

This is completely straightforward. As in Step 1, we may use the fast Fourier transform (FFT) to compute \( u(r_j, \theta, z_k) \) at equispaced points \( \theta_i = \frac{2\pi i}{N_\theta} \).

\[ u(r_j, \theta_i, z_k) \approx \frac{1}{N_\theta} \sum_{i=0}^{N_\theta-1} e^{i\frac{2\pi i}{N_\theta} \omega_{l,m}} u^{(0)}(r_j, z_k). \]  

(26)
3.6. Computing derivatives of the solution

One useful feature of spectral solvers is that derivatives are straightforward to compute with high order accuracy.

1. First and second r-derivatives: Our ODE solver returns both the solution \(\hat{u}^{(n)}(r, \kappa)\) and its derivatives \(\hat{u}_r^{(n)}(r, \kappa)\) and \(\hat{u}_{rr}^{(n)}(r, \kappa)\) on our grid. Thus, we can compute \(u_r(r, \theta, z)\) and \(u_{rr}(r, \theta, z)\) by the same technique as for \(u(r, \theta, z)\): evaluating the inverse \(z\)-Fourier transform and the \(\theta\) Fourier series for \(\hat{u}_r^{(n)}\) and \(\hat{u}_{rr}^{(n)}\), respectively.

2. z-derivatives: In the present paper, \(z\) derivatives are obtained through multiplication by \(ik\) in the inverse Fourier transform step:

\[
u^{(n)}(r, z) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{ikz} \hat{u}^{(n)}(r, \kappa) d\kappa \quad \Rightarrow \quad \frac{\partial^n}{\partial z^n} u^{(n)}(r, z) = \frac{1}{2\pi} \int_{-\infty}^{\infty} (ik)^n e^{ikz} \hat{u}^{(n)}(r, \kappa) d\kappa
\]

and the quadrature rule described above for logarithmic singularities.

3. \(\theta\)-derivatives: In the present paper, we also compute \(\theta\)-derivatives spectrally, by differentiating the Fourier series:

\[
u(r, \theta, z) = \sum_{n=-\infty}^{\infty} u^{(n)}(r, z) e^{in\theta} \quad \Rightarrow \quad \frac{\partial^n}{\partial \theta^n} u(r, \theta, z) = \sum_{n=-\infty}^{\infty} (in)^n u^{(n)}(r, z) e^{in\theta},
\]

using the FFT.

4. Mixed derivatives: Since derivatives in \(r, \theta, z\) are computed at independent steps of the algorithm, they are easily combined. For example, if we want to compute \(u_{rzz}\), we start with \(\hat{u}_r^{(n)}(r, \kappa)\) and compute the inverse \(z\) Fourier transform on the function \(-\kappa^2 \hat{u}_r^{(n)}(r, \kappa)\) to get \(u_{rzz}(r, \theta, z)\), followed by evaluating the Fourier series in the \(\theta\) direction via the FFT.

Remark 3. One can easily obtain \(z\) derivatives without numerical differentiation, once \(u, u_r, u_{rr}\) and \(u_{00}\) are known. The original PDE (7) becomes a second order ODE in \(z\), and the method of spectral integration [17] can be applied directly. We have not implemented this option, since the condition number of Fourier differentiation is only \(O(N)\), so that with 1000 points in \(z\) (or 1000 azimuthal modes), one can still obtain at least 10 digits of accuracy in double precision.

4. The biharmonic equation in cylindrical coordinates

For the Rosenbluth potential \(G_b\), we must solve the biharmonic equation in free space

\[
\Delta^2 u(\nu) = f(\nu).
\]

In cylindrical coordinates \(\nu = (r, \theta, z)\), after Fourier transformation in \(z\) and \(\theta\), we obtain the fourth order Bessel-type ODE:

\[
u_{r r r r} + \frac{2}{r} \nu_{r r r} = \left(1 + \frac{2n^2}{r^2} + \frac{2k^2}{r}\right) \nu_{r r} + \left(\frac{1 + 2n^2}{r^3} - \frac{2k^2}{r}\right) \nu_r + \left(\frac{4n^2 - 4n^2}{r^4} + \frac{2k^2 n^2}{r^2} + k^4\right) \nu = \hat{f}^{(n)}
\]

(27)
4.1. Boundary conditions for the fourth order Bessel-type equation

Since (27) is fourth order, we need four boundary conditions to have a properly posed ODE. We impose two at \( r = 0 \) and two at \( r = R \), beyond which the equation is homogeneous. To ensure regularity at the origin, for the \( n = 0 \) mode, it is sufficient to impose

\[
\hat{u}^{(0)}_r(0, \kappa) = \hat{u}^{(0)}_{rrr}(0, \kappa) = 0.
\]

For the \( n = 1 \) mode, we set

\[
\hat{u}^{(1)}(0, \kappa) = \hat{u}^{(1)}_r(0, \kappa) = 0,
\]

and for modes \( n \geq 2 \), we set

\[
\hat{u}^{(n)}(0, \kappa) = \hat{u}^{(n)}_r(0, \kappa) = 0.
\]

These conditions are easily derived by taking the limit of the equation (12) as \( r \to 0 \) and the fact that the null space of the differential operator is spanned by

\[
\{ I_n(|\kappa|r), r I'_n(|\kappa|r), K_0(|\kappa|r), r K'_n(|\kappa|r) \}.
\]

It remains to determine a radiation condition at \( r = R \), so that the derivative of the solution is bounded at infinity. As for the Poisson equation, we proceed by observing that the solution \( \hat{u}^{(n)}(r) \) for \( r > R \) must take the form

\[
\hat{u}(r) = C^1_{n, \kappa} K_0(|\kappa|r) + C^2_{n, \kappa} r K'_n(|\kappa|r),
\]

where \( C^1_{n, \kappa}, C^2_{n, \kappa} \) are unknown constants. This follows since the derivatives of \( I_n(|\kappa|r) \) and \( r I'_n(|\kappa|r) \) grow without bound. The solution on \([0, R]\) and its derivative must match this solution at \( r = R \), so that

\[
\begin{align*}
\hat{u}^{(n)}(R) & = C^1_{n, \kappa} \cdot K_0(|\kappa|R) + C^2_{n, \kappa} \cdot r K'_n(|\kappa|R), \\
\hat{u}^{(n)}_r(R) & = C^1_{n, \kappa} \cdot \kappa K'_n(|\kappa|R) + C^2_{n, \kappa} \cdot (K''_n(|\kappa|R) + \kappa r K'''_n(|\kappa|R)), \\
\hat{u}^{(n)}_{rr}(R) & = C^1_{n, \kappa} \cdot \kappa^2 K''_n(|\kappa|R) + C^2_{n, \kappa} \cdot (2 \kappa K'''_n(|\kappa|R) + \kappa^2 r K^{''''}_n(|\kappa|R)), \\
\hat{u}^{(n)}_{rrr}(R) & = C^1_{n, \kappa} \cdot \kappa^3 K^{'''}_n(|\kappa|R) + C^2_{n, \kappa} \cdot (3 \kappa^2 K^{''''}_n(|\kappa|R) + \kappa^3 r K^{'''''}_n(|\kappa|R)).
\end{align*}
\]

Eliminating the constants, we obtain two exact “radiation” boundary conditions to be imposed on the combination of \( \hat{u}^{(n)}(R) \) and its first three derivatives. The formula is complex and omitted, since we won’t use it. We will instead use an exact solution based on the Green’s function.

4.2. Discretization and solution

The solution of the biharmonic equation is analogous to that of the Poisson equation, so we just highlight the differences.

After separation of variables, we need to solve a fourth order Bessel type equation. As before, we could proceed by expanding the highest derivative in a Chebyshev series and integrating, but the resulting linear system again loses precision because of the singular nature of the differential operator at the origin. (The loss is, in fact, much more severe than for the second order (Poisson) equation.)

Alternatively, we can construct the Green’s function for the ODE using the linearly independent
fundamental solutions $I_0(\kappa r)$, $rI'_0(\kappa r)$, $K_0(\kappa r)$, $rK'_0(\kappa r)$, imposing the regularity condition at $r = 0$ and the decay condition as $r \to \infty$:

$$G^{(0)}(r, s) = \begin{cases} [I_0(\kappa r)sK'_0(\kappa s) + rI'_0(\kappa r)K_0(\kappa s)]/W(s), & r \leq s \\ [K_0(\kappa r)sI'_0(\kappa s) + rK'_0(\kappa r)I_0(\kappa s)]/W(s), & r > s \end{cases} \quad \text{with} \quad W(s) = -\frac{2\kappa}{s}$$

The solution involves computing four integrals (instead of two). The sweeping method is virtually the same as that used for the Poisson equation.

$$\hat{u}^{(0)}(r) = K_n(\kappa r) \int_0^r sI'_0(\kappa s)f(s)/W(s)ds + rK'_0(\kappa r) \int_0^r I_0(\kappa s)f(s)/W(s)ds$$

$$+ \int_r^\infty I_0(\kappa s)f(s)/W(s)ds + rI'_0(\kappa r) \int_r^\infty K_0(\kappa s)f(s)/W(s)ds.$$  

The Fourier transform of the solution in $z$ has a more severe singularity in the biharmonic case, due to the fact that the free-space Green’s function does not decay. Fortunately, however, we are only interested in second derivatives of the biharmonic potential, and they have only logarithmic singularities, so our special-purpose quadratures from section 3.4 yield the desired accuracy. (More elaborate methods involving singularity subtraction could be developed if one wanted the biharmonic potential or its first derivatives.)

5. The collision operator

Now that we’ve described how to solve the Poisson and biharmonic equations (5) and (6), we turn our attention to the collision operator (4). If we express all the derivative terms in cylindrical coordinates, the axisymmetric collision operator becomes:

$$C(f^a, f^b) = \frac{\gamma_{ab}}{m_a} \left[ C_b(f^a, f^b) - 2 \left( 1 + \frac{m_a}{m_b} \right) C_p(f^a, f^b) \right]$$

$$C_p(f^a, f^b) = -4\pi f^a f^b + f^a_H^b + f^a_H^b$$

$$C_b(f^a, f^b) = -8\pi f^a f^b + f^a \left[ 2G^b_{rr} + 2G^b_{rr} + \frac{2G^b_{rr}}{r^2}G^b_{rr} - \frac{1}{r^2}G^b_{rr} \right] + f^b \left[ \frac{2}{r}G^b_{rr} + 2G^b_{rr} + 2G^b_{rr} \right]$$

$$\quad + f^b \left[ f^a G^b_{rr} + 2f^a G^b_{rr} + f^a G^b_{rr} \right]$$

6. Numerical Examples

In order to test the convergence of the algorithm, it is desirable to compare the results to a nontrivial exact solution. For a right-hand side consisting of a radially symmetric Gaussian:

$$f(\rho) = \frac{E}{(4\pi V)^{1/2}} e^{-\frac{\rho^2}{2}} \quad E = e^{-\frac{x^2}{2}}, \quad \rho^2 = x^2 + y^2 + z^2.$$  

we can compute the exact solution to both the Poisson and the biharmonic equations

$$\Delta u = f \quad \Delta^2 v = f$$
as well as to the components of the collision operator $C_p, C_b$:

\[
\begin{align*}
    u(\rho) &= -\frac{R}{4\pi \rho}, \\
    v(\rho) &= -\left[\frac{\rho}{8\pi} + \frac{\nu}{4\pi \rho}\right] R - \frac{\sqrt{\nu}}{4\pi^{3/2}} E, \\
    C_p &= -\frac{E^2}{8\pi^2 \nu^3} + \frac{ER}{16\pi^{3/2} \nu^{5/2} \rho}, \\
    C_b &= -\frac{E^2}{2\pi^2 \nu^3} + \frac{ER}{4\pi^{3/2} \nu^{5/2} \rho}.
\end{align*}
\]

(30)

After a change of variables to cylindrical coordinates, we can find explicit analytic formulas for all quantities produced by our solvers (although some of the formulas need to be treated carefully to avoid catastrophic cancellations in their numerical evaluation).
Example 1: Let us first consider the convergence of the solver for a single Gaussian of variance $\nu = 0.223$. Using the Chebyshev (spectral integration) solvers, note that after an initially rapid convergence, the higher derivatives start to diverge due to the ill-conditioning of the linear system (Fig 1).

Figure 1: The relative $L_\infty$ errors for the spectral integration based solvers, when applied to the Poisson and biharmonic equations (top and bottom, respectively).
Using the Green’s function based ODE solver, we see that the solution and its derivatives converge spectrally without the loss of precision for fine grids that affects the spectral integration based schemes (Fig. 2).

Figure 2: The relative $L^\infty$ errors for the Green’s function based solvers, when applied to the Poisson and biharmonic equations (top and bottom, respectively).
Fig. 3 depicts the runtime performance of the solver. In the present implementation, the spectral integration based code for the Poisson equation requires about 25 seconds for 5 million grid points, while the biharmonic solver is about three times slower. This is within a small factor of the performance of FFT-based codes for doubly or triply periodic constant coefficient PDEs on regular grids. The Green’s function based solvers are a bit slower at present. We estimate that straightforward optimization/precomputation could result in a factor of 2-3 speed-up.

![Overview of runtimes of all solvers − Good FFT size data points](image)

Figure 3: Runtime for the Green’s function and spectral integration based solvers, when applied to the Poisson and biharmonic equations.
Example 2: The Coulomb collision operator appears as a source term in the Boltzmann-Fokker-Planck equation (1). Not considering convective and electromagnetic terms, it affects the evolution of ionized gas consisting of a single species as

\[ \partial_t f + (\text{convective and electromagnetic terms}) = C(f, f) \]

For a single species of ion, the Maxwellian distribution (a Gaussian centered at the origin) is an equilibrium state. From the formulas (28) and (30), it can be verified analytically that \( C(f^G, f^G) = 0 \). Computing this result numerically on a 192 \( \times \) 128 grid we obtain zero to about 14 digits. Note that \( C_p \) and \( C_b \) do not vanish independently. There is a real cancellation between the two contributions when inserted into the formula (28).

\[
\begin{array}{|c|c|c|}
\hline
\|C_p(f^G, f^G)\|_\infty & \|C_b(f^G, f^G)\|_\infty & \|C(f^G, f^G)\|_\infty \\
3.403 \cdot 10^{-3} & 1.361 \cdot 10^{-2} & 5.851 \cdot 10^{-14} \\
\hline
\end{array}
\]

To illustrate the diffusive nature of the collision operator, let us consider a perturbation to the equilibrium solution, by constructing an anisotropic Gaussian density, with slightly different variances in the \( r \) and \( z \) directions:

\[ f(r) = \frac{1}{\sqrt{4\pi v_r v_z}} e^{-(x^2 + y^2)/(4v_r)} e^{-z^2/(4v_z)} \]

where \( v_r = 1.107 \) and \( v_z = 1.353 \).

Figure 4: Anisotropic Maxwellian, its difference from a Maxwellian and its computed collision operator

Notice that where the anisotropic distribution is too small (the valley in the central plot), the collision operator is positive, thus it tends to increase \( f \). Where the anisotropic distribution function is too large (the peaks in the central plot), the collision operator is negative, tending to decrease \( f \). Thus, the collision operator indeed has the effect of moving an anisotropic Maxwellian towards an isotropic equilibrium distribution.
To demonstrate the full three dimensional solver, we placed three Gaussian source densities at
different locations in $\mathbb{R}^3$, two with positive weight and one with negative weight. The following
plot shows the solution overlaid on the $r-\theta-z$ grid as a three dimensional contour surface plot
with an octant “cut out”:

We used Gaussian variances of 0.2, 0.6 and 0.3 centered at the points with Cartesian coordinates
$(4.3, 1.2, 3.6), (-1.1, 4.1, -0.8), \text{ and } (5.3, 3.5, -3.2)$ and weights $-1.0$, $-1.3$, and $1.2$, respectively.
With a grid in $(r, \theta, z)$ of $192 \times 64 \times 96$ on $[0, 16] \times [0, 2\pi] \times [-16, 16]$, we obtained 12 digits of
accuracy in the solution and its gradient. The total execution time was 18 secs. on a single core
of a 2.5GHZ CPU, using an oversampling factor in $z$ of 4.
7. Conclusion

This paper describes a new fast solver for separable elliptic partial differential equations in cylindrical coordinates that is both fast and high-order accurate, with solution times comparable to a few FFTs using the same number of degrees of freedom. Combined with the Rosenbluth formalism, it permits the rapid evaluation of the Coulomb collision operator in kinetic models of ionized gases.

Our solver is particularly useful when the number of azimuthal modes is small. For full three-dimensional problems, it is quite likely that Cartesian-based methods will be more effective, particularly since one can use fast multipole-based, fully adaptive solvers. Here, we require regular grids in the \( \theta \) and \( z \) directions, which is sufficient for many applications. Several open problems remain. One involves the construction of fast, fully implicit collision operators, so that large time steps can be taken in the Boltzmann-Fokker-Planck equation (for which there is already a significant literature). Another involves the development of methods for solving elliptic partial differential equations in complicated axisymmetric geometries rather than in free space - that is, interior or exterior to a surface of revolution. These problems are currently being investigated, with progress to be reported at a later date.

Acknowledgements

We thank C.S. Chang and Eisung Yoon for several useful discussions. This work was supported by the National Science Foundation under grant DMS06-02235 and the Department of Energy under contract DEFG0288ER25053.

References


